

**TEXT CLASSIFICATION USING A HIDDEN
MARKOV MODEL**

Kwan Yi

Graduate School of Library and Information Studies

McGill University

Montreal

January 2005

**A thesis submitted to the Faculty of Graduate Studies and Research in partial
fulfillment of the requirements for the degree of Doctor of Philosophy**

Copyright © Kwan Yi, 2005



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-494-12966-2

Our file Notre référence

ISBN: 0-494-12966-2

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

ABSTRACT

Text categorization (TC) is the task of automatically categorizing textual digital documents into pre-set categories by analyzing their contents. The purpose of this study is to develop an effective TC model to resolve the difficulty of automatic classification. In this study, two primary goals are intended. First, a Hidden Markov Model (HMM) is proposed as a relatively new method for text categorization. HMM has been applied to a wide range of applications in text processing such as text segmentation and event tracking, information retrieval, and information extraction. Few, however, have applied HMM to TC. Second, the Library of Congress Classification (LCC) is adopted as a classification scheme for the HMM-based TC model for categorizing digital documents. LCC has been used only in a handful of experiments for the purpose of automatic classification. In the proposed framework, a general prototype for an HMM-based TC model is designed, and an experimental model based on the prototype is implemented so as to categorize digitalized documents into LCC. A sample of abstracts from the *ProQuest Digital Dissertations* database is used for the test-base. Dissertation abstracts, which are pre-classified by professional librarians, form an ideal test-base for evaluating the proposed model of automatic TC. For comparative purposes, a Naïve Bayesian model, which has been extensively used in TC applications, is also implemented. Our experimental results show that the performance of our model surpasses that of the Naïve Bayesian model as measured by comparing the automatic classification of abstracts to the manual classification performed by professionals.

RÉSUMÉ

La catégorisation textuelle (CT) est la tâche d'analyser le contenu de documents textuels numériques afin de les catégoriser automatiquement dans des catégories prédéterminées. L'objet de cette étude est de développer un modèle de CT capable de résoudre les difficultés encourues lors de la classification automatique de texte. Dans cette étude, nous voulons atteindre deux buts. D'abord, un Modèle caché de Markov (*Hidden Markov Model*, MCM) est suggéré comme nouvelle méthode pour la catégorisation textuelle. Le MMC a déjà été utilisé dans un grand nombre d'applications dans le domaine du traitement de texte tels la segmentation textuelle la recherche d'information et le prélèvement d'information. Peu de chercheurs, cependant, ont appliqués le MCM à la CT. En second lieu, la *Library of Congress Classification* (LCC) est adoptée en tant que système de classification pour notre modèle de CT pour la catégorisation de documents numériques basés sur le MCM. La LCC a rarement été utilisée pour classifier des documents de façon automatique. Notre technique de classification présente un prototype général pour un modèle de CT, basé sur le MCM, et un modèle expérimental, basé sur notre prototype, est mis en oeuvre afin de catégoriser des documents digitaux dans leurs catégories selon la LCC. Un échantillon de résumés provenant de la base de données *ProQuest Digital Dissertations* est utilisé pour notre test de base. Les résumés de thèses, préclassifiés par des bibliothécaires professionnels, constituent une base de test idéale afin d'évaluer le modèle proposé de CT automatique. À titre de comparaison, un modèle Bayésien naïf (*Naïve Bayesian Model*), souvent utilisé dans des applications de CT, est aussi mis en oeuvre. Nos résultats expérimentaux démontrent que la performance de notre modèle MCM surpasse celle du modèle Bayésien naïf lorsqu'on compare notre classification automatique des résumés à la classification exécutée par les bibliothécaires.

ACKNOWLEDGEMENTS

Working with this dissertation was exciting, instructive, and fun in the last five years of my study at McGill. Without the help, support, and encouragement of many persons, I would never have been able to finish this work.

First, I would like to thank my supervisor Jamshid Beheshti for his inspiring and encouraging ways in guiding me to a deeper understanding of knowledge work from all angles, his invaluable comments on any subject including my personal life, and his generosity, kindness, and concern about my financial condition and willingness in welcoming discussions. From our discussions, I received much more than what I had expected; our talks always turned out to be more than joyful. I feel lucky to have had him as my supervisor.

I also give special thanks to all my committee members: Andrew Large for reading this work thoroughly several times and his priceless comments in all aspects, John Leide for providing invaluable advice that nobody else had thought of, and Doina Precup for providing priceless guidance in modeling and experiments. For this dissertation, they have worked as a great team in that their subject specialties and fruitful comments really embody this dissertation. I learnt a lot by reading their valuable comments.

I'd also like to extend my respects to my defense committee members. Their thoughtful input to this treatise makes this work much more valuable.

I am very grateful to Mélanie Raymond who has edited this manuscript and also written the French abstract, and her spouse Young-Lark Jin who has supported me in many ways. Also, special thanks to all my friends including, Chansoo, Emeka, Hansu, Heungsun, Jaeyoung, Kyungkyun, Min, Pung, Seungjin, Sungkyu, Taeho, and Tao (in alphabetic order).

Thanks also to all my colleagues and the academic and administrative staff at the Graduate School of Library and Information Studies for providing me with a good working atmosphere. I've especially enjoyed lunching and chatting with colleagues.

At last, to my beloved wife Jeeyoun and lovely daughter Judith-Younsoo, thanks for supporting me with your dedicated love and understanding, and continual endurance for the last fourteen years, and to my parents (Dalsun Yi & Sunlim Park) and mother-in-law (Heonlim Park), thanks for devoted love and patience. What can I say about them? I know it is absolutely impossible to express their devotion with mere words.

We would like to give special thank to the Online Computer Library Center (OCLC), INC for allowing us to use data from the OCLC Cat CD database for this research under cooperative agreement. The financial support of the Social Sciences and Humanities Research Council of Canada is also gratefully acknowledged.

TABLE OF CONTENTS

List of Figures.....	iv
----------------------	----

List of Tables	vi
----------------------	----

1 Introduction.....	1
1.1 Introduction.....	1
1.2 Statement of the Problem	2
1.3 Purpose of the Study.....	4
1.4 Research Questions.....	6
1.5 Significance of the Study	7
1.6 Organization of the Dissertation	8
2 Automated Text Classification.....	10
2.1 Classification	10
2.2 Library Classification Systems.....	12
2.2.1 Background.....	12
2.2.2 Dewey Decimal Classification.....	13
2.2.2.1 Introduction	13
2.2.2.2 Major Characteristics	14
2.2.2.3 Relation to Other Bibliographic Systems	15
2.2.2.4 Summary.....	15
2.2.3 Library of Congress Classification	16
2.2.3.1 Introduction	16
2.2.3.2 Major Characteristics	17
2.2.3.3 Relation to Other Bibliographic Systems	18
2.2.3.4 Summary.....	19
2.3 Automated Text Classification.....	19
2.3.1 Definitions	19
2.3.2 Elements of a TC Task	21
2.3.3 TC and Clustering.....	22
2.4 Machine Learning Approach to TC	25
2.4.1 Introduction.....	25
2.4.2 Machine Learning Principles	27
2.4.3 Machine Learning and Its Use in TC Applications	29
2.5 Automated Classification and Library Classification Systems	32
2.5.1 Introduction.....	32
2.5.2 Pharos.....	33
2.5.3 Scorpion	34
2.5.4 DESIRE	35
2.5.5 Wolverhampton Web Library (WWLib).....	36

2.6	Summary and Conclusion	37
3	Hidden Markov Model and its Applications to TC.....	40
3.1	Hidden Markov Model	40
3.2	Components of a Hidden Markov Model	45
3.3	Three Fundamental Problems for Hidden Markov Model	47
3.3.1	The First Problem.....	48
3.3.2	The Second Problem	56
3.3.3	The Third Problem	57
3.4	HMM in Information Management Applications.....	61
3.4.1	Information Retrieval	62
3.4.2	Information Extraction	62
3.4.3	Text Segmentation.....	63
3.4.4	Text Summarization	64
3.4.5	Summary	65
3.5	Summary and Conclusion	66
4	A Conceptual Framework For An HMM-based TC Model	68
4.1	Introduction.....	68
4.2	HMM Classification Model	69
4.2.1	Theoretical Background	69
4.2.2	Model Prototype	70
5	An HMM-based Text Document Classification System	74
5.1	Introduction.....	74
5.2	Model Data	78
5.2.1	Training and Test Data.....	78
5.2.2	Selection of LC classes	80
5.2.3	Data Selection.....	83
5.2.4	Data Processing	100
5.2.4.1	Text Analysis.....	100
5.2.4.2	Feature Selection.....	102
5.3	Relationship of LCCs and LCSHs	103
5.4	Building a HMM-based Classifier	106
5.4.1	Introduction	106
5.4.2	Model Training.....	106
5.4.3	Experimental Settings	110
5.4.4	Model Inference	113
6	Experiments and Their Results.....	114
6.1	Introduction	114
6.2	Delimitations and Limitations.....	114
6.2.1	Delimitations	114
6.2.2	Limitations	116
6.3	Use of Data Sets and Baseline Classifier	117

6.3.1	V-fold Cross Validation	117
6.3.2	Baseline Classifier	118
6.4	Model Effectiveness	118
6.4.1	Introduction	118
6.4.2	Classification Accuracy	121
6.4.3	Classification Accuracy Confidence	122
6.5	Experimental Results	123
6.5.1	Introduction	123
6.5.2	HMM-based system vs. NB-based system	123
6.5.3	Hard vs. Soft Disciplines	129
6.5.4	Analysis of Experimental Results	133
7	Conclusions and Discussion	139
7.1	Summary and Conclusions	139
7.2	Contributions	142
7.3	Further Research	144
Appendix A. List of the Abbreviations Used		146
Appendix B. List of Hard/Soft Disciplines		149
Appendix C. Selected Perl Scripts		150
Bibliography		164

LIST OF FIGURES

2.1	Similarity measurement in <i>Clustering</i> and <i>Classification</i>	24
2.2	A framework for a machine learning task.....	26
3.1	A Markov model for tossing one coin	42
3.2	A hidden Markov model for a beverage vending machine	44
3.3	Interpretation of the induction step of the forward procedure.....	51
3.4	A dynamic algorithm procedure for the calculation of a forward variable.....	52
3.5	Interpretation of the induction step of the backward procedure.....	55
3.6	Illustration of being in state i at time t and in state j at time $t+1$, using the forward and backward variables	59
4.1	A generic HMM prototype for a specific subject	72
4.2	A proposed HMM prototype for a specific subject	73
5.1	An overview of the proposed HMM-based classification system	77
5.2	An example of an OCLC-MARC record from the CatCD™ Recent Books, May 2000	89
5.3	An example of raw training data	93
5.4	An Example with 050 and 650 fields only	93
5.5	An example of the output of the text manipulation process.....	93
5.6	A snapshot of the OCLC WorldCat™ database search screen	97

5.7	An instance of the dissertation list using the search query in Figure 5.6.....	98
5.8	Relationships of LCSH and LCC numbers to LCC subjects	105
5.9	A trained HMM for the subclass QR (Microbiology).....	112
6.1	Probability mass functions of classification accuracy for the HMM-based and NB-based systems.....	128
6.2	Cumulative probability functions of classification accuracy for the HMM-based and NB-based system	128
6.3	Classification accuracy graph for the HMM-based TC system	131
6.4	Classification accuracy graph for the NB-based TC system.....	131
6.5	Classification accuracy graph for three top-level LC classes (Q, S, N).....	132
6.6	Trends of classification accuracies and alpha values	132

LIST OF TABLES

5.1 The abridged list of the hard/soft disciplines (Source: McGraph, 1978, p. 22).....	82
5.2 The LCC top-level class distribution of MARC records in the OCLC CatCD™ Windows, May 2000	88
5.3 Interpretations of indicators and sub-field codes for 050 and 090 tags	90
5.4 Interpretations of indicators and sub-field codes for the 650 tag.....	91
5.5 Statistics of OCLC-MARC records from OCLC CatCD.....	92
5.6 The selected three main classes and their subclasses	99
6.1 Contingency table for the multi-valued classes $C = \{C_1, C_2, \dots, C_n\}$	119
6.2 Comparison of the actual category rankings from two TC systems	120
6.3 Analysis of experimental results by ranking of actual categories for the HMM system.....	138
6.4 Analysis of experimental results by ranking of actual categories and LC classes	138

Chapter 1

INTRODUCTION

1.1 Introduction

Within this new information environment, the nature of digital information may be characterized as being both dynamic, mainly due to Web content, and growing in quantity. Information overflow, in particular, has been a severe obstacle. Locating and retrieving information relevant to information users' needs and interests has been a primary research issue in the field of information retrieval. The adoption of classification has been applied to tackle this particular problem.

A widely-known commercial application of the introduction of classification to digital textual information is the classification of Web pages, called *Web directories*, by Yahoo!, LookSmart, and Open Directory. With the exception of Open Directory, which relies upon more than fifty thousand volunteers (Open Directory project¹), these portals employ a few hundred professional librarians to manually classify and index Web pages into a hierarchical subject structure. Perhaps due to the high cost of this labor, less than one percent of all the digital information available on the Web is reported to be covered by the Web directory services (Sullivan, 2003). The rate of classifying information cannot catch up with that of creating information. The problems described above clearly describe why research on automatic classification is necessary, as opposed to manual classification.

The task of automatic classification is a relatively new IR subfield. Since machine learning (ML) serves as a theoretical foundation for the methodologies in this task, its scope is often referred to as the intersection of IR and ML. The ML paradigm is learning

¹ Open Directory Project. <<http://dmoz.org>>. Visited on 18 July 2003.

knowledge on pre-defined issues such as a subject or event. There are two different types of learning paradigms involved in this learning process, depending on the way in which data is labeled data: supervised and unsupervised. In supervised learning, data (examples or instances) are all pre-labeled and the labeled data are used in the process of learning, whereas, in unsupervised learning, data are not labeled and unlabeled data are used. Various types of statistical classification models based upon the ML paradigm have been adopted and attempted for the task of classification (Crawford *et al.*, 1991; Lewis & Ringuette, 1994; Ng *et al.*, 1997; Joachims, 1998). The central issues of this decade-duration of research have been the development of effective classification models and evaluation methods, and the comparison of different models. With such a short research history being taken into consideration, their research achievements are remarkable enough to be applied to real-world problems. Nevertheless, it should be pointed out that the classification task is by nature highly difficult due to the subjectivity of classification. Therefore, the problem of identifying the subjects of a digital document and of classifying them into a certain set of categories still remains a highly challenging task.

1.2 Statement of the Problem

Text Classification (or Text Categorization)² (TC) may be viewed as a systematic approach to organizing information in light of information retrieval problems. It is a means to cope with the information glut. From the standpoint of TC, information users' needs and interests are pre-determined and transformed into a certain set of categories. TC aims to classify newly acquired information based upon these categories in order to provide information users with easy access to relevant information.

Since the early 1990s, interest in automated TC has rekindled and research on it has flourished based upon the ML approach. In the ML framework, two different groups of learning are documented: supervised and unsupervised. In supervised learning, all

² These two terms, text classification and text categorization, are popularly used in scholarly articles in the fields of information science and computer science, to refer to the same task. In this dissertation, the term "text classification" will be used.

training data are labeled (classified) and such labeled data are used in learning activities, whereas, in unsupervised learning, training data devoid of labels are used. To date, various TC models, supported by different theoretical foundations, have been proposed and experimentally tested. Examples of the ML-based TC models are as follows: regression models (Schütze *et al.*, 1995), Neural Network (NN) classifiers (Yang, 1994), Bayesian models (Lewis and Ringuette, 1994), decision trees (Crawford *et al.*, 1991), and Support Vector Machines (SVM) (Joachims, 1998). The kernel of TC research is to have the ability of topically comprehending the content of text. The difficulty of this task lies in that it inherently has a highly subjective nature at least as much as there are inconsistencies among human-beings' decisions. Many classifiers under the ML paradigm have been developed, and their performances have been measured and compared to humans' decisions. The results from some high-performing classifiers show that their classification capabilities have improved such that they are comparable to that of human beings, but there still remains a significant gap. Moreover, experimental results of various TC models in different environments have shown that the performance of TC models is not reliable within different settings and applications. The fluctuation of the models' performances indicates that there might be particular situations where different models would work better or worse than others. The Naïve Bayesian (NB) model has been one of the more popular methods used in TC due to its simplicity and relative effectiveness (Mitchell, 1997; Lewis, 1998; McCallum & Nigam, 1998). In some experimental studies, however, the performance of the NB model has turned out to be inferior to other models such as SVM, k-Near Neighbor (k-NN), NN (Joachims, 1998; Yang & Liu, 1999). The outcome of many studies confirms that there is no single omni-competent TC model. Instead, distinct models seem to be robust for different aspects of TC and within different contexts. The following characteristics of TC models have been reported in research articles: k-NN-based models are easily scalable to large data sets (Yang, 1997); NN-based are best suitable for applications to obscure intrinsic structures (Schweighofer & Merkl, 1999); NB-based are appropriate for their simplicity and extensibility to Web documents with links (Lee & Myaeng, 2002); and SVM-based may be used for their resistance to over-fitting and large dimensionality (Glover, et al., 2002). The current research on TC is

still far from revealing the true nature of these models, the settings in which the models were tested, and the relation between models and tasks models are applied.

1.3 Purpose of the Study

This study proposes a machine-learning classification model for digitalized textual documents (hereafter called digital documents). The focus of this dissertation is on the investigation of the hidden Markov model (HMM) as a model for subject detection and to examine its applicability for the TC application.

The primary goal of this research is to design an automatic, content-based classification model for digital documents based on a traditional subject classification scheme. The research goal can be split into its components, and each described separately: (1) automatic, (2) content-based, (3) classification model, (4) digital document, and (5) subject classification scheme.

1. Automatic:

The *manual* process of classification has an inherent limitation in the number of documents that can be covered. Although human intervention may produce more accurate results in classifying documents, the *automatic* classification process is more desirable and promising when a large quantity of documents to be classified and the cost consumed for the classification process by human beings are taken into consideration. In fact, these are the reasons that the research on automatic classification is receiving more attention and becoming more significant than ever before.

2. Content-based:

The classification process in this research relies upon the content of the documents only. That is to say that the classification is performed on the full text rather than any metadata elements that have been attached to the text. This text content-based research treats a document as a list of words, and a document is viewed as a container for all the words written in the document (essentially a *bag-of-words*.)

Since text is the research target for subject classification, the major issue for this research is how to understand a document's content semantically for the discovery of its subjects. In this study, the task of apprehending the subjects of a document is tackled statistically where the statistical regularities of word occurrences in text are unearthed.

3. Classification model:

The classification model's design is the central issue of this research. This study applies the ML-based approach to the classification task, where the classification rules for a specific subject are statistically learned from a set of documents pre-classified for the same subject. The HMM has solid theoretic foundations and has been successfully applied in a range of text-related applications (Rabiner, 1989). In this study, HMM will be used as a classification model and described for its application to the classification task.

4. Digital document:

As mentioned in the previous section, the classification task in this study is interested in digitally formatted text as its target object. To avoid confusion, the term *digital document* will be used to denote the text written in a document of digital format. Therefore, although a digital document can comprise various types of information such as text, images, figures, and hyperlinks, this study will refer to text only, unless otherwise specified.

5. Subject classification scheme:

A subject classification scheme signifies a classification structure under which digital documents will be categorized. The selection of a classification scheme is closely related to the issues of a classification model's design and of the availability of data (a set of pre-classified documents under the classification scheme) for the training and test of a classification model. Related to this, the classification model should prepare the algorithms and procedures to represent the classification scheme.

1.4 Research Questions

The primary research question guiding this study is: Can the HMM be used to automatically classify digital documents utilizing a conventional subject classification system? The primary research question can be divided into the subordinate questions which attempt to address more specific concerns and problems provided by this study:

1. **Representation Problem:** How can the proposed HMM represent a specific classification scheme?

A subject-based classification system can be viewed as a subject map because it serves as a guide to help people find specific subjects. Our concern, therefore, can be rephrased as the problem of representing a specific subject map by a HMM-based model. A subject map comprises information for the topics of a particular classification system as well as for the relationships between these topics. Therefore, the problem to be tackled may be rewritten as how to make a HMM-based TC model that is able to recognize topics and their relationships from a subject map.

- a. How is the model able to identify the subjects in the classification scheme?
- b. How does the model recognize relationships between different subjects?

2. **Decision Problem:** How does the HMM determine the most relevant subject categories for a new document?

The primary issue of the decision problem focuses on the mechanism of allocating the degree of topical relevance of a document to each subject in the subject map considered in the model. The proposed TC model accepts a new document as its input, analyzes it statistically using HMM, and provides a ranked list of subjects as a result of the analysis, with the ranking being determined by a particular mechanism. The two specific questions below summarize the decision problem:

- a. What is the strategy for deciding the most suitable subject for a document?

- b. How is the list of relevant subjects ranked?
- 3. **Remodeling Problem:** Given the rapidly changing digital environment, how flexible and adaptable is the proposed model?

After a subject map has been used for the construction of a TC model, it must be possible to add new subjects and remove obsolete ones. The established model should support this dynamic, fine-tuning feature so that it reflects the changes within the subject map. Furthermore, the model must have the capability of upgrading itself through a re-training process with newly acquired training data for any currently used subject, not necessarily just for new subjects. The remodeling problem might include the following questions:

 - a. How does the model make adjustments to acknowledge new training data for existing topics?
 - b. How does the model incorporate the introduction of new subjects?

1.5 Significance of the Study

This section describes the significances of the research in theoretical and applicable points of view.

1. Extension of the use of Library of Congress Classification (LCC) to a digital library application. This study brings into being a classification application where the proposed classification model is evaluated. In the application, digital textual documents are organized by subjects according to LCC classes and consequently, the organized documents are accessed and retrieved by exploring various subjects of interest. LCC is one of the popular classification systems used in libraries, particularly in academic libraries, and the use of LCC is incorporated into Online Public Access Cataloging (OPAC) systems. Therefore, the realization of LCC as a classification system for digital documents shows the potential of integrating digital information into OPAC systems, thereby creating a new multifaceted type of library collection. Consequently, digital and non-digital documents on a specific subject could be found virtually in the same location in on-line applications. So far, no

standardized classification scheme for digital information has been made, and the development of a systematic framework for the classification of digital information is in demand. This study proposes a solution to this practical issue.

2. Extension of HMM to TC. Another significance of this study is in the introduction of the HMM to TC research. HMM has been widely used in various applications such as speech recognition and bioinformatics. For their part, IR-related areas, such as text segmentation and event tracking (van Mulbregt, 1998; Yamron, 1998), text summarization (Conroy, 2001a, 2001b), information retrieval (Miller, 1999), and information extraction (Freitag, 1999; Leek, 1997) are relatively new to the HMM. Only a few studies, however, have reported on the use of the HMM to TC-related fields. This study may be considered as expanding the application of the HMM to the field of TC.
3. The HMM as a framework for concept representation. In the proposed classification model, since a specific subject is represented by a HMM, a HMM can be also viewed as concept representation, rather than as a classifier. By the same reason, a classification model for a set of categories may be called a subject map representing the categories. From this perspective, the use of the HMM will be extended to the new application of concept representation.

1.6 Organization of the Dissertation

The rest of this dissertation is organized as follows: The next two chapters are generally referred to as the literature review of this study. Chapter 2 is mainly written for reviewing the subjects in this study. The concept and characteristics of two major library classification systems and automated TC and their approaches are reviewed. Then, some research projects on automated TC using a library classification system are described. Chapter 3 is mainly written for reviewing the methods used in this study and related applications for its use. HMM is used as a model of the TC system implemented in this study. HMM and its use in related applications are delineated. Then, Chapter 4 is prepared to provide a conceptual framework for this study where the model's prototype is

described. In Chapter 5, this study's methodologies are explained in detail, including data selection and the system building procedure. Chapter 6 describes the experimental environments that were designed as well as the experimental results. Finally, Chapter 7 discusses our conclusions and future directions.

Chapter 2

AUTOMATED TEXT CLASSIFICATION

The two components of the automatic classification system in this study are: (1) the use of a popular library classification system for an automated classification and (2) the adoption of HMM as an automated classification model. This chapter is to provide background on the first component, whereas the following chapter is on the second component. This chapter begins with the exploration of the concept of classification. In the following section, the two major library classification systems will be reviewed with the focus on the comparison of their roles and systems. Then, their uses in automatic classification will be described.

2.1 Classification

Human beings seem to be born with an inherent desire to organize (Taylor, 1999) and classification is a prototypical form of organization (Svenonius, 2000). The need for organization has led to the development of classification systems and organization tools. The human efforts for classification have been manifested in various academic disciplines (Satija, 1998): logic for methods of logic, psychology for learning- and memory-related works, philosophy for identifying and revealing the relationships among variables (Soper *et al.*, 1990), linguistics with reference to terminology and semantics, and library and information science for the design and use of classification systems. The traditional principles of classification in library science are derived from the principles of classification in logic and philosophy (Chan, 1994).

In a general and broad sense, classification is defined as the fundamental activities of analysis, sorting, filtering, identifying, categorizing, grouping, arranging, ordering,

ranking, correlating, organizing, and controlling, etc. (Satija, 1998) According to this definition, classification is deemed as an essential process for an organized system. There is another type of definition found in a concrete and narrow sense, saying that classification is “*the act of grouping like things together*” with the ‘things’ being referred to as concrete entities, the ideas of such entities, or abstractions (Buchanan, 1979, p. 9). This last view on classification seems to be mainly concerned with the activity of categorizing or grouping.

General Classification Theory (GCT) provides a comprehensive and concrete vision for the principles of classification, and at the same time, it has been a foundation for major library classification systems such as the LCC and the Dewey Decimal Classification (DDC) (Marcella & Newton, 1994). The ten tenets of GCT are as follows (Richmond, 1990, pp. 17-18):

1. *Every thing, object, notion, etc. has to have a distinct and unambiguous description of its unique qualities.*
2. *Principles involving likeness and distinctness must be used in creating classes.*
3. *Hierarchies and other relational methods are necessary in order to group fundamental characteristics and to identify fundamental differences clearly.*
4. *The final system should appear as a logical progression from general to particular.*
5. *The system must be hospitable to all knowledge, including things that never were; things that never shall be; and things that are impossible.*
6. *Each classification system must have means of covering every context, including future additions. Its hospitality has to be such that additions and changes can be made easily.*
7. *Each classification system must have cross-references and an index.*
8. *A method of constant updating is mandatory for adjusting the old and adding what is new.*
9. *A method must be found for automatic adjustment of class numbers to suit the needs of the system as it grows. This suggests adoption of hospitable computers as a necessity.*
10. *A concordance, in addition to tables, schedules and index, probably would be exceedingly valuable in keeping up with terminology.*

The GCT principles listed above describe guidelines for the definition of class (the first through the fourth in GCT) and the principle of notation (the fifth, sixth, and eighth) of

library classification systems, and are reflected in modern library classification systems and organization tools (the seventh) such as subject headings.

2.2 Library Classification Systems

2.2.1 Background

Classification is an essential foundation of library and information science (Palmer, 1971). *Library classification* is defined as the subject-based systematic way of organizing library materials (Maltby, 1975). *Library classification schemes* are viewed as the representation of knowledge (Lois, 1994; Rafferty, 2001; Satija, 1998). More specifically, they are concerned with a system mapping the fields of knowledge into subjects in a systematic way (Dittmann & Hardy, 2000), and subjects in classification systems, from the general to the specific, are defined through schedules of classes, divisions, and sections (Mortimer, 2000).

Library classification schemes were originally developed to organize primarily printed materials such as books and serials, and this has been their major use for over a century. In the online environment, the potential role of a library classification scheme as a tool for subject access to information has been recognized and explored (Markey & Demeyer, 1986; Svenonius, 1983). Classification schemes are incorporated into the systems, and the embedded classification schemes provide a means of subject searching with the help of subject headings or index terms. In the electronic environment prompted by the computer revolution, both the means of organization and the nature of the objects to be organized have been dramatically affected (Svenonius, 2000). Electronic information and Internet resources such as online databases and electronic journals, Web sites, videos, and images have emerged as new types of objects to be organized, catalogued and accessed. The use of library classification schemes for dealing with these new types of objects has been explored: Vizine-Goetz (1997) compared Yahoo's twenty top most popular categories with the DDC and the LCC. The results of comparing the mapping between Yahoo and library schemes indicated that the DDC and LCC topic coverage were wide enough to cover Internet resources. Koch *et al.* (1997) reviewed various classification schemes used

on Web sites, ranging from universal schemes, such as DDC and LCC, to subject-specific schemes such as the National Library of Medicine (NLM) and Engineering Information (EI), and found that the DDC, LCC, NLM, and EI classification schemes were integrated within the subject schemes of Library of Congress Subject Headings (LCSH), Medical Subject Headings (MeSH), and the EI thesaurus. Williamson (1997) investigated nine Web sites using major library classification schemes (three each from LCC, DDC and Universal Decimal Classification (UDC)) with respect to level of division applied, adaptations to the systems, method of display etc., and concluded that the universal library classification schemes provide a framework for organizing Internet resources and contribute to improving retrieval at the first two levels of schemes.

In this study, text in a digital environment is set as the object of interest for automatic classification. Also, the use of a library classification scheme, LCC, as an organizing and browsing tool is explored in the framework of a statistical model. This research is distinguished from the cataloging work of electronic information resources and Internet resources embedded in online cataloguing systems in that this study attempts to understand the textual contents and to assign the most probably relevant classes based on content analysis.

2.2.2 Dewey Decimal Classification

2.2.2.1 Introduction

The DDC is a universal classification scheme for dividing knowledge into a hierarchical structure of the division by ten. Since its first publication in 1876, the DDC has become the most widely used library classification system in the world (Chan, 1996) and is the most popular in most public libraries in the United States. Melvil Dewey devised it alone as an assistant college librarian based on an earlier classification system by W.T. Harris. Melvil Dewey himself was involved in the revision as a supervisor until its thirteenth edition. An editorial group called the *Decimal Classification Editorial Policy Committee*, consisting of the staff in the organizations of the Library of Congress (LC) and the Online

Computer Library Center (OCLC), as well as some publishers, works for the revision of DDC.

2.2.2.2 Major Characteristics

DDC is a discipline-major classification scheme and the divisions of DDC main classes are executed by discipline, rather than by subject (Scott, 1998). There is no single DDC class devoted to a given subject (Taylor, 1992), which connotes that materials belonging to a single subject can appear in multiple places reflecting different disciplines. After the subdivision by discipline, subject, geographical period specification, and form are applied for organizing knowledge when necessary. At the top-level of the 21st DDC, there are ten main classes split into ten academic disciplines from the Humanities, the Social Sciences, etc. The disciplines are not all evenly treated to reflect today's dominant organization of knowledge. That is, the sub-disciplines under Humanities, such as Language and Philosophy, are located at the same level as the Social Sciences and other sciences, which are supposed to be subordinate (Chan, 1994).

The DDC is a hierarchical classification in that a class in a level indicates a more general discipline or subject than a class in its subordinate level (Bloomberg & Weber 1976). Starting with main classes, each of them is divided into ten divisions, and each subclass is divided into ten sections, etc., indicating the transition from the general into the specific. There exist three relationships between classes: *coordinate*, *subordinate*, and *superordinate*, and at least two sets of relationships are applied to a class (Dewey, 1989). Notation is the system of symbols used to represent the classes in a classification system. The DDC notation is expressed in Arabic numerals from zero (0) to nine (9) and decimal points, with decimal expansion for divisions, and a number with its location signifies a class level and its corresponding discipline or subject. The number in the first digit refers to a class in the first level, the one in the second digit indicates a class in the second level, etc. After the third level, further topic hierarchies are achieved by consulting corresponding summary tables. Therefore, the hierarchical order and relationship between classes is preserved and manipulated through the DDC notation (Williamson, 1995).

The Relative Index is a guide for the subjects or topics related to disciplines (Scott, 1998), which is another unique feature of Dewey classification. Due to the discipline-major classification nature of DDC, items for a subject are dispersed into various disciplines. The Relative Index brings together the various scattered aspects of a topic to one place (Bloomberg & Weber, 1976).

2.2.2.3 Relation to Other Bibliographic Systems

Research for the integration of DDC and the LCSH has been conducted. A statistical mapping of LCSH to each Dewey number was included in the electronic version of DDC 20 (Comaromi *et al.*, 1989). In 1994, a project for linking DDC and LCSH started at OCLC. In the OCLC project, the pairs of LCSH and DDC numbers were collected from the OCLC WorldCat database and a list of popular LCSH suitable for Dewey numbers was produced, relying on simple pair frequencies and a statistical estimator in a later study (Vizine-Goetz, 1998). The product of WebDewey, the electronic version of the DDC, includes about 90,000 LCSH mapped to Dewey numbers and links to LCSH authority records corresponding to the mapped LCSH (OCLC WebDewey). Also, the classification schemes of LCC and NLM Classification and the subject headings systems of LC Children's Headings and MeSH have been used in OCLC's vocabulary research projects to make links with DDC for the expansion of DDC vocabulary (Vizine-Goetz *et al.*, 2004).

2.2.2.4 Summary

The salient merits and weaknesses of DDC are described, with respect to their comparison to LCC. The merits of DDC may include: (1) the notation system consisting of numbers in decimals is easily understood and thus can be adapted to systems in other countries; (2) the decimal number-based notation of hierarchical structure facilitates the shelving, organizing, and understanding due to its systematic organization; (3) the notation system self-explains the relationships of classes, such as coordination or subordination. The weaknesses of DDC may include: (1) some disciplines in main classes are inappropriate as today's academic disciplines, and the disciplines of Humanities are

not equally treated in comparison to other coordinate disciplines within the Social sciences, Pure sciences, and Applied sciences; (2) the decimal system of ten base has a limitation on the number of subjects on a same level to ten divisions only; (3) by the virtue of the sequential attribute of its number system, DDC can grow indefinitely in the direction of specificity. However, it lacks flexibility in creating new classes between adjacent classes in a same coordinate level, e.g., between 550 and 560.

2.2.3 Library of Congress Classification

2.2.3.1 Introduction

The LC classification was designed for classifying the over one and a half million (which was a huge volume at that time) collection of the LC and its expansions in 1897. The original development of LCC with reference to the LC collections is stated in Charles Bead (1968, p. 18):

“The LC classification, being completely based on the Library’s collections, is coextensive in scope with the book stock of the Library of Congress. Therefore, the LC classification is comprehensive but not truly universal at the present time. Expansion of the classification is governed by and depends upon the acquisition of new material.”

Cutter’s Expansive Classification was selected as the chief guide for developing the new scheme with considerable changes in notation (qtd. Chan, 1999, p. 7)³. Unlike DDC, which was devised through the efforts of a single professional librarian, the overall process of the design and development of LCC was controlled under the supervision of J.C.M. Manson and C. Martel with a group of specialists assigned to the development of each of the LC main classes. As a consequence, the schedules for individual main classes and their subclasses were published separately. Class Z (bibliography and library science) was published as the first LC class in 1902 and the schedule for class K (law) was the last

³ The original source is from Hansen, J. C. M. “The Library of Congress and Its New Catalogue: Some Unwritten History.” Essays Offered to Herbert Putnam by His Colleagues and Friends on His Thirtieth Anniversary as Librarian of Congress, 5 April, 1929. Eds. William Warner Bishop & Andrew Keogh, New Haven, Connecticut: Yale University Press, 1929.

in 1969. Although the initial design and development of LCC was aimed at classifying LC collections only, LCC has been widely utilized in most academic and research libraries as well as large-size public libraries in the North America.

2.2.3.2 Major Characteristics

Like DDC, LCC is a classification system organizing knowledge by discipline. However, a salient difference between LCC and other modern classification systems including DDC is that LCC is essentially an *enumerative* scheme, in the sense that aspects of a subject are all pre-coordinated and listed in detail in auxiliary tables of schedules, as opposed to DDC and Colon Classification using notational extension for pinpointing the specificity of subjects (Chan, 1999).

For LCC notation, alphabetic letters and Arabic numerals are used. An LCC main class or subclass is represented by capital letters. For further divisions, a range of 1 and 9999 integers with possible decimal extensions and Cutter numbers, when necessary, are appended after letter symbols. When two LCC numbers are compared, the relationship between the classes is not clearly identified due to the ambiguity expressed by their notations because LCC notation preserves the order but not the hierarchical structure of LCC (Williamson, 1995). Another aspect of notation is *hospitality*. A classification system is said to be hospitable if it is capable of dealing with further expansion (Chan, 1994). Compared to the Dewey system especially, LCC is relatively high in hospitality, resulting from the benefit of generous notational provision. There are “*hundreds of number-letter combinations compatible with the notation that have not yet been employed or have ever been retired in favor of new locations The scheme will accommodate for a long time the many new subjects and aspects of subjects not yet anticipated*” (Wynar 1992, p. 351). Not only bountiful combinations of letters and numeric symbols, but also the introduction of decimal extensions and Cutter numbers are the contributing factors for its hospitality.

As mentioned, the LC notation is not hierarchical. However, LCC has a hierarchical structure, in the sense that disciplines or subjects are sprouting from the general to the specific as further developed. The nature of the LCC hierarchy is similar to that of the

DDC. A set of main classes on the top of the hierarchy represents a list of disciplines, and each of them is divided into subclasses for more specific disciplines, except the E and F (history in America), and Z (bibliography and library science) classes. Then, further subdivisions are generally made by topic, place, time, and form.

2.2.3.3 Relation to Other Bibliographic Systems

The explicit links between LCC and LCSH can be found in several sources. In USMARC subject authority records, there are fields for a classification number (field 053) and for heading fields (fields 150 and 151). Vazine-Goetz and Markey (1989) found in their analysis of the LC Subject Authority file that LC classification numbers appeared in approximately 43% of topical subject heading records (field 150). The US Machine Readable Cataloging (MARC⁴) records Format for Bibliographic Data has fields for LC classification data (field 050) and for controlled subject headings (fields 600, 610, 630, 650, and 651). The USMARC classification record format has fields for Index Terms (fields 700-754) that have subject or thesaurus terms such as LCSH or MeSH for additional subject access to the classification number (field 153) (Guenther, 1996). In the print version of LCSH, there are some subject headings associated with a range of LCC classes rather than specific classes suggested (Koch, *et al.*, 1997). The LCC has also linked with other classification schemes through information in bibliographic MARC records. In USMARC Format for Bibliographic Data, there are fields for a Dewey classification number (field 082) and a NLMC call number (field 060) as well as a LC call number (field 050). The similar linking data can also be found in other MARC catalogue records supplied by bibliographic utilities.

LCSH has been involved in various mapping activities with LCC and DDC. In OCLC vocabulary projects, LCSH/DDC pairs were generated using OCLC WorldCat records containing both DDC numbers and LCSH (Vazine-Goetz, 1998). Other mappings of LCSH to other classifications and thesaurus such as LCC and Educational Resources Information Center (ERIC) were also made. The relations of the classification schemes,

⁴ A MARC record refers to bibliographic data for a library item that can be read by a computer, including the information on its classification, description, and subject.

subject headings, and thesauri associated in the mapping research projects were tabularized in Vizine-Goetz *et al.* (2004). With the vocabulary mappings of LCSH, classes represented by the associated classification systems' notations can have descriptive representations in LCSH. Vizine-Goetz *et al.* (2004) summarize what the mapped LCSH can bring: (1) it provides additional indexing vocabulary, (2) assists catalogers in assigning subject headings, and (3) helps the use of classification schemes for automated classification services.

2.2.3.4 Summary

Some merits and weakness of LCC are listed here, as opposed to DDC. The merits may include: (1) the LCC notation is flexible and hospitable to accommodate further expansion; (2) each of the LC class schedules has been developed by subject experts, rather than generalists; (3) LCC reflects the nature of the collections in academic and research libraries due to its original focus on the LC collection. The weaknesses may include: (1) the LCC is generally deemed to be an enumerative scheme. Alphabetical subject arrangements are often used when a hierarchical structure is logically needed; (2) LCC was not designed suitably for general libraries but rather for the purpose and collections of the LC, which causes it to dedicate relatively less space for the humanities and philosophy and more space for social sciences; and (3) inherent multi-topical works such as multi-element work cannot be precisely arranged unless certain provisions are made.

2.3 Automated Text Classification

2.3.1 Definitions

Lewis (1992, p. 37) described TC as “*the classification of documents with respect to a set of two or more predefined classes*,” and Sebastiani (2002, p. 1) defined it as the activity of “*labeling natural language texts with thematic categories from a predefined set*.” A TC may be seen as the task of assigning a pre-defined set of classes to documents. Its three

components may be expressed in an operational definition: analyzing textual documents, understanding their relevant classes, and classifying them into a pre-defined set of classes.

Some notations related to TC tasks are defined, which will be used throughout this dissertation. Let $D = \{d_1, d_2, \dots, d_n\}$ be a set of documents, and $C = \{c_1, c_2, \dots, c_m\}$ be a set of classes. Also, define C^* to be the power set of C , that is, the set of all subsets of C . More formally, a TC can be viewed as a function $F: d \xrightarrow{c} \{0,1\}$, where $d \in D$ is a document and $c \in C$ is a class, that accepts a document as an input and yields true as 1 or false 0, given a class. The output of 1 means that the document is interpreted to fall into the considering class, and it is interpreted not to fall into the class with the output of 0.

Essential are the following understandings on TC tasks:

- There is no limitation on the nature of the classes to be used. Thus, any arbitrary set of classes can be adopted, such as a set of the classes consisting of ‘Like’ and ‘Hate.’
- The content of documents is the target research object of this task, rather than a document’s metadata.

The primary element of a TC system is the realization of a function F . As described in the proposed TC definition, the function serves to measure the relevance of a document given a class. The scope and methods where the function works can be limited by the TC conceptual model on which it relies.

TC tasks can be classified into different types, according to the number of classes and the number of class labels. If there are only two classes to be considered, it is said to be a binary classification task, where the value of m in the set C is equal to 2. With more than two classes, it is called a multi-class classification, where the value of m is larger than 2. When each document is associated with one class label, it is called a binary-label classification, where the class label of a document is an element of the set C . In multi-label classification, each document has at least one class label, where the class label of a document is an element of the set C^* .

2.3.2 Elements of a TC Task

A definition of a general TC task may be described as the role of a function that takes a set of documents as input and relevant classes as output. In this section, the key elements of a TC process are explored to provide a deeper understanding of a TC process: classification object, classification scheme, and classification model.

- Classification object:

This element refers to the input of a TC function, that is, a set of objects that need to be classified. In TC applications, a digitized textual document is considered as a target classification object. It consists of a list of words. Generally, the document length in words is not as long as a book, and not so short as to comprise only a few words. Particularly, dealing with short documents seems to be a challenge in TC.

A set of corpuses has been popularly used in TC research for the enhancement of TC models and related techniques: Reuters⁵ Newswire data (Lewis, 1992) only for the scopes of economics and business and the Ohsumed⁶ medical document collection. Four different versions of the Reuters corpus, for the range of 9,603 (93 classes) and 14,704 (135 classes) documents, have been developed and adopted. Ohsumed's collection is a subset of a medical comprehensive database, compiled by the National Library of Medicine, covering 270 medical journals (Hersh *et al.*, 1994). These data collections have been widely involved as a common data source in TC experiments for comparisons of different TC models and methods. The WebKB⁷ is a collection consisting of 8,282 Web pages, established and maintained by the text learning group of Carnegie Mellon University (Blum & Mitchell, 1998). This collected corpus was manually classified into seven classes

⁵ A Reuters corpus (<http://about.reuters.com/researchandstandards/corpus/>) has been released to provide a high quality benchmark collection for the research of information retrieval and machine learning systems. There are several different versions provided, which has been modified by research groups for their use. The current standard version, Reuters Corpus Volume 1, is available, containing 810,000 English language Reuters from 1996-08-20 to 1997-08-19.

⁶ Available from <ftp://medir/ohsu.edu/pub/ohsumed>.

⁷ Available from <http://www-2.cs.cmu.edu/~webkb>

such as *student*, *faculty*, and *course*, and used in TC experiments with Web documents (Craven & Slattery, 2001; Fürnkranz, 1999). Different from the academic Web pages of WebKB, a corpus of company Web pages (Ghani *et al.*, 2000), compiled from the Hoovers Online Web resource⁸ that contains detailed information about a large number of companies, was also used to study Web pages classification (Yang *et al.*, 2002).

- Classification scheme:

The classification scheme is linked to a pre-defined set of classes. In the research on TC, a wide range of classification schemes has been included, ranging from a simple binary classification, such as ‘favorite’ and ‘non-favorite’, to a complex classification scheme with its structure, such as LCC. A few TC applications (Dolin *et al.*, 1998, 1999; Koch and Ardö, 2000; Shafer, 1997; Shafer *et al.*, 1999) adopt the conventional classification schemes used in libraries. Some TC applications based on the library classification systems are described in Section 2.5.

- Classification model:

The third element indicates a model for a TC function, often called a classifier for its realization. A classification model includes the following essential functions: (1) It should support a way of representing a document; (2) it should provide a method of integrating the classification scheme used; and (3) it should present a method of measuring the similarity between a document and a class. The three functionalities above will be used as criteria in the comparative study of classification models.

2.3.3 TC and Clustering

An IR technique of *clustering* is akin to TC, in grouping similar documents together, which originated in the 1970s (van Rijsbergen, 1979). In a more recent IR book (Baeza-

⁸ <http://www.hoovers.com>

Yates & Ribeiro-Neto, 1999 p. 438), it is defined as “*the group of documents which satisfy a set of common properties.*” As the definition implies, in clustering, any set of classes is not taken into consideration in document grouping. Instead, by measuring conceptual distances between documents, a group of documents close to each other is formed.

TC and clustering rest upon similar principles, but are quite distinct in their approach and techniques. Figure 2.1 depicts how they are different. A salient feature lies in the utilization of a set of classes. In clustering, similar documents are not constituted against a class, whereas, in TC, they are. Subsequently, the objects of similarity measurement are different. In TC, the degree of similarity between a document and a target class is measured to see how relevant the document is to the class. In a clustering process, however, the similarity among documents is measured in terms of a conceptual distance, to see how similar the documents are in content.

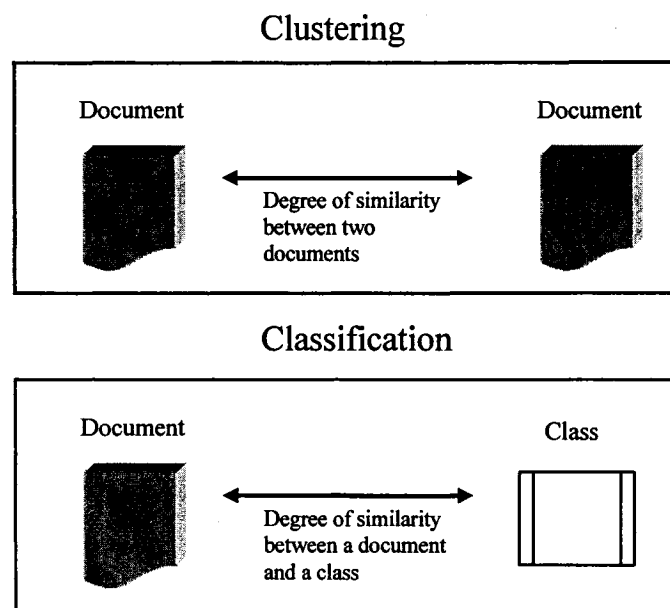


Figure 2.1: Similarity measurement in *Clustering* and *Classification*

2.4 Machine Learning Approach to TC

2.4.1 Introduction

Research on TC has evolved to tackle the problem of finding the category most relevant to a document, given a set of categories. In the 1970s, the approach based on Salton's vector model was adopted to find subject clusters. In this approach, a text document is represented by a vector where a set of index terms selected from the document is used as the elements of the document vector. Also, a form of vector represents a category, which is called a category vector. In this approach, the physical distance between document vectors and category vectors in Euclidean space estimate the conceptual similarity of a document to a specific category. This approach is still widely used in modern TC systems (Baeza-Yates & Ribeiro-Neto, 1999).

In the 1980s, the knowledge-based approach attempted to identify the rules governing subject classification from domain experts, and incorporate them into the system with the help of knowledge engineers. Since information about the rules is limited to the knowledge of subject-domain specialists, the coverage of the obtained information is limited by the knowledge of domain experts, referred to as the *knowledge bottleneck* problem (Sebastiani, 2002).

In the 1990s, a new procedure for collecting the subject-related rules from data was supported by machine learning techniques. The machine learning based TC systems automatically *learn* the classification rules for the recognition of subjects from human experiences. Different machine learning techniques support different learning algorithms or *learning* classification rules. In this context, *human experience* means a set of pre-classified data consisting of classification objects and the corresponding subject labels. As the major models used in TC are adopted from research in Artificial Intelligence (AI), the development of different approaches to TC has followed the evolution of AI research. As the major paradigm in AI research has shifted from a knowledge-base approach to a machine learning based approach, so have the methodologies used in TC.

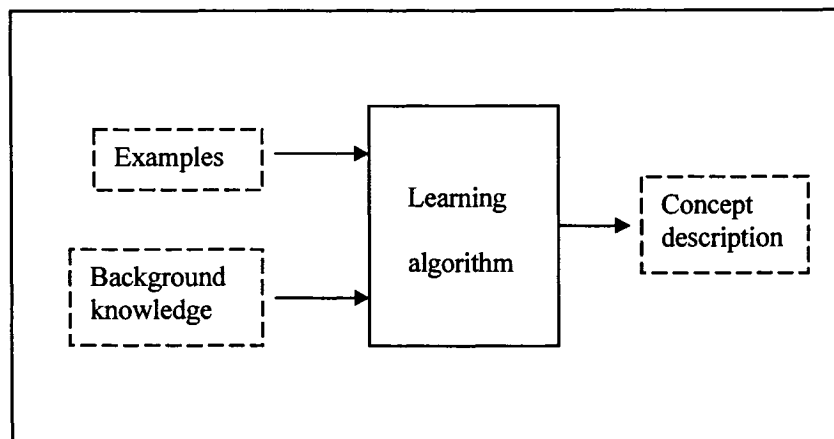


Figure 2.2: A framework for a machine learning task (Kubat *et al.*, 1997)

2.4.2 Machine Learning Principles

Learning is an ambiguous and abstract concept difficult to define. In Webster's Ninth New Collegiate Dictionary, it is defined as "(1) *the act or experience of one that learns*; (2) *knowledge or skill acquired by instruction or study*; (3) *modification of a behavioral tendency by experience*." This definition of learning implicitly refers to human or animal learning, which has long been a research topic for psychologists and zoologists. In ML, however, the computer becomes the object learning, where the ultimate goal of ML is to enable computers to have the ability of acquiring knowledge or experience. One might ask, "Why is ML needed?" There are several reasons why ML is important. First, there are learning tasks infeasible to human beings due to a large amount of information. The amount of information available about the tasks is too large to be handled by human beings. To make matters worse, information about the tasks keeps changing. Second, some tasks are not well defined, and are represented by examples that are more favorable to machines, rather than human beings. Web IR can be seen as an example of a ML task. In the Web IR task, machines (search engines) equipped with the ability of classifying all the indexed Web pages into relevant and non-relevant sets to a specific user query, present relevant results to the users, and repeat this process for each query. It is impossible to conduct such a task by relying on the human learning process.

The definition of *learning* from the ML perspective has the same base as the one for human learning, but focuses on the machine's performance. Learning in ML is viewed as the process of improving the system's performance by acquiring knowledge from experience (Langley, 1988) and is described as a cycle encompassing the following four components: *learning*, *knowledge*, *performance*, and *environment*. Mitchell (1997) identifies (1) task, (2) performance measure, and (3) experience as the three features for a general machine learning problem.

For the description of a general ML principle, the framework for a ML task by Kubat *et al.* (1999) will be used and depicted in Figure 2.2. In this framework, the four components, *examples*, *background knowledge*, *learning algorithm*, and *concept description*, are co-related as input, output, and a black box (function) in between.

Examples⁹ as an input to the black box convey knowledge for the target concept to be learned. An *example* consists of a *description* of the concept and *value* to the description. The *description* of an *example* is represented in various forms, such as vector, rule, and list of terms. The decision of the form for the description is subject to the learning algorithm applied in the black box. One of the most popular forms is the vector-based representation. In this case, each component of a vector is called a feature¹⁰. The *value* of an *example* can have any value among a finite set of numbers or categorical values, subject to the target concept. In most classification tasks of ML, the *value* is binary, which specifies a positive or negative *example*. Let X be an *example*. A vector-based representation of it can be $X: (X_1, \dots, X_n) = \text{True}$.

Background knowledge as another input to a learning algorithm in the framework consists of prior knowledge about the target concept to be learned. For example, when a chess game is considered to be a learning task, chess rules could be the background knowledge in the ML framework, whereas a set of the changed *chessboard* positions can be the examples for the experience of the chess task. Inductive ML methods, such as decision tree learning and neural networks, are inductive learning paradigms that make generalizations about a target concept based mainly on a number of training examples. For their part, analytic ML methods, such as explanation-based learning, use a deductive reasoning to achieve concept learning. Prior knowledge is the main resource component for analytic methods, along with training examples.

The *Learning algorithm* serving as a black box in the ML framework is the learning method that is concerned with how to learn and what is to be learned. Langley (1996) presented five paradigms on learning algorithms: (1) neural networks (2) instance-based or case-based learning (Bareiss, 1987; Rissland, 1989) (3) genetic algorithms (4) rule induction and (5) analytic learning (DeJong & Mooney, 1986; Mitchell *et al.*, 1986). Other recent major learning streams are the reinforcement learning algorithm (Sutton, 1988) and the SVM method (Joachims, 1998 & 1999). A general goal of ML is to make the machine gain knowledge from previous experience. In the inductive ML approach, the

⁹ Other various terms are also used: *sample*, *instance*, *input vector*, *feature vector*, etc.

¹⁰ It is also called *attribute* and *variable*.

target knowledge will be achieved in a representation form specific to the learning algorithm, and a learning algorithm is applied in such a way that the parameters of the knowledge representative structure are trained. Different ML paradigms support different representations of knowledge, and adopt different learning methods. In neural network algorithms, knowledge is represented as a graph consisting of nodes and edges, and, in rule induction, condition-action rules are used. In other methods, functions, logic programs and rule sets, finite-state machines, grammars, and problem solving systems have been adopted to represent knowledge (Nilsson, 1996).

Concept description is the realization of what a ML model learns from the set of examples used as the target concept. Thus, it can be different due to the type of knowledge representation and training examples collected.

2.4.3 Machine Learning and its Use in TC Applications

ML research is a multi-disciplinary field, influenced by various disciplines such as artificial intelligence, computational complexity theory, information theory, statistics, control theory, psychology, and philosophy (Mitchell, 1996). Due to their multi-disciplinary nature, ML techniques have been applied to a wide range of tasks with a certain level of success, including but not limited to IR and document management applications (Belew, 1989; Blosseville *et al.*, 1992; Gordon, 1988), speech recognition and natural language processing (Charniak, 1993; Waibel *et al.*, 1989), pattern recognition and image processing (Duda *et al.*, 2000), game playing (Tesauro, 1995), bioinformatics and epidemiology (Cooper *et al.*, 1997), cognitive science and computational learning theory (Natarajan, 1991) and autonomous performance and control applications (Pomerleau, 1989).

The ML paradigm has been known as a framework for learning knowledge from experience (Langley, 1996). Requiring experience, available from digital documents, it becomes rapidly popular in text-related tasks, such as TC and other IR-related fields. It turns out that the inductive ML techniques from neural networks, symbolic learning, and genetic algorithms have been commonly used on IR applications including TC (Chen, 1995).

ML techniques have been applied to various TC contexts: Classification of in-patient discharge summaries (Larkey & Croft, 1996), flora data (Cui *et al.*, 2002), legal document analysis and classification such as case law (Lame, 2001; Pannu, 1995; Schweighofer & Merkl, 1999; Thompson, 2001), and patent document classification (Larkey, 1998). The automatic classification of Web documents is a new challenge as numerous heterogeneous features in authorships, vocabularies in use, internal structures, and types of objects are coalesced (Chakrabarti *et al.*, 1998). Various ML techniques used in TC application such as SVM (Joachims, 1997), RIPPER (Cohen, 1995), Quinlan's FOIL (Quinlan, 1990), k-NN (Yang, 1997), and NB (Lewis & Ringuette, 1994) have been used in automatic Web document classification contexts (Craven & Slattery, 2001; Dumais & Chen, 2002; Sun *et al.*, 2002; Yang *et al.*, 2002). The NB algorithm among different ML methods is the most popular for the task of the Web classification, due to its efficiency (Oh *et al.*, 2000) and its utilization as a strong baseline algorithm in text classification (Yang *et al.*, 2002).

Subject or topic has been a standard type of class for classification in TC as well as other classification works. However, other types also have been considered for TC applications. *Genre* has been used as a central element to classify documents in some TC research (Kessler *et al.*, 1997; Stamatatos, 2000). Lee & Myaeng (2002) presents a method of statistical features (term frequency and inversed document frequency) for automatic genre detection using two ML algorithms (Naïve Bayesian and similarity-based algorithm). In their experiment, seven genres - *editorial*, *report*, *review*, *research paper*, *homepage*, *question & answer*, and *product specification* - are selected and tested on some web documents, and show the superior performance of the statistical based method. The task of *essay grading* has been a challenging research topic in the fields of linguistics and artificial intelligence (Landauer *et al.*, 1997). Larkey (1998) introduces a Bayesian independence classifier for the grading task, tests the model on documents on various subjects including social science, physics, and law, and shows the performance comparable to human graders. The increasing number of spam mail was recognized as a significant problem due to its high volume and heavy burden on network traffic and computer servers (Campbell, 1994; Cranor & LaMacchia 1998; Gwynne & Dickerson,

1997). The first work on spam mail filters can be found in (Cohen, 1996), where an extension of the rule learning algorithm RIPPER (Cohen, 1995) was used and trained using eleven mail folders from three different persons' real mail folders, each folder consisting of spam and legitimate mails. The RIPPER rule-based algorithm was then compared to a term frequency (TF) and inversed document frequency (IDF) weighting based filter and showed its superior performance. After that, other ML algorithms were also involved in building an anti-spam mail filtering system: Decision Trees (Carreras & Márquez, 2001; Hidalgo *et al.*, 2000), k-NN (Androutsopoulos *et al.*, 2000; Hidalgo *et al.*, 2000), NB (Rennie, 2000; Sahami *et al.*, 1998; Schneider, 2003), Stacking (Sakkis *et al.*, 2001), SVM (Drucker *et al.*, 1999; Kolcz & Alspector, 2001), and Rocchio (Drucker *et al.*, 1999; Hidalgo, 2002).

As classes of interest in TC are often organized in a hierarchical structure, such as Yahoo, the US patent database, and library classification systems, the use of hierarchical structures has been considered in TC research for improving classification accuracy. Koller & Sahami (1997) utilize the hierarchical structure to decompose a classification task covering a large number of classes into several classification subtasks each of which deals with a much smaller number of classes. The decomposed tasks lead to having a smaller number of features for classes. They show that the classification accuracy with only 10 or 20 features outperform those with full features of 1,258 and 283. In McCallum *et al.*'s work (1998), a statistical technique called *shrinkage* is applied to smooth parameter estimations for child-node classes with the help of parent-node classes. In tests with Web documents and newsgroup data, they show a reduction in error of about 30%, when compared to classifiers for non-hierarchical structures. More recently, Dumais & Chen (2000) explore the use of hierarchical structure in a similar approach to that of Koller & Sahani, but the former apply a binary feature version of the SVM classification model in the hierarchical context. The experimental results report that hierarchical models achieve a 4% increase in overall F1¹¹ metric for classification accuracy over non-hierarchical models.

¹¹ $F1 = 2 \times \text{precision} \times \text{recall} / (\text{precision} + \text{recall})$ (van Rijsbergen, 1979).

2.5 Automated Classification and Library Classification Systems

2.5.1 Introduction

Remember that library classification systems, such as LCC and DDC, have long been developed and used for organizing and arranging collections in libraries. Recently, as the need for tools to organize and manage digital information has emerged, the use of library classification systems has been explored as a classification framework for digital information. However, this did not occur until the machine-readable form of DDC or LCC became available at the end of the 1980s and the early 1990s, respectively (Finni, 1987; Guenther, 1992). One of the pioneer works on automated classification based on a library classification system can be found in Larson's work (1992), where he attempted to classify a set of MARC records into LCC, based on title and subject headings, by creating clustering vectors for the LC subclasses of interest. In the experiments, 5,765 clusters were created for LC class Z with 30,471 MARC records from the University of California's Melvyl online cataloging system and 286 MARC records were tested over the Z class. Various performance results were reported based on different combinations of title and subject headings, and different similarity methods. It was concluded that his method failed to prove its effectiveness sufficiently to provide a relevant classification number for an item. However, it was speculated that this work would help librarians to determine relevant classification numbers for unclassified items by providing a list of potential classification numbers based on subject headings and titles. The most recent work directly linked to Larson's work can be found in Frank and Paynter's paper (2004). Their work aims to assign LCC to metadata of Internet resources using LCC and LCSH. The classifier is trained using 800,000 library catalog records and tested on an independent set of 50,000 records. Unlike Larson, who used the IR techniques from the SMART system (Salton, 1971), this work uses Sequential Minimal Optimization (Platt, 1998) to calculate similarity measures between classes and the documents to be classified. The classification accuracy of this classifier is reported to be from 55% to 80% approximately, when the predictions for the top N (equivalent to 1, 2, 5, 10, and 15) are considered. In the following sections, some classification-related research projects where

traditional library classification schemes were adopted as the basis for a classification system for digital documents will be reviewed.

2.5.2 Pharos

Pharos (Dolin *et al.*, 1997; Dolin *et al.*, 1998; Dolin *et al.*, 1999) is an information architecture prototype accommodating heterogeneous sources in content and format, derived from the Alexandria Digital Library project (Andresen *et al.*, 1996). As an initial prototype of the Pharos architecture, an automatic classification system based on the LCC was implemented for the purpose of creating the profiles of heterogeneous digital information.

In this project, the Latent Semantic Indexing (LSI) was applied for automatically classifying newsgroups within the LCC. LSI is an approach for modelling the underlying (latent) structure of term associations by applying a matrix computation technique called singular value decomposition (SVD) (Deerwester *et al.*, 1990). This approach is often compared with Salton's vector space model (VSM) (Salton, 1971), since both approaches start with a term-by-document matrix representing documents. In LSI, the original term-by-document matrix is decomposed by the SVD method and a set of latent semantic factors are determined by choosing top K singular values. The selected LSI factors play the role of new coordinates for a new vector space for the representation of terms and documents. Deerwester *et al.* (1990) describe the three advantages of using LSI factor-based representation, compared to the VSM assuming term independence: synonymy, polysemy, and term dependence. The storage and computational costs for matrix decomposition, however, are pointed out as some of the drawbacks of LSI (Hull, 1994).

As a training data set for the LSI-based IR system, 1.5 million catalogue records from the University of California Santa Barbara library were used, and title, subject headings, and LCC fields from the records were extracted. For a specific holding, title and subject heading data are viewed as a description for a specific category denoted by a LCC number. Such a relationship between a LCC number and its descriptors forms training data for the classification system. Once a LSI-based IR system was developed, the relevance of an article to each subject category of LCC was measured, and relevant scores

beyond a certain threshold were only considered as contributing factors to a category of an article. In this way, all the articles in a newsgroup were treated, and summing up all the contributing factors from all the articles formed a profile of a newsgroup over LCC. In Dolin's PhD dissertation (1998), 7214 MARC records from the 21 major classes of LCC were classified, and the experimental results yielded an average median of 13.0 ± 3.9 and an average mean of 76 ± 19 for about 4,200 LC classes. In another experiment with articles from 2,500 Usenet newsgroups, the classification accuracy for the experiment is not reported since articles that were not pre-classified were involved.

2.5.3 Scorpion

Scorpion¹² was a research project conducted by OCLC¹³ from 1996 to 1999, addressing "*the challenge of applying classification schemes and subject headings cost effectively to electronic information*" (Thompson *et al.*, 1997, p. 1). The purpose of the project was to develop an automated method of recognizing the subject categories of digital documents, according to the DDC (Shafer *et al.*, 1999), to help human catalogers create catalog entries for the increasing amounts of electronic resources.

For its automatic classification internal mechanism, Scorpion relies on a clustering method (Subramanian & Shafer, 1998). Clustering is one of the techniques used to group similar objects by measuring the similarity between two objects. DDC has been maintained through the Editorial Support System (ESS) at OCLC and Forest Press, and EES records contain all the information needed to produce DDC schedules and tables. By applying the clustering algorithm to EES records, a set of conceptual clusters for EES data is pre-determined. Given an input document, Scorpion measures similarities between the input and the pre-defined clusters and considers the nearest cluster as the most probable place for the input document. A function of term frequency is used as a

¹² <http://www.purl.oclc.org/scorpion>

¹³ The Online Computer Library Center, Inc., founded in 1967, is a major bibliographic utility and nonprofit membership organization serving more than 45,000 libraries in 84 countries and territories around the world providing bibliographic resources and services to its member libraries. Source: <http://www.oclc.org/about/default.htm> visited on 10 April 2004.

measurement of similarity in Scorpion. However, more detailed procedures, such as the fields of ESS records and pre-processing, are not published in its reports.

For application, the Scorpion system (Shafer, 1997b) takes online documents to be classified as database queries to the system, a database of Dewey numbers with their descriptions, and returns a ranked list of the potential DDC classes relevant to the documents as a result. A couple of papers were released to report an evaluation of the Scorpion system (Shafer, 1998; Shafer *et al.*, 1999). For the evaluation, a collection of bibliographic records for Internet resources in which DDC classes were human-assigned was used. Unfortunately, however, detailed experimental results were not unveiled, presumably because a comparison could not be properly done because the human-assignment of DDC classes was based solely on phrases describing Internet resources. Their conclusions confirmed Scorpion's early expectation (Shafer, 1997a) that automatic classification cannot replace manual classification, but that it can provide a cost effective solution to support human catalogers.

2.5.4 DESIRE

The DESIRE¹⁴ project, started in 1996, is a large-scale international project funded by the European Union including many researchers from four countries. The Web page¹⁵ for the project explains that its purpose is 'to build large scale information networks for the research community.' This international project aiming to develop a high quality research information database has been conducted in two phases: DESIRE I & II. The objective of DESIRE I is to combine a high quality of subject-based information resources supported by subject experts and exhaustive resources on the same subjects acquired by automated Web crawlers. Enhancements to the first phase have been done in DESIRE II. This project, performed in collaboration with OCLC in the adoption of automatic classification methods, was used in the Scorpion project.

Prototype research to establish a subject gateway for engineering-subject resources was conducted in DESIRE II. In the experimental research, the EI thesaurus, containing

¹⁴ <http://www.desire.org>

¹⁵ <http://www.desire.org/html/aboutus/aboutus.html> visited on 27 August 2003

more than 800 engineering classification categories, was used for document classification as well as browsing-based search services, and a set of Web pages, whose categories were pre-assigned by human experts on the subjects, were automatically classified according to the EI classification structure. The automatic classification of assigning the EI categories to a Web document relies on a simple term matching algorithm. The description of Web documents such as metadata, headings, and plain text are matches against the terms from the EI thesaurus representing an EI classification category, where other heuristic factors including term complexity, type of classification, match location, and match frequency are tested. In the system's evaluation with approximately 1000 Web pages, the automatic classification's accuracy was compared to the classification staffs' decisions. Overall, the fact that about 60% of the automatic classifications were correctly or more finely matched to the human decisions was reported¹⁶. With the collaboration of OCLC's Knowledge Organization group, the same engineering database was classified with DDC. In this case, LC Subject Headings were added to the terms representing classification categories, and the words and phrases from documents, rather than the full-text, were intended to be matched. A comprehensive report on the DDC-based classification has not yet been published and a more detailed procedure has not been reported.

2.5.5 Wolverhampton Web Library (WWLib)

WWLib¹⁷ is a Web search engine project for UK-based documents, where DDC is used to organize the collected documents. An interesting feature of the experimental WWLib is to treat a Web page as an item in a library and to prepare cataloging records describing information including the title, Universal Resource Locator (URL), DDC category, and description to the collected Web pages. In general, Web search engines present results in the order of relevance to users' requests, whereas the WWLib provides the relevant Web pages in terms of DDC category.

¹⁶ <http://www.lub.lu.se/desire/DESIRE36a-overview.html>

¹⁷ <http://www.scit.wlv.ac.uk/wwlib>

To serve as a search engine, the WWLib architecture contains similar components to a general search engine: (1) Spider – to retrieve documents from the Web, (2) Indexer – to store into a database the contents and metadata generated, (3) Analyzer – to analyze the retrieved Web documents for URL contained information, (4) Classifier – to assign DDC categories to the documents, (5) Builder – to analyze the contents of the documents and produce their metadata, and (6) Searcher – to allow users to search this WWLib database. The Classifier component performs the process of classifying Web documents automatically, which relies on simple word matching. The classifier in the WWLib compares a stream of words extracted from documents and the description of DDC categories (Wallis & Burden, 1995). The words occurring in Web documents are *weighted* according to the tags used for them, and a stemming technique is applied. Also, to take advantage of the hierarchical structure of DDC, a method for the relevance of a document to both a class and its upper class is taken into consideration. In the later version (WWLib), a more rich set of description for DDC classes including synonyms is considered. A formal experiment for the measurement of the system's performance seems not to be undertaken. Instead, an informal testing result was reported with the randomly selected 17 URLs (WWLib); where 13 cases out of 17 were simply reported to be relevant without divulging more detailed procedures such as evaluation methods and data selection.

2.6 Summary and Conclusion

This chapter has reviewed the works on automated classification and the use of library classifications. The automated classification work dates back to the early 1960s. Until the 1980s, *clustering* was the main focus of work on automated classification, by grouping similar documents rather than using classes as references for classification. In this approach, clustering and IR-related techniques were mainly adopted to measure the similarities between documents for the clustering. Since the 1990s, the ML-based approach has settled in as a major stream in automated classification. In comparison to the clustering approach, a pre-defined set of classes, depending on the classification task, are introduced, and the work of classification may be viewed as the assignment of classes to

documents. Thus, the ML approach to automated classification and its principles are covered in this chapter.

The introduction of library classifications into automated classification tasks has attracted TC researchers due to their need of organizing documents from various disciplines. This chapter has covered the classification applications using library classifications that appeared in some of the major research work and research projects. From the standpoint of application, most research projects (Doline *et al.*, 1999; Frank & Paynter 2004; Larson, 1992; Shafer, 1997b) for their task have been limited to the classification of metadata - cataloging records – in the design and testing of their classification systems. Even in the DESIRE and WWLib projects, which were designed for the classification of Web documents and resources, a careful comprehensive experiment was not explored or reported. Though a number of research projects were performed, the task has not yet been applied for full-text documents. Thus, the work in this present study may contribute to fill a gap.

IR techniques were applied in most TC projects described above as major techniques for automatic classification, including the term reduction technique by LSI, similarity measurement between documents via the clustering method, Salton's VSM, and term weighting based on term and document frequencies. In addition, the use of a machine learning technique called a linear SVM was recently reported (Frank & Paynter, 2004) in the project of automatically classifying LCSHs into LCCs.

From the viewpoint of the classification scheme, either LCC or DDC, both of which are popular library classifications used in most research and academic libraries in North America, have been used in classification applications. These two library classifications were designed for the same purpose, but have their own strengths and weaknesses. However, the one rationale behind the choice of a library classification seems to lie on the availability of the data used. There are three reasons for this claim. First, the reasons specified in research articles for their choices are quite general enough to be applicable to other classifications, and are not specific enough to be applicable to the selected classification. For example, an article on the WWLib project states the reason for using DDC as follows “*DDC is a universal classification scheme covering all subject areas and*

geographically global information” (Jenkins *et al.*, 1998, p. 1), although the point also can be applicable to LCC or other library classifications. Second, research on automated classification tasks focuses on the classification accuracy of an implemented classifier and this classification accuracy is not affected by the choice of a classification scheme, but may be affected by the data needed for building TC systems, especially in the machine learning approach. In conclusion, judging from a review of the reasons written in research articles for the choice of a library classification scheme, the decision seems to be based upon personal preference rather than the tasks at hand or the availability of data. This is due to the advantages and disadvantages of different classification schemes not being successfully applicable to TC systems users.

In this study, LCC is used as a classification scheme for automated classification, rather than other library classification schemes. Regardless of the claim specified above, a few reliable facts that drive this study to favor LCC are: (1) LCC and LCSH notations can be found in almost all MARC records, (2) LCC and LCSH have been maintained by the same authoritative institution, the Library of Congress, which provides reliable, high-quality services, and (3) LCC has been used by most large academic and research libraries. In addition, there is a practical issue related to the choice: the availability of the classified documents. In general, it is hard to get labeled data (classified data) in machine learning techniques. The same phenomenon is applied to this case. The set of data required for building the system (LCC and LCSH, and their relationships) is available in most MARC records, rather than DDC and LCSH.

Chapter 3

HIDDEN MARKOV MODEL AND ITS APPLICATIONS TO TC

3.1 Hidden Markov Model

The Markov model proposed by Andrei A. Markov in the early 1900s (Markov, 1913) is often called an observable (or visible) Markov model, so as to be distinguished from the hidden Markov model. Since the HMM is derived from traditional Markov theory, this chapter begin with a brief description for a Markov model.

A Markov model is a statistical model, and can be viewed as a typical transitional diagram composed of the following components: a set of different states; transitions between states; and transition probabilities, which are probabilities linked to transitions. Beginning from a *start* state, a transitional process continues until it reaches an *end* state. The outcome of this sequence of states (observations) within a Markov model is called a Markov chain. Formally, a Markov chain can be represented by a series of random variables, $X^{(0)}, X^{(1)}, X^{(2)}, \dots, X^{(m+1)}$, each of which takes on a value from the state space, $S = \{s_1, s_2, \dots, s_N\}$.

The main characteristic of a Markov model is that it predicts the future based on the present rather than the past; this is termed the *Markov condition*. When the probability of a sequence of random variables, $P(X^{(0)}, \dots, X^{(m+1)})$ is calculated, all the previous random variables are considered according to:

$$P(X^{(0)}, \dots, X^{(m+1)}) = P(X^{(0)})P(X^{(1)} | X^{(0)}) \dots P(X^{(m)} | X^{(0)}, \dots, X^{(m-1)})P(X^{(m+1)} | X^{(0)}, \dots, X^{(m)}).$$

By assuming the *Markov condition*, however, Markov theory only takes into consideration the current random variable by:

$$P(X^{(0)}, \dots, X^{(m+1)}) = P(X^{(0)})P(X^{(1)} | X^{(0)}) \dots P(X^{(m)} | X^{(m-1)})P(X^{(m+1)} | X^{(m)}).$$

To present the Markov model's features a simple example will be provided. Figure 3.1 shows an example of a simple Markov model for tossing a coin. Each state of the model corresponds to the outcome of an observation (also called an *event*), either Heads (H) or Tails (T) of a coin. The numbers along the lines or curves specify transition probabilities linked to a directed path between the two states connected by the lines or curves. The arrows indicate the direction of the state transition. The *S* in the figure shows the special state serving as the starting point in the model. This model must start from the special state and can move to other states by adding the probabilities corresponding to the direction that it takes.

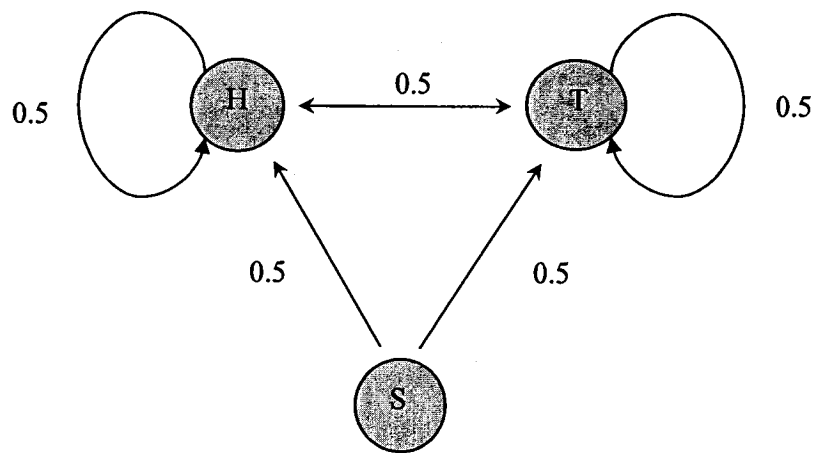


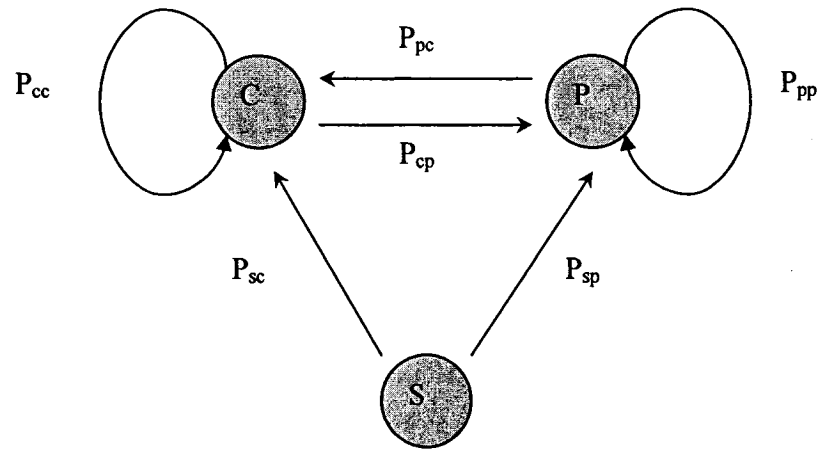
Figure 3.1: A Markov model for tossing one coin

Given the diagram above, an interesting question might be raised for a way of obtaining the probability of a sequence of observations. For example, given $S = \{H, T, T, H\}$, the probability of the sequence of the observations can be expressed in terms of the current model M :

$$\begin{aligned}
 P(S / M) &= P(H, T, T, H / M) \\
 &= P(H)P(T / H)P(T / H, T)P(H / H, T, T) \\
 &= P(H)P(T / H)P(T / T)P(H / T) \text{ by the Markov condition} \\
 &= (0.5)^4 = 0.0625
 \end{aligned}$$

So far the (observable) Markov model and its characteristics along with a simple example have been reviewed. A hidden Markov model is the extension of a Markov model, and differs from an observable Markov model in that a sequence of states corresponding to a list of observations is not immediately observable. In a Markov model, an observation is represented as a state of the model, whereas, in a hidden Markov model, it is viewed as a probability function of a state, not a state itself. Therefore, given an observation sequence, the corresponding path (a sequence of states) is uniquely obtained with a Markov model, whereas a unique corresponding path is not recovered with a HMM.

Let us consider a beverage vending machine to clarify the invisible aspect of a HMM. It is assumed that there are two preferable positions in the machine - Coke and Pepsi - and that these two positions are randomly changed. When the machine is utilized, a user does not have any information on the current machine position. Figure 3.2 shows a HMM describing the vending machine, where the C state points out the Coke preferable state and the P state specifies the Pepsi preferable state. These Coke and Pepsi events are associated with their emission probabilities in each state, rather than states themselves as they would be in a Markov model. In Figure 3.2, the emission probabilities for the state C are $P_c(c)$ for the probability of producing a Coke in the state and $P_c(p)$ for producing a Pepsi.



P_{sc} : Probability of C state from S state

P_{sp} : Probability of P state from S state

P_{pc} : Probability of Coke from P state

P_{pp} : Probability of Pepsi from P state

P_{cc} : Probability of Coke from C state

P_{cp} : Probability of Pepsi from C state

Figure 3.2: A hidden Markov model for a beverage vending machine

For example, given the beverage machine M_h , the probability of seeing the output sequence, $S = \{Coke, Pepsi\}$ can be calculated.

$$\begin{aligned}
 P(S / M_h) &= P(Coke, Pepsi / M_h) \\
 &= P(Coke)P(Pepsi / Coke) \\
 &= (P_{sc}P_c(c)P_{cc}P_c(p)) + (P_{sc}P_c(c)P_{cp}P_p(p)) + \\
 &\quad (P_{sp}P_p(c)P_{pp}P_p(p)) + (P_{sp}P_p(c)P_{pc}P_c(p))
 \end{aligned}$$

where $P_i(j)$ is the probability that the observation j occurs at the state i . As illustrated above, given the observation sequence, all the possible paths are searched and evaluated in the process of acquiring the probability of the observations. In practice, however, taking all the possible paths for a sequence of observations into consideration is impossible due to its infeasible running time. Since the computational time for the possible paths grows as N^T , where N is the number of states in the model and T is the number of observations, even with a small number of states and observations, this type of calculation is not feasible. For instance, the computational time for 20 observations with a simple model consisting of 3 states is 3^{20} (approximately 3,500,000,000), which is an extremely large number for any practical consideration. In the following section, the HMM will be defined formally.

3.2 Components of A Hidden Markov Model

A HMM can be characterized by the following components:

- N , the number of different states: In the early stages of model design, the most important consideration is to decide what the structure of the model will be, including what states will represent and the number of states included in the model. In the previous section's example, HMM's states for a vending machine illustrate the machine's different preferable states. The set of states in a model can be denoted by $S = (S_1, S_2, \dots, S_N)$, where there are N states in a model. The symbol q' denotes a state at time t . The HMM of the vending machine always starts at the special state S at time 0. When the transition from state S to state C is

made, the state C is said to be the current state at time 1. In this case, $q^0 = S$ and $q^1 = C$.

- **M**, the number of distinctive observations (outputs or symbols): The output of a model is a sequence of observations. The observations of a model are represented by $V = (V_1, V_2, \dots, V_M)$. In the HMM for the vending machine, there are two different observations, *Coke* and *Pepsi*.
- **Π** , initial state probabilities: $\Pi = \{\pi_i\}, i \in S$, and $\pi_i = P[q^1 = S_i]$ which is the probability of being at the state S_i at time 1. The initial state probabilities are the transition probabilities from a special start state to other states of a model. Thus, the states linked directly with a special starting state are associated with the initial state probabilities. In the example above, the probabilities (P_{sc} and P_{sp}) along the links from state S to the other states are the initial state probabilities of the model.
- **A**, state transition probabilities: $A = \{a_{ij}\}$, where $i, j \in S$, and $a_{ij} = P[q^{t+1} = S_j | q^t = S_i]$, which is the probability of being at state j at time $t+1$ and at state i at time t . The state transition probabilities are the probabilities for the transitions between the states in a model, including that of self-transition to the same state. The summation of the outgoing transition probabilities from a state should always be 1. In the HMM example, the summation of the outgoing transition probabilities from the state C or P should be 1: $P_{cc} + P_{cp} = 1$ and $P_{pc} + P_{pp} = 1$.
- **B**, observation symbol emission probabilities: $B = \{b_i(k)\}$, where $i \in S, k \in V$ and $b_i(k) = P[V_k | q^t = S_i]$, which is the probability of emitting the symbol V_k at state j at time t . An emission probability is the probability that an observation symbol occurs in a state. It is clear that the emission probability of a symbol in a state is different from that of the same symbol in a different state. As the example shows,

the emission probabilities for the symbol Coke are $P_c(c)$ in state C and $P_p(c)$ in state P , and those for the symbol Pepsi are $P_c(p)$ in state C and $P_p(p)$ in state P .

In summary, a HMM, λ , is defined by the three different sets of probabilities as follows:
 $\lambda = (A, B, \Pi)$.

3.3 Three Fundamental Problems for the Hidden Markov Model

There are three fundamental problems related to the design and implementation of a HMM. The solutions to these quandaries are known and should, therefore be implemented when applying the HMM:

Problem 1: Given a model $\lambda = (A, B, \Pi)$ and a sequence of observations $O = (O_1, O_2, \dots, O_t)$, how can the probability of an observation sequence, $P(O | \lambda)$, be estimated?

By using the solution to this problem, the best model to describe the observation sequence can be estimated. Since the probability, $P(O | \lambda)$, connotes how well the model represents the given observation as a result of the process of obtaining probabilities, the model producing the highest probability given a sequence of observations is selected as the best-matching model.

Problem 2: Given a model $\lambda = (A, B, \Pi)$ and a sequence of observations $O = (O_1, O_2, \dots, O_t)$, how is a sequence of states $Q = (q^1, q^2, \dots, q^t)$ that optimally describes an observation sequence determined?

The answer to this problem is applied when state paths are the goal that this model attempts to represent. The difficulty for the solution lays in the fact that the number of all possible paths is intractable. A commonly used, dynamic, programming-based algorithm was developed to show this problem.

Problem 3: Given a sequence of observations $O = (O_1, O_2, \dots, O_t)$, how can model parameters $\lambda = (A, B, \Pi)$ best describing the observation sequence be estimated?

This question is equivalent to the task tackled during the training process of a model. The goal of training a model is to obtain a model that extracts its statistical properties from a given set of training data and represents them. One of the major concerns at this stage is to be cautious of over-training or under-training a model. Particularly, if a model is over-trained, which means that a model reflects training data too perfectly, the model lacks the capability of being predictable.

3.3.1 The First Problem: Obtaining the Probability of an Observation Sequence

Given a sequence of observations $O = (O_1, O_2, \dots, O_t)$, and a model $\lambda = (A, B, \Pi)$, the calculation of the probability $P(O | \lambda)$ is needed. There is a straightforward method of calculating the probability. For a sequence of states $S = (S_1, S_2, \dots, S_t)$, the probability of the observation sequence O is:

$$P(O | S, \lambda) = \prod_{i=1}^t b_i(O_i).$$

The probability of a state sequence can be written as:

$$P(S | \lambda) = \pi_1 a_{12} a_{23} \dots a_{(t-1)t} = \pi_1 \prod_{i=1}^{t-1} a_{i(i+1)}.$$

Thus, the probability of the observation sequence O along a fixed path S of states is:

$$P(O, S | \lambda) = P(O | S, \lambda) P(S | \lambda).$$

Therefore, the probability of the observation sequence O in a model is:

$$P(O | \lambda) = \sum_{S \in Z} P(O | S, \lambda) P(S | \lambda),$$

where Z is a set of all the possible combinations of states of length t . The computational analysis for the direct algorithm is as follows. To obtain the observation probability for a path sequence, $T-1$ multiplications for the consideration of the state sequences and T multiplications for output emission are required, and thus a total of $2T-1$ are needed for a state-sequence path of length t . There are N states in the model. Since there are N possible choices for a state, the number of possible state sequences is N^T . In summary, the described algorithm requires $(2T-1)(N^T)$ multiplications, which is logistically infeasible, even with small numbers for N and T . There is a much more efficient method of calculating the probability of a sequence of observation, based on a dynamic algorithm, which is referred to as *the forward-backward algorithm* (Baum & Egon, 1967; Baum & Sell, 1968).

The Forward Procedure

A forward variable $\alpha_t(i)$ is defined as $\alpha_t(i) = P(O_1, O_2, \dots, O_t, q^t = S_i | \lambda)$. The definition says that a forward variable $\alpha_t(i)$ is the probability of generating the sequence $O = (O_1, O_2, \dots, O_t)$ and being at state S_j at time t , given the model λ . The inductive steps to obtain a forward variable $\alpha_t(i)$ are as follows:

1. Initialization:

$$\alpha_1(i) = \pi b_i(O_1), \quad 1 \leq i \leq N.$$

2. Induction:

$$\alpha_t(j) = \left[\sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(O_t), \quad 2 \leq t \leq T, \quad 1 \leq j \leq N.$$

3. Final:

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i).$$

At step 1, *Initialization* of the induction, for the symbol O_1 , the initial state probabilities to each state and the emission probabilities associated with the initial state are pre-calculated and stored. Figure 3.3 illustrates the procedure for step 2, *Induction*. Consider the forward variable $\alpha_t(j)$ at time t as shown in Figure 3.3. Since any state could be the previous state of the state S_j when the forward variable of S_j is considered, the state transition probabilities from all the previous states are summed up. At this point, it is assumed by the rule of induction that for any i , $\alpha_{t-1}(i)$ at time $t-1$ is already calculated. Since the emission probability $b_j(O_t)$ for O_t is associated with the state S_j , $b_j(O_t)$ is multiplied with the summation of the probabilities from all the paths incoming to the state S_j , to acquire the forward variable $\alpha_t(j)$ at time t in the induction step. Finally, as shown in step 3, the computation of the probability of a sequence of observations is performed through the summation of all the forward variables ending at each state.

Figure 3.4 graphically illustrates the induction step of the forward procedure in the computation of $P(O | \lambda)$. During step 2 of the induction process, $N(N+1)$ multiplications and $N(N-1)$ additions are required for the calculation of a forward variable to an observation. Since there are T observations in the sequence, $T-1$ times of multiplications and additions are multiplied. Also, N times of multiplications for step 1 and $N-1$ times of additions for step 3 are added to the computation time total. Putting them all together, $N(N+1)(T-1)+N$ multiplications and $N(N-1)(T-1)+(N-1)$ additions are needed. Therefore, the forward procedure for the calculation of $P(O | \lambda)$ has a running time of $\Theta(N^2T)$ ¹⁸.

¹⁸ $\Theta(g(n))$ is a set of functions $f(n)$ satisfying that there exist positive constants c_1, c_2 , and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$ (Cormen et al., 2001).

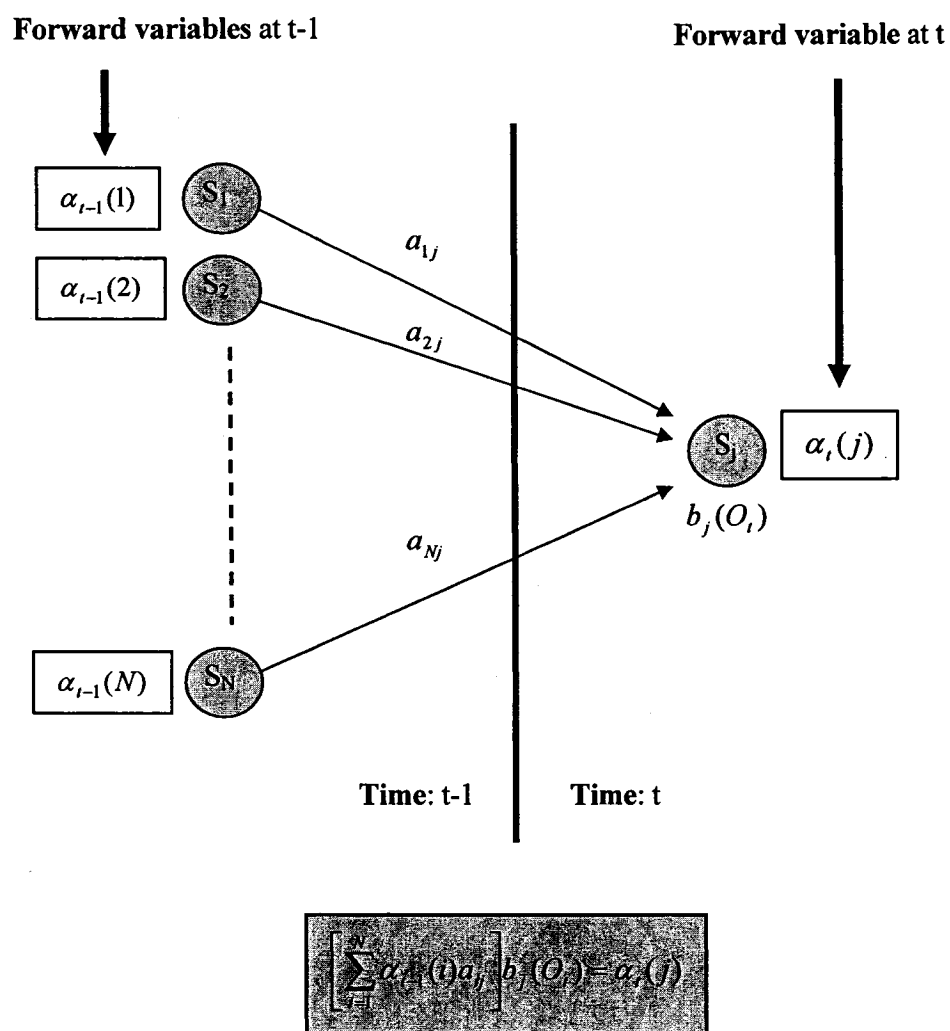


Figure 3.3: Interpretation of the induction step of the forward procedure

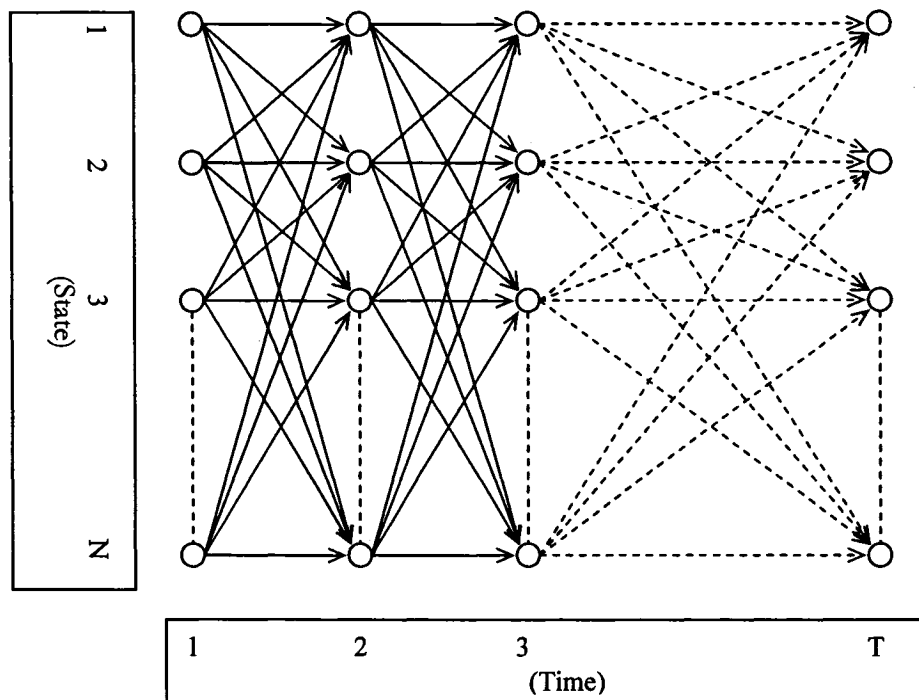


Figure 3.4: A dynamic algorithm procedure for the calculation of a forward variable

The Backward Procedure

Both of the forward and backward algorithms work in a similar way, as derived from a dynamic programming technique, except that they approach the same goal from opposite directions. Due to this fact, the backward procedure will be described here, though it will be employed in solving the third problem of parameter re-estimation.

A backward variable $\beta(i)$ is defined as $\beta(i) = P(O_{(t+1)}, O_{(t+2)}, \dots, O_T, q_t = S_i | \lambda)$, which is interpreted as the probability of starting at state S_i at time t and producing the observation sequence from O_{t+1} to the end, given the model λ . The induction procedure for a backward variable $\alpha(i)$ is as follows:

1. Initialization:

$$\beta_t(i) = 1, \quad 1 \leq i \leq N.$$

2. Induction:

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad 1 \leq t \leq T-1, \quad 1 \leq i \leq N.$$

3. Final:

$$P(O | \mu) = \sum_{i=1}^N \Pi_i \beta_1(i).$$

Two different characteristics are considered in the comparison of the forward and backward algorithms. First, they are distinct in the direction of the variable calculation. When backward variables are approximated, the backward probability of a state adopts those of its following states. However, the calculation of forward variables in a state uses the forward probabilities of its previous states. Figure 3.3 and 3.5 display the relationship of two consecutive states in estimating forward and backward variables. Second, in the backward procedure, the symbol emission probability occurs at the pre-calculated state that is the *following* state of the current state, whereas in the *forward* algorithm, it derives from the current state that is not pre-calculated.

As the similarity of the two algorithms might indicate, the *backward* algorithm takes almost the same computational time as the *forward*. In the second step of induction, $2N^2(T-1)$ multiplications and $(N-1)N(T-1)$ additions are needed. The N multiplications and $(N-1)$ additions are additionally required for step 3. In total, the order of the running time for the backward algorithm is $\Theta(N^2T)$ which is the same as that of the *forward*. The *forward* and *backward* variables lead to the same result when they are applied to the same model. It is simply a matter of order.

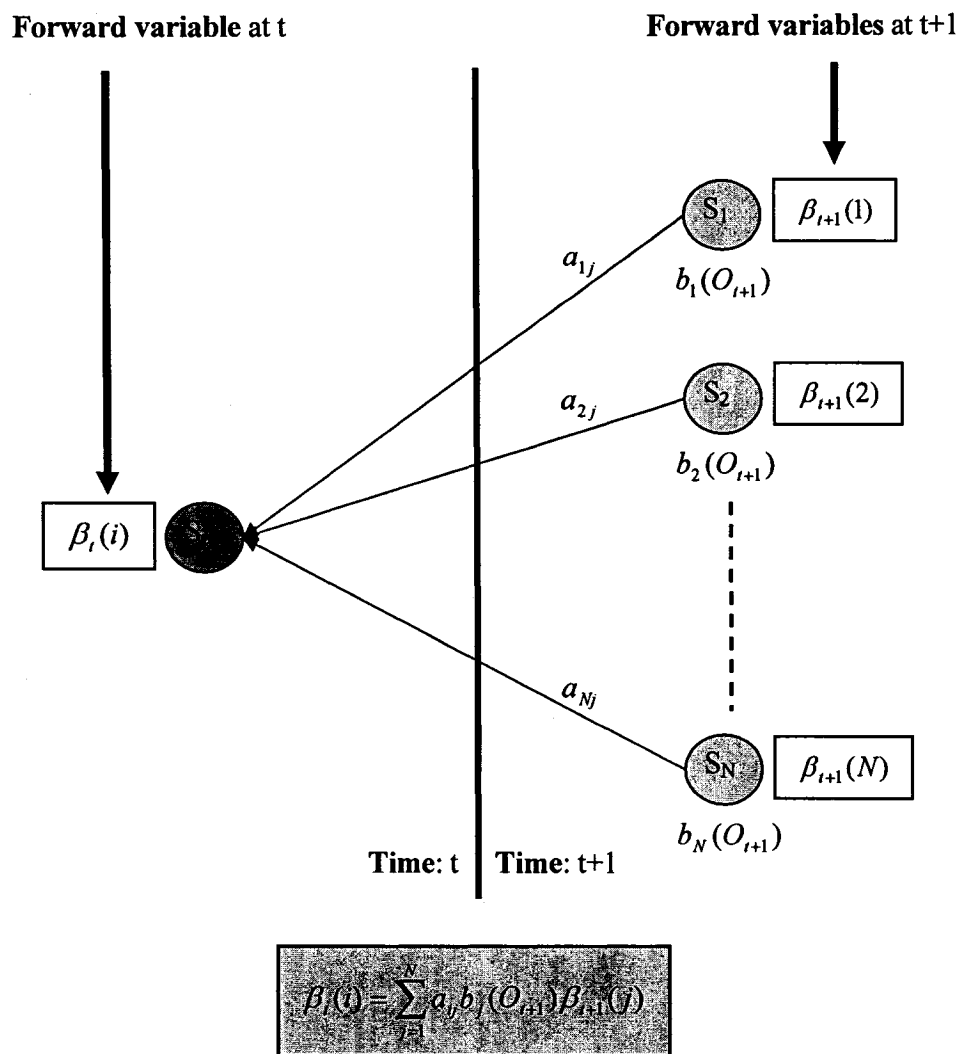


Figure 3.5: Interpretation of the induction step of the backward procedure

3.3.2 The Second Problem: Finding an Optimal Path

The second problem for finding an optimal path is described here: Given a model $\lambda = (A, B, \Pi)$ and a sequence of observation $O = (O_1, O_2, \dots, O_T)$, how is a sequence of states $Q = (q_1, q_2, \dots, q_T)$ optimally describing the observation sequence found? What does it mean that an ordered list of states optimally describes a sequence of observations? It is interpretable as follows. Given a HMM model, many different paths of states can be considered to generate the same observations. Since a probability is assigned to a specific path of states for a list of given observations, the goal of this problem is to find a path of states, which is associated with a maximum probability.

The Viterbi algorithm (Viterbi, 1967; Forney, 1973) is known as a technique to find a best single state sequence for an observation sequence, that is, to maximize $P(Q | O, \lambda)$. Generally, $P(Q, O | \lambda) = P(Q | O, \lambda)P(O | \lambda)$. However, for a fixed observation sequence O , $P(Q, O | \lambda) = P(Q | O, \lambda)$. Before explaining the Viterbi algorithm further, let's define two variables given the observation sequence $O = (O_1, O_2, \dots, O_T)$, $\delta_t(i)$ indicating the highest probability of being at state i after the first t observations, and $\psi_t(i)$ representing a sequence of states with the highest probability of being at state i after the first t observations. Formally, the two variables are defined as follows:

$$\psi_t(i) = \arg \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_{t-1}, q_t = i, O_1, O_2, \dots, O_t | \lambda),$$

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_{t-1}, q_t = i, O_1, O_2, \dots, O_t | \lambda).$$

Now, the inductive steps of the Viterbi algorithm based on the dynamic programming technique are shown in pseudo-code as follows:

1. *Initialization:*
 $\delta_1(i) = \pi b_i(O_1), \quad 1 \leq i \leq N.$
 $\psi_1(i) = 0.$
2. *Induction on t , $2 \leq t \leq T$:*

$$\delta_t(j) = \max_{1 \leq i \leq N} (\delta_{t-1}(i) a_{ij}) b_j(O_t), \quad 1 \leq j \leq N$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} (\delta_{t-1}(i) a_{ij}), \quad 1 \leq j \leq N$$

3. Termination & Backtracking Path:

$$q_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i),$$

$$q_t^* = \psi(q_{t+1}^*), \quad 1 \leq t \leq T-1.$$

As it is recognizable by comparing the Viterbi to the forward procedure, the fundamental principle of the Viterbi algorithm is equivalent to the forward procedure. The two algorithms are based on the same dynamic programming technique but each uses it differently. They differ in that the path with a maximum probability is chosen in the Viterbi algorithm, instead of summing up the probabilities from all the paths, which is done in the forward algorithm. According to the Viterbi algorithm, given a current time t and a current state s , a previous state is selected by finding a maximum probability along the path from a start state at time 0 to the current state at time t . The selected previous states are recorded in the array vector $\psi_t(j)$ with information about the time and current state. At the end of the induction process, the last state of the best path producing maximum probability is known. The last state information is stored at $q_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i)$. As a last step, a best single path from the second last state to the last is traced using the array vectors $\psi_t(j)$ and $\delta(j)$, which is approximated as a best single state sequence by the Viterbi algorithm.

3.3.3 The Third Problem: Estimating Model Parameters

Given a sequence of observations $O = (O_1, O_2, \dots, O_T)$, a model $\lambda = (A, B, \pi)$ that would best describe the observation sequence needs to be estimated, best in the sense of maximizing the probability of a given observation sequence. Therefore, the third problem might be represented by the following expression:

Find a model $\lambda = (A, B, \pi)$ satisfying that the probability $P(O | \lambda)$ is maximized.

There is no known method to find $\lambda = (A, B, \pi)$ to maximize $P(O | \lambda)$ globally, but some techniques are known for finding a model maximizing the probability locally: the Expectation Maximization (EM) method which is equivalent to the Baum-Welch method (Demster, 1977) and the gradient approach (Levinson, 1983). In this section the EM method will be described for HMM applications.

Before going into further discussion, the definitions of two notations $\xi_t(i, j)$ and $\eta_t(i)$ need to be specified. The variable $\xi_t(i, j)$ is the probability of being in state i at time t and in state j at time $t+1$, given a model λ and an observation sequence O . Using the forward and backward variables defined in Section 3.3.1, the definition of $\xi_t(i, j)$ is derived as follows:

$$\begin{aligned}\xi_t(i, j) &= P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \\ &= \frac{P(q_t = S_i, q_{t+1} = S_j, O | \lambda)}{P(O | \lambda)} \quad \text{by (1)}\end{aligned}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} \quad \text{by (2)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)} \quad (3)$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad (4)$$

- (1) The definition of the conditional probability.
- (2) The definition of the forward and backward variables. (Figure 3.6 illustrates how the forward and backward variables are applied in this situation.)
- (3) Rewritten using the forward and backward variables.
- (4) Another representation using the forward and backward variables.

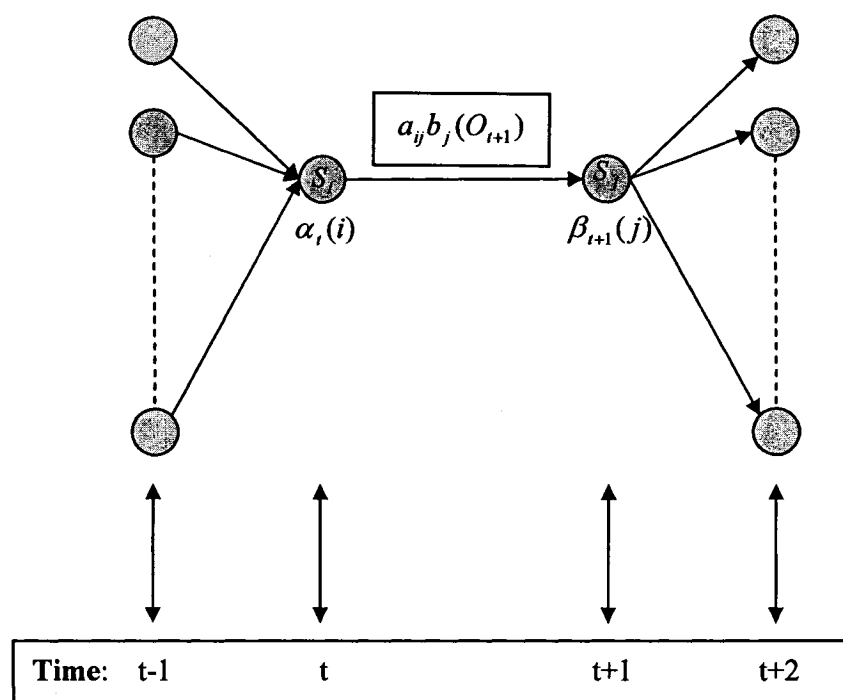


Figure 3.6: Illustration of being in state i at time t and in state j at time $t+1$, using the forward and backward variables

The variable $\gamma(i)$ is the probability of being in state i at time t , given an observation sequence. Using the definition of the variable $\xi_t(i, j)$, $\gamma(i)$ is defined as below:

$$\gamma(i) = \sum_{j=1}^N \xi_t(i, j).$$

When each value of the two variables $\xi_t(i, j)$ and $\gamma(i)$ over time t is accumulated, the summations are equal to the expected number of states associated with the variables. The definitions of expectations are as follows (Note that in the formulas below, the summation is performed up to the time $T-1$, not T because no transition occurs at time T):

$$\sum_{t=1}^{T-1} \gamma(i): \text{ The expected number of transitions from state } i,$$

$$\sum_{t=1}^{T-1} \xi_t(i, j): \text{ The expected number of transitions from state } i \text{ to } j.$$

Now, the training procedure of a model begins, during which the parameters of a model are estimated using the EM algorithm. Let λ be an initial model with $\lambda = (A, B, \pi)$. The parameters of the model could be selected randomly or according to a model designer's choice relying on one's experience or reasonable decision. In addition, training data should be prepared to serve as input data for the model. Once a set of training data is run into the initial model, the new parameters symbolized as $\bar{A}, \bar{B}, \bar{\pi}$ are obtained using the formula as follows:

$$\bar{\pi}_i = \gamma_1(i)$$

= Expected frequency in state i at time $t=1$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

= (Expected number of times in state i at a certain time and state j at the next time) / (expected number of times in state i)

= (Expected number of transitions from state i to j) / (expected number of transitions from state i)

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \mathcal{N}(j)}{\sum_{t=1}^T \mathcal{N}(j)}$$

= (Expected number of times in state j and producing the symbol v_k) / (expected number of times in stat j).

Then, the two probabilities $P(O | \lambda)$ and $P(O | \bar{\lambda})$ of the observation sequence are compared, conditional on two different models $\lambda = (A, B, \pi)$ and $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$. As $P(O | \bar{\lambda}) \geq P(O | \lambda)$ is true by Baum (1968), the re-estimation process is repeated until it is not improved up to a threshold point set by a user. That is, the model $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ replaces the current model $\lambda = (A, B, \pi)$, and such a re-estimation process is repeated until the new model does not improve the probability of the observations with the range of a setting threshold. As mentioned earlier, the EM algorithm does not guarantee that the globally optimal values can be reached, but it attains a local maximum close to the value of the initial model, which could be a local or global maximum.

3.4 HMM in Information Management Applications

Since the late '80s, HMM has become a popular statistical model in machine learning applications and has been successfully applied in a wide range of applications in the areas of signal processing, computational molecular biology, pattern recognition, and speech recognition due to its rich theoretical aspects (Rabiner, 1989). Recently, its applications have spread into new domains in the area of information management. In this section, a number of HMM applications in the field of information management are reviewed.

3.4.1 Information Retrieval

Information retrieval is the task of finding documents within a collection of text documents in order to satisfy a user's information need formulated as a query (Baeza-Yates & Ribeiro-Neto, 1999). While research in information retrieval is extensive, dating back some forty years, very little work has been done in the application of HMM to this field.

Miller *et al.* (1999; 1999a) presented an information retrieval model based on HMMs in a general domain. In their model, a HMM is constructed to provide the probability that a document is relevant to a given query. A ranked list of relevant documents, based on the probabilities, is provided to the user. In an expanded version of HMMs, topical information, synonyms, and other refinement features such as blind feedback, bigram modeling, and query weighting are also implemented to improve the information retrieval system's performance.

3.4.2 Information Extraction

Information extraction is the task of extracting specific information fragments from text documents (Freitag, 2000). Although the application of HMM to the information extraction domain is relatively new, HMM has been successfully applied to the problem of finding specific information in the following context: named-entity extraction (Bikel *et al.*, 1997, 1999); extracting gene names and gene locations from scientific abstracts (Leek, 1997); finding specific types of information from the headers of research papers (Seymore, 1999); building a model for retrieving free-style passages such as a sentence, a paragraph, a complete page, or another format, (Elke, 1994); recognizing different contents from FAQs in Usenet newsgroup (McCallum *et al.*, 2000a) or from Reuters newswire data (Freitag, 1999).

When applying information extraction, the HMM's design is particularly important because the model's configuration represents the internal structure of the specific information which will be extracted. Rabiner (1989) says that the optimal HMM structures for tasks have been explored in their operational setting, optimal in the sense

that the selected structure stands to gain the best performance. Seymore *et al.* (1999) and McCallum *et al.* (2000b) propose a method of automatically learning the model structure from available data, and show comparisons between different model structures.

3.4.3 Text Segmentation

Text segmentation is the task of automatically dividing a stream of text into topically homogeneous blocks (Allan *et al.*, 1998). Research on text segmentation using HMM was initiated by the DARPA-sponsored Topic Detection and Tracking (TDT) study, the purpose of which was to explore techniques for segmenting a stream of text from newswires, television captions, or automatic speech recognition transcripts. As in the TDT study, the task of segmenting a text stream involves finding the subject boundaries and topic transition points within a given block of text. In order to detect topic boundaries within a text stream, the statistical machine learning technique, HMM, is used. HMM-based segmentation models are applied to large streams of newswire and broadcast news (van Mulbregt, 1998). Yamron *et al.* (1998) use HMM techniques combined with a language model for a text segmentation model, and the experiments performed on the TDT Corpus show around 65% recall and 65.8% precision for the exact match of segmentation boundaries and about 80% for the match of a hypothesized boundary within 50 words. Blei and Moreno (2001) extend the idea of segmentation by embedding Hofmann's aspect model into HMM and comparing the two models' performance in segmenting articles and noisy transcripts of radio audio archives. Their experimental results show that AHMM outperforms HMM for small window size of 10-15 words, but HMM does better for larger windows. It seems that HMM is a potential tool for dealing with the problem of text segmentation. In addition to the aforementioned experiments on the use of HMM and its variations to text segmentation applications, Allan *et al.*, (1998) also hold the view that HMM is a promising method for application to the text segmentation.

3.4.4 Text Summarization

As for the text summarization task's goal, Mani (2001, p. 529) defines it as: *"to take an information source, extract content from it, and present the most important content to the user"* Due to the difficulty of achieving human-quality text summarization including discourse understanding, most of the recent work in automatic text summarization considers the work as the selection of the sentences that convey the main ideas of a document (Amini, 2002). The work of machine-generated text summarization has been explored statistically, linguistically, and through combination of both approaches. The statistical models including decision trees, naïve Bayesian, and neural networks have been applied to this task and produced reasonable summaries (Chuang, 2000).

Conroy (2001a; 2001b) was the first to propose a Hidden Markov model for text summarization. The proposed HMM estimates, given a sentence in a document, the probability that it can be included in a set of sentences for the document summary, and based on the probabilities assigned to sentences in the document, a list of sentences representing the document summary are determined. The text summarization model based on HMM takes the following three features into consideration as the important properties for the determination of whether or not a sentence can be a summary sentence: (1) positioning dependence of sentences (2) the number of terms in the sentence (3) the correlation between sentence terms and document terms. For the model's evaluation, a test set of 1304 documents from the TREC data set were selected, and, based on their human-generated summaries, the HMM-based model was shown to perform better than the naïve Bayesian method according to the averaged values ranging from 4 to 12¹⁹ over different sets of document sources. Conroy's application of HMM to text summarization is a pioneer work and has not been replicated at the time of this writing. Judging from the previous successful application of HMM to other information management domains,

¹⁹ As measured by the following formula based on the number of terms in summaries written by human (H) and by HMM system (M): The number of terms in common between the two summaries (by human and the system) is divided by the sum of H and M, then is multiplied by 100.

more profound research on HMM for machine-generated text summarization is expected to have considerable success.

3.4.5 Summary

In summary, this chapter covers the fundamental principles of HMM along with a description of its components. In addition, the problems that HMM frequently confronts when applied to an application are specified and the standard algorithms adapted as their solutions are described in depth.

For the past two decades, researchers dealing with topics related to sequential processing such as signal processing have been interested in the utilization of HMM as a model to simulate events occurring sequentially. As a consequence of these active research activities, the HMM-related algorithms in the machine learning approach have been developed, and highly praised methods dealing with HMM have become available. The presence of standardized procedures for HMM is recognized as one of the strong assets of the model.

HMM is well-known as one of the most popular research methods in speech recognition and pattern recognition since the 1980s. However, in the fields of information management such as information retrieval, information extraction, and text segmentation, it was not until recently that the use of HMM appeared in scholarly articles. Previous research has proved the HMM's capabilities as a means to efficiently parameterize the statistical features of sequential events and to effectively work as a framework for a series of events. A textual document can be treated as a sequential process of words consisting of a list of words from the standpoint of the statistical model. Based on the observation of the HMM's prior successful application in modeling a sequential pattern, there is reason to believe that Allan *et al.* (1998) are correct to say that HMM, as a proficient model in text segmentation, can be expanded to the other information management applications.

3.5 Summary and Conclusion

HMM has been a popular statistical model with a wide range of applications in quite diverging fields, such as speech recognition (Rabiner, 1989), character recognition (Hu *et al.*, 1996), DNA and protein modeling (Hughey and Krogh, 1996), behavior analysis and synthesis (Jebara and Pentland, 1999). The fundamental principles of HMM and the related algorithms have been covered in this chapter. HMM has been viewed as the stochastic generalization of a finite state machine, comprising states, transitions connecting states, and output symbols governed by probability distributions. However, it was quite recent that IR and its related fields, which deal with digital text as a target object, were explored as new applicative fields of HMMs. So far, some of the applications have been briefly reviewed to see what tasks have been involved and how HMMs have been applied to those tasks.

In this study, an automated classification model based on HMMs is designed and developed. One might ask why HMM is investigated for a TC model. There is experimental and theoretical evidence that lead us to be optimistic about a HMM-based TC model, and for the investigation of HMM being a standard TC model in this study. Firstly, the flexibility of the HMM structure is suitably applicable to the TC problem. Decisions about the topology of the HMM with regard to applications remains an open question (Rabiner, 1989). However, a priori knowledge about the application under consideration may be useful for the design of a HMM topology. Especially in a TC application, information about the structure of the classification scheme adopted is presumably useful for imposing a topology on the HMM. Secondly, quite a few research projects on the HMM and its variations have been done. The sound theoretical foundation supporting the statistical model provides a solid basis for use of the HMM-based model on the TC problem. Finally, as a result of the HMM's application to the extensive problems listed above, there is growing consensus that the HMM successfully formalizes the problem of dealing with non-text sequential processes (Levinson *et al.*, 1983; Rabiner & Juang, 1986; Kundu & Bahl, 1988; Baldi *et al.*, 1998; Bengio, 1999). It is also reported that the applicability of the HMM to text-based sequential processes is successful

(Charniak, 1993). The HMM has played a leading role as a standard model in two text-management applications: the statistical Language Model (LM) (Jelinek, 1985), and Part-of-Speech (POS) Tagging (DeRose., 1988). In these applications, the HMM was used as a model to provide the statistical ranking of a set of words for LM applications, and to assign text categories to the words of a text corpus for POS applications. The described statement seems to directly impact the applications of sequential processes with HMM, rather than directly impact the TC problem. Since the TC task is an instance of sequential process, however, it can be presumed that they might be pertinent to the TC topic under discussion.

Chapter 4

A CONCEPTUAL FRAMEWORK FOR AN HMM-BASED TC SYSTEM

4.1 Introduction

In this chapter, a conceptual framework for building the classification system proposed in Chapter 1 will be provided. Recall that the goal of this study is to build a HMM-based classification system to organize digital documents using the LCC. The two key features of the system are the HMM as a classification model and the LCC as a classification scheme. These two components, derived from different disciplines, are coalesced into the conceptual framework to be applied to an automated classification task.

As described in Chapter 2, LCC has served a role as an organization platform for library collections, such as books and journals. In this study, however, a new role for the LCC is sought to extend its coverage to digital documents. Thus, our attention turns to the problem of how it can be adopted for dealing with digital documents in the automated classification milieu. It should be emphasized, however, that the applicability of a library classification scheme as a platform for classifying digital documents is not the major issue of this study.

The HMM has been applied to a wide range of applications, including signal processing and biology (see Chapter 3 for the applications to some IR-related tasks). In this study, the use of HMM is explored for the task of automated classification based on LCC. In the task, HMM plays a role as a learning model for acquiring knowledge of LCC classes and, at the same time, as a prediction model for estimating the most probable classes for documents. In Section 4.2, a HMM prototype for the classification task using LCC is projected, and its roles in both aforementioned models are explained.

4.2 HMM Classification Model

4.2.1 Theoretical Background

The conceptual process of text classification may be described as the process of finding a relevant category c , for a given document d , that is, $P(c/d)$. By applying the Bayesian rule, the TC conceptual process can be seen as a combination of three components as follows:

$$P(c/d) = \frac{P(d/c)P(c)}{P(d)} \quad (4.1)$$

The review of each components will be explained. The first component appearing in the denominator of the expression (4.1) is $P(d)$, which is the prior probability of a given document d . In other words, it is how often a document d is likely to occur. In this study, all the documents are treated the same without any preference, and $P(d)$ is assumed to be constant for any category class. Therefore, the component in the denominator can be disregarded. The second component in the numerator part of the expression (4.1) is $P(c)$, which is the prior probability of a category c . It is equivalent to the prior knowledge on a given category c . In this study, since it is not known, the same probability will be applied to each of the categories. The last component in the expression (4.1) is $P(d/c)$, which is the probability of a document d given a category c . Therefore, under the Bayesian rule, the probability of a category c given a document d is approximated by the probability of a document d given a category c , without prior knowledge of documents and categories. The expression for the approximation is written in (4.2).

$$P(c/d) \approx P(d/c) \quad (4.2)$$

In this study, a TC classifier is recognized as a model for calculating the probability of a document d given a category c to approximate the probability of a category c given a

document d . Given an input document d , a HMM-based TC model is used to identify a category c from a set of all the categories of interest (symbolized as C in expression 4.3) that produces the highest probability as written in (4.3), meaning that the category c with the highest probability is inferred as the most relevant category for the document d .

$$\max_{c \in C} P(d/c) \quad (4.3)$$

A HMM, represented by $M = (I, E, T, O, S)$, is generally characterized by the five major components: initial probabilities I , output symbol emission probabilities E , state transition probabilities T , a set of output symbols O , and a set of states S . There are a couple of important assumptions underlying a HMM. First, an emission probability of a symbol is only subject to the current state. Second, the current state is dependent on only one previous state, instead of the history of all previous states. More details on the principles and algorithms of HMM can be found in Rabiner's article (1989).

4.2.2 Model Prototype

Generally speaking, the ideal HMM prototype for an application has not yet been known either theoretically or practically. However, building a HMM prototype is commonly approached by reflecting the nature of an application to which it will be applied.

In the design of a HMM prototype, there are several issues relating to the choice of a model: (1) choice of observation symbols, (2) model size, and (3) model type. The first issue is the observation symbols of the proposed model. As described in Chapter 3, observation symbols are the output of a HMM. In a HMM, an observation symbol is associated with an emission probability in a state, and it is supposed to be matched with an input symbol that comes from documents, as in our study. Thus, a set of observation symbols for our model is a collection of text words used for training the model. Precisely, words selected as features from the training document set serve as observation symbols in our model.

Next, model size and type are inseparable issues. The issue of designing model size is related to the problem of deciding the number of states and the state roles, and the third

issue of a model type is associated with how states are connected. At first, it begins with the determination of the roles of states. For our model, two dummy states, called START and END, are used to indicate where the model starts and ends. Thus, it always starts from the START state and must end with the END state. They are called dummy because they symbolize only where to start and stop the process of a model, without producing any symbol-associated emission probabilities like other states. Next, in the context of this application, the role of regular states (as opposed to dummy states) is considered. Starting from the belief that diverse information sources may convey information on different aspects of a specific subject domain, this model's assumption for the role of regular states is that if a state represents a particular information source for a target subject and all the states are connected in a way of combining all the information from the different sources considered, various information sources are believed to serve to be complementary in covering as many features of a specific subject domain as considered. Figure 4.1 shows a generic HMM prototype supporting the assumption mentioned above. In the Figure, a regular state is corresponding to a specific information source, and all the regular states are completely connected including self-directed loops.

Figure 4.2 displays the prototype of a HMM for a specific subject, as proposed for our study. The proposed HMM consists of four distinct states, two for dummy states and two for non-dummy states. The two internal states appearing inside the model represent two different information sources. The current model incorporates two different information sources (ISs) into the system, with the future possibility of adopting more diverse ISs. The first IS state, called *Dissertation Abstract Information Source*, refers to the subject information source based on the selected dissertation abstract descriptions from the ProQuest digital dissertations database. The second IS state, called *Subject-heading Information Source*, refer to the one relying on subject headings from OCLC MARC records. Based on our assumption, the proposed HMM prototype has been designed to mirror the idea that the abstract IS can provide more subject-specific information than the other IS, and the subject-heading IS can deliver subject-general information, for a target subject.

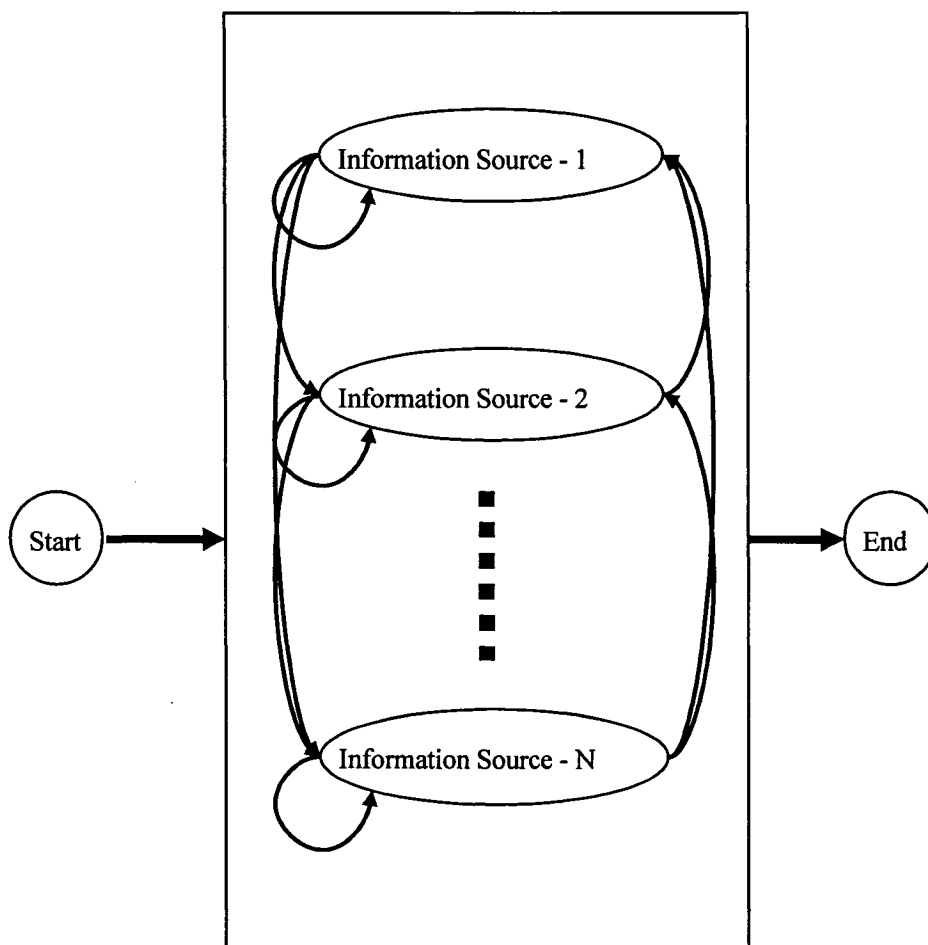


Figure 4.1: A generic HMM prototype for a specific subject

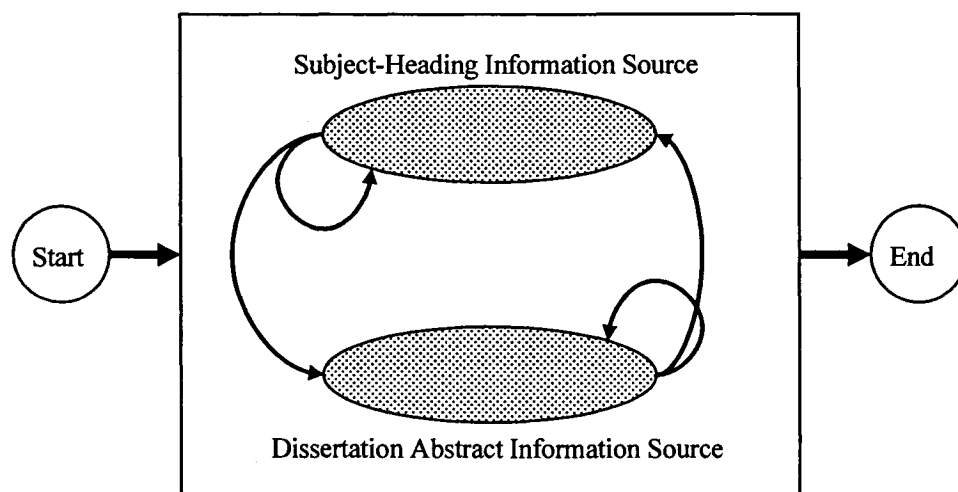


Figure 4.2: A proposed HMM prototype for a specific subject

Chapter 5

AN HMM-BASED TEXT DOCUMENT CLASSIFICATION SYSTEM

5.1 Introduction

Having described the theoretical framework in the previous chapter, the interest of this study makes a transition to how HMMs can be applicable as a classification model for digital documents. To return to our motivation for building a HMM-based TC system, recall that prior attempts in the field of TC led to the development of a number of TC models such as the Support Vector Machine (see Chapter 2), and there is no single, best classifier performing for various test settings (Sebastiani, 2002). Furthermore, some contradictory results have been reported across various classifiers with different types of documents tested (Sebastiani, 2002), and thus, the performance of TC models cannot be separated from the nature of TC tasks, which is directly related to the types of text documents used. Regardless of a great deal of research activity on TC, however, only a limited number of document types have been involved and tested so far, including Reuters collection (Lewis & Ringuette, 1994), medical documents (Hersh *et al.*, 1994), and Newsgroups (Lang, 1995).

In this study, the HMM as a new type of TC model is used in an attempt to solve the problem of classifying scholarly documents, namely *dissertation abstracts*, and experimented in a limited environment within a large-scale classification framework. The ability to automatically classify documents relies on the validity of HMMs. As it is described in Section 3.4, a great deal of previous work has provided significant clues for its learning capability in text-related tasks, along with a wide range of other learning and

prediction tasks. Thus, that motivates the reason to believe that the HMM may be used as an effective tool for learning information regarding categories of interest.

Figure 5.1 illustrates the overall process of the HMM-based TC system for the TC problem defined in Chapter 1 with the limitations described in Section 6.7. The classification system process can be divided into four sequential components as displayed: (1) *Model Data*, (2) *HMM Classifier*, (3) *Automated Classification*, and (4) *Evaluation*. The large arrows in bold between phases indicate the system flow among components. Unlike the *Automated Classification*, the other three components are processed off-line. Recall that the complexity of the running-time of the second component is said to be bounded by the multiplication of the number of states in a model and the number of terms (which is a large number) generated by the model (see Chapter 3). Although the running time for the second component is quite extensive, the total system running-time is not affected by this factor, because it is an off-line processing.

In the *Model Data* component, a set of data needed for building and evaluating the system is collected and processed, and training and test data sets are generated as an output. The two dotted lines starting from this component indicate where the data sets are used. The HMM Classifier module provides the model design's full process including training models, based on the training data set. Then, a complete TC system is constructed by putting the trained HMMs together, which is shown by the dotted line connecting the *Classifier* and *Classification* components. Once the system is developed, the system performance is measured by test data. The experimental results are analyzed in the *Evaluation* component. A dotted arrow linking the first and last components is used only to indicate a conceptual process of taking into account the classified documents as system output to be used as new training data sets for the system. Since the preparation of classified (labeled) documents is a highly time-consuming and cost-expensive process, such a logical mechanism might be viewed as a means of generating classified data (supervised data) from unclassified data (unsupervised data).

In addition, the system overview in Figure 5.1 also illustrates the general architecture of a HMM-based TC system for an arbitrary TC task. A specific TC problem is embedded in the figure and is linked only to data sections. Therefore, regardless of

classification systems and document types, it reserves all the general features of a HMM-based system, by replacing data segments only.

In the remainder of this Chapter, the first two components will be covered and how to build up the system will be included. In Section 5.2, it is described how the training/test data are selected and manipulated. Section 5.3 provides technical details as to how HMMs are designed and trained, including building our complete system. The remaining two components, *Automated Classification* and *Evaluation*, will be covered subsequently.

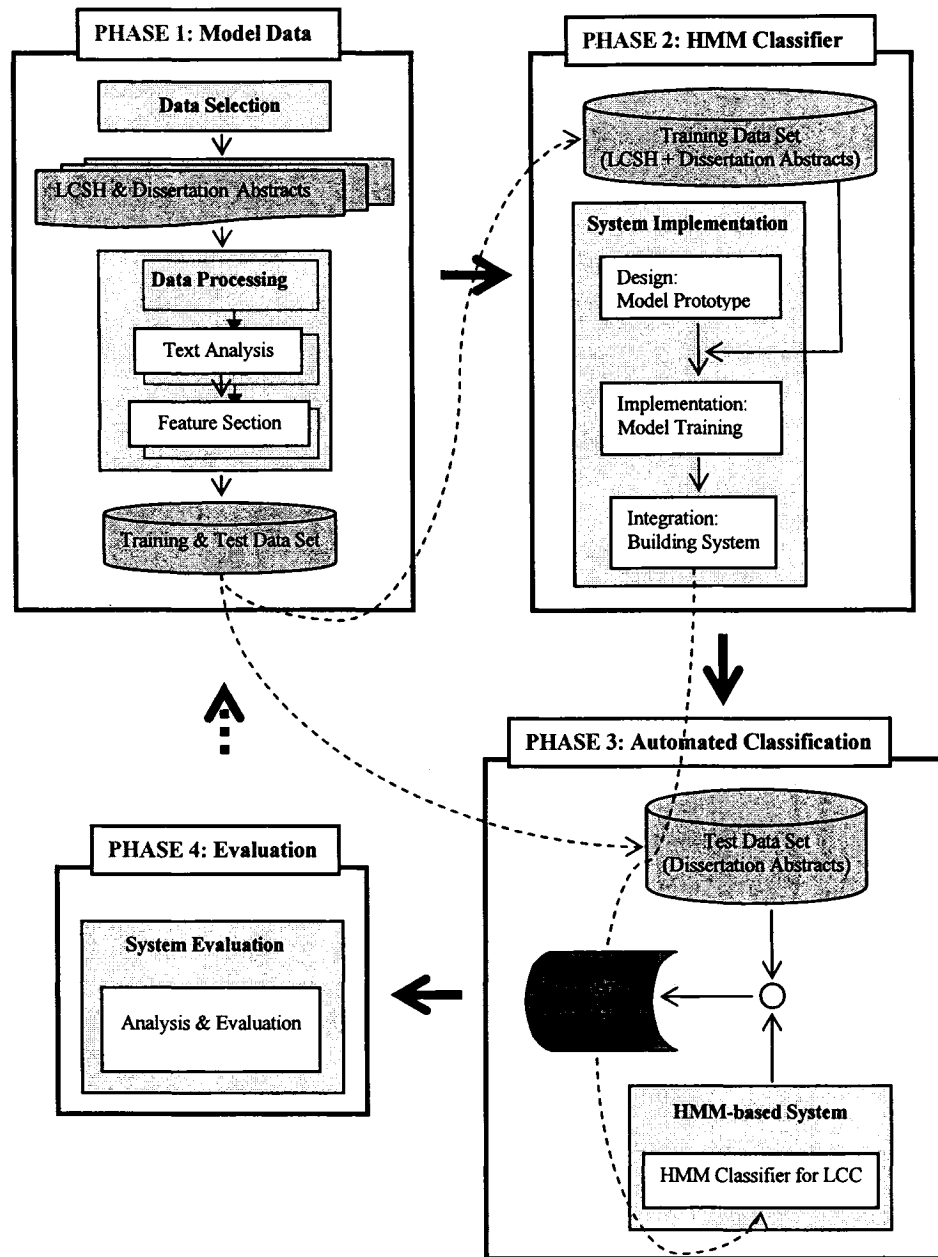


Figure 5.1: An overview of the proposed HMM-based classification system

5.2 Model Data

5.2.1 Training and Test Data

The machine learning paradigm, in the context of our TC task, has two capabilities: (1) learning knowledge for a class from a set of documents labeled as the class (called training data) and (2) predicting class labels for documents (called test data). In terms of data, a training data set and a test data set are similar in that each consists of a set of documents, each of which is associated with a class label, but differ in that training data are used for the learning purpose in (1), and test data are used for prediction in (2). A crucial rule that is not to be violated is that any document in the data sets should be used for only one purpose, because adopting training data for testing yields over-estimated results when measuring system performance, which is caused due to classifiers over-fitting (Mitchell, 1994).

What data will be used for the training and test data in this study? To answer this question we begin by evoking the motivation for the choice of data (in Chapter 2) to remind ourselves that the TC system's performance, based on a same TC model, remains unstable depending on the tasks using different types of data, and only limited numbers of different data sets have been involved in previous studies. Hence, the study of new document types for TC tasks is significant in enriching TC research and also aids in the discovery of the correlation of model characteristics to new document types for performance improvement.

Since the LCC is selected as the classification scheme for this study (see Section 2.6 for the selection of LCC), our training/test data set documents should have LCC numbers as their class information. For this study, available digital sources that satisfy the required conditions were explored, and the following sources were chosen as matches for our purposes: (1) MARC records; and (2) dissertation abstracts.

MARC Records

The bibliographic data of items in traditional libraries, as it appeared on catalog cards, is written in MARC records. The data shown on a MARC record includes: 1) classification data such as LCC and DCC class numbers, 2) descriptive data or bibliographic description including title, author, edition, publication, etc., 3) subject cataloging data or subject headings such as LCSH or MeSH. The classification data and subject-cataloging data of a MARC record, therefore, provide two different topical representations for a single item: classification data for symbolizing a certain subject scope and subject-cataloging data for referring to it by description. For such a reason, subject-cataloging data can be interpreted as descriptors for a classification number shown on the same item. The underlying assumption on which our approach stands is that authoritative subject vocabularies for a subject field referred to by a classification number can be obtained from MARC records because the list of subject vocabularies used for subject-cataloging data has been constantly controlled and maintained by authoritative organizations. Although it is admitted that the selection of vocabularies for a subject may suffer from human indexer inconsistency, the efforts by professional catalogers deserve to be considered as neutral judgments. The relationship between subject vocabularies and subject fields is described later in this chapter.

Dissertation Abstracts

A dissertation or thesis abstract provides a short descriptive summary of a research work written by the researcher. Also, professional catalogers catalog dissertations and theses, the source of abstracts, which means that a specific classification number depending on a classification scheme is assigned and subject data are available for the dissertation or thesis. Dissertation Abstracts can be considered reliable sources of topical descriptions to specific subjects, reliable due to being made by professionals. In comparison to the subject data in MARC records, dissertation abstracts are different in that they are more descriptive since abstracts usually consist of a number of paragraphs.

There are inherently valuable features in dissertation abstracts as a data source for TC applications. First, the appropriate LCC number and a textual description of the subject

are present. A dissertation abstract is viewed as a textual source pertaining to the particular subject described by the dissertation, and an LCC number appearing on a MARC record for a dissertation corresponds to the topical subject of the dissertation. Second, the information in the data source can be thought to be highly reliable in that dissertation abstracts in the database are written by the authors of dissertations who know the content the most, and a LCC number is selected by professional indexers who have been trained and experienced in their work.

Ideal data for training and testing a TC system based on a machine learning approach should contain both the textual information describing a specific topic and the most relevant class information on the topic, from the classes found in the selected classification system. From this point of view, a set of dissertation abstracts is an ideal data set with high reliability because the data for the TC system is obtained from qualified professionals, especially considering that a pre-classified data set covering a full range of subjects such as LCC is not available.

5.2.2 Selection of LC classes

The proposed text classification model is built and its performance is tested, based on the textual documents that were pre-classified using LCC, which are called either training data or test data depending on whether they are used in the process of system building or system evaluation. The ideal situation would be that the proposed system is built on coverage of all the LCC-supported classes and evaluated on the same class range. Due to the limitations on time spent for system development, however, a few selected LCC classes are considered for the development and evaluation of our classification system. The selection of these main LCC classes is based on Harter's claims, as described next.

One issue that makes the automatic text understanding-related works such as machine translation and information retrieval difficult may be the ambiguity of natural language. Harter (1986) states that the semantic ambiguities caused by homographs of words or phrases differ by discipline. In McGrath's project (1978), randomly selected academic faculty members were asked to evaluate sixty three disciplines on a scale between 5 (the

hardest) and -5 (the softest)²⁰. According to the total score assigned to each discipline, a ranked list of the disciplines was generated. Table 5.1 shows the ranked disciplines ranging from the *hardest* discipline to the *softest* discipline (the full version of the list is produced in Appendix B). It was claimed by Harter that the semantic ambiguities inherent in soft disciplines will generally make online search problems caused in language more difficult than for hard disciplines.

The selection of three LC classes for the experiments in this study was made in the context of investigating Harter's claim in text classification. Three main LC classes are supposedly chosen to represent the softest discipline, the hardest, and the middle. Since McGrath's classification of academic disciplines and LC main classes are somewhat different from each other, the selection of three LC classes is attempted as a way of considering the disciplines as a reference, rather than obeying them as they are, as follows: First, the 'Fine Arts' main class in LCC is selected to represent soft disciplines since this discipline is ranked as the softest discipline. Second, the 'Science' LCC main class is considered to be representative of the hard disciplines because all the disciplines under the 'Science' class are listed on the top portion towards the hard disciplines: Mathematics, Chemistry, and Physics are ranked first, second, and fifth hardest disciplines, respectively. Third, Agriculture was chosen to represent an intermediate discipline between the hard and soft disciplines.

²⁰ "Hard" and "soft" concepts were originally introduced by Biglan (1973) in establishing the relation of discipline and paradigm. Harter (1986, p.34) describes their definitions as "In general, paradigmatic disciplines are the hard sciences, while the social sciences and humanities are soft."

Hardest Discipline



Softest Discipline

1. Mathematics
2. Chemistry
3. Electrical Engineering
4. Chemical Engineering
5. Physics
6. Mechanical Engineering
7. Civil Engineering
8. Statistics
9. Spanish
10. German
-
28. Library Science
29. Aerospace Studies
30. Agriculture
31. Finance
32. Office Administration
33. Ind. and Tech. Educ.
35. Business Communication
36. General Business
37. Economics
38. Home Economics
39. Management
-
55. Applied Arts
56. Dance
57. Education
58. Psychology
59. Sociology
60. Political Science
61. Philosophy
62. Recreation
63. Fine Arts

Table 5.1: The abridged list of the hard/soft disciplines (Source: McGraph, 1978, p. 22)

5.2.3 Data Selection

The training / test data sources described in the previous sections are collected from two different databases. This section will describe the database sources where the training / test data sets are obtained, and give a detailed explanation of how the training / test data are interpreted and acquired.

MARC Records

OCLC had issued a Microsoft Windows version cataloging system called CatCD™²¹, containing bibliographic and authority records in a subset of the OCLC Online Union Catalog (OCLC 2002). A set of CatCDs™ comprises Recent Books, Visual Materials & Computer Files, LC Name Authorities, and LC Subject Authorities CDs. The OCLC Recent Books CatCDs™, consisting of more than one million MARC records issued in May 2000, are used as a MARC record source for the training data of the model. Table 5.2 shows the distribution of the 1,153,070 bibliographic records in the Recent Books CatCDs to the twenty-one LC top-level classes. Classes D, E, and F refer to the same subject, HISTORY, where E and F especially are for the same subject, AMERICAN HISTORY. There is significant variation in the number of MARC records according to different subjects, with a maximum of 248444 for the subject P (language and literature) and a minimum of 2737 for the subject Z (library science). The non-uniform distribution in quantity may be important as it may affect the predictive power of the trained model, which will be discussed further in the experimental results.

An instance of an OCLC-version bibliographic MARC record extracted from the OCLC CatCD™ is presented in Figure 5.2. The information of standard bibliographic USMARC formats can be divided into four main components: 1) *leader*, 2) *record directory*, 3) *variable control fields*, and 4) *variable data fields* (Chan, 1994). The *leader* provides specific information relating to MARC record processing such as logical record length, whereas the *directory* gives the relative locations of *variable fields* within a record, consisting of a series of fixed-length fields. The *variable control fields* are used to

²¹ The OCLC CatCDs™ for Windows was discontinued in 2001.

have information concerning record selection and processing such as language and control number. The *variable data fields* provide various classification-related numbers (International Standard Book Number, LC call number, or DDC number), and bibliographic and subject information.

Figure 5.2 displays an example of a formatted OCLC-MARC record. In the example, a series of *variable data fields* are displayed with three-digit tags attached at the beginning of the fields. The three variable data fields are of interest in considering training data. The tags identifying the three data fields are as follows: 050 (LC call number), 090 (locally assigned LC-type call number), and 650 (subject added entry - topical term). A *variable data field* in a formatted record consists of a three-digit field tag, a two-digit indicator, sub-fields, and data. The interpretation of the numeric codes occurring in the three variable data fields can be found in Tables 5.3 and 5.4. As an example, let us see the double zeros occurring after the 050 field tag in Figure 5.2, referring to the first and second indicators in this field. By consulting the 'ind1' and 'ind2' columns of the 050 tag in Table 5.3, the double zeros are interpreted to mean that the item referred to by the MARC record in Figure 5.2 is in the LC, and the LC call number in the 050 field is assigned by LC. The remaining part of the 050 field in Figure 5.2 reads 'AG243 β b .A425 1994' and the ' β ' and the 'b' of the ' β b' indicate the sub-field separator and sub-field code (SFC), respectively. According to the definition of 'b' under the SFC column of the 050 tag in Table 5.3, '.A425 1994' denotes an item number. Note that the first SFC 'a' is conventionally omitted in the field of 050 as well as other fields. Therefore, 'AG243' must be read as a LC classification number, caused by the definition of the 'a' row under the SFC column. Also, the 090 and 650 fields can be decoded in a similar way. For 650 fields, various subject heading sources are used, and the second indicator of a 650 field represents a subject heading used for the field. That is, in Figure 5.2, the first two 650 fields use LCSH and the last two use the LCSH for Children's Literature.

After more than one million MARC records collected from the OCLC CatCDTMs are examined, some records are found to be without a 050 or a 090 field, and others are without a 650 field. Table 5.5 shows the statistics of only the records that contain both 050 and 650 fields, or 090 and 650 fields among the records extracted from the OCLC

CatCD™s, May 2000. It is very important that MARC records used for this research contain both a 050 or 090 field and at least one 650 field because the presence of both fields - classification number field and subject heading field – are the reason why MARC records are being used as our training data. A 050 field or a 090 field has the LC classification number for an item, which is supposed to be reliable information on the subject with which the item is associated. These MARC records may provide a suitable data source for this TC research project because the classification related information such as LC numbers and subject headings has been selected and assigned by professional experts. Although the problem of inconsistency in making decisions regarding the subjects of a document among human catalogers is admitted (see section 1.3), it is still valuable in that better decision work does not seem to be available at present.

The following two steps give details on the process of extracting training/test data from the fields of interest in MARC records.

1. Find proper classification number.

The classification number field of a MARC record is comprised of a tag number (050 or 090), an indicator (two one-digit numbers), a sub-field code, and a sub-field. The first sub-field code of the 050 or 090 tag, βa^{22} , (which is implicitly expressed as the default, and is not shown in Figure 5.2) indicates that the following sub-field is a classification number. The second sub-field code of the 050 or 090 tag, βb , denotes that the corresponding sub-field is an item number for 050 or a local cutter number for 090. Since the classification number in the first sub-field has sufficient information to cover the top two-level LC classes at which our system is aiming, only the first sub-field data are kept for further processing. Therefore, only the first field data 'AG243' in the example in Figure 5.2 is retained for further processing and the remaining parts are disregarded as shown in Figure 5.3.

²² β is used as the delimiter of sub-field code in OCLC-MARC record. It may vary from system to system.

2. Extract the first subject heading field.

A 650 field has the same structure as those with 050 or 090 fields except that it interprets indicators and sub-field codes differently. The indicators for a 650 field refer to the source of the subject terms appearing in the sub-fields. For instance, as specified in Table 5.4, the second indicator 0 of the 650 field denotes that the terms in the fields are selected from the LC Subject Headings, whereas the second indicator 1 of the same field indicates that they are selected from the LC Subject Headings for Children's Literature. As shown in Table 5.4, the nine (from 0 to 8) different sources of subject systems are used to designate the source of subject headings for the 650 field.

Figure 5.3 presents a set of only the interesting fields which have been selected from the original record in Figure 5.2. As shown, it is common to have multiple 650 fields in a MARC record because the subjects associated with an item might not be limited to only one topic. Only the first 650 field is used in this study, however, because based on the LC rules for the assignment of classification numbers, the primary subject heading is the only one used for classification number assignment (Larson, 1992). Although, as Larson points out, it is found out that there is considerable inconsistency in subject heading assignments (Chan, 1989), the decision by professionals can be considered to be reliable and respectful for this study since the work of assignments is naturally subjective.

In this study, the various subject sources are not treated differently, which means that different subject headings are disregarded. After the rules in the first and second steps have been applied to the training data's 050 or 090 fields in Figure 5.3, the resultant data are shown in Figure 5.4 and 5.5.

Given the first subject heading, all the sub-fields are included such as general, chronological, and geographic subdivisions. In the previous step, only the first sub-field (referring to a classification number) is kept and the remaining sub-fields (referring to item numbers or local cutter numbers) are discarded. As a result, the collected subject heading becomes a set of

descriptive terms for more specific topics than represented with the classification number, which is acceptable because descriptions for narrower topics can be a subset of descriptions for broader topics.

LCC Main Classes	The Number of MARC Records
A – General Works	4593
B – Philosophy. Psychology. Religion	79908
C – Auxiliary Sciences of History	14592
D – History (General) and History of Europe	102555
E – History: America	16888
F – History: America	31183
G – Geography. Anthropology. Recreation	38126
H – Social Science	172897
J – Political Science	29865
K – Law	64802
L – Education	31651
M – Music and Books on Music	13323
<i>N – Fine Arts</i>	<i>52802</i>
P – Language and Literature	248444
<i>Q – Science</i>	<i>77425</i>
R – Medicine	50450
<i>S – Agriculture</i>	<i>19645</i>
T – Technology	75913
U – Military Science	8404
V – Naval Science	2737
Z – Bibliography. Library Science. Information Resources (General)	16867
<i>The total number of MARC records</i>	<i>1153070</i>

Table 5.2: The LCC top-level class distribution of MARC records in the OCLC CatCD™ Windows, May 2000

OCLC: 28498460	Rec stat: p			
Entered: 19930625	Replaced: 19950626			
Type: a	ELVl:	Src: j	Ctrl:	Lang: eng
BLVl: m	Form:	Conf: 0	Biog:	MRec: Ctry: vau
	Cont: f	Gpub:	Fict: 0	Indx: 0
Desc: a	Ills: a	Fest: 0	DtSt: s	Dates: 1994,

U010 93-11599/AC
 U040 DLC &c DLC
 U020 0809494590 (lib. bdg.)
 U020 0809494582 (trade)
 U050 00 AG243 Bb .A425 1994
 U082 00 031.02 B2 20
 Y049 OCLC
 U245 00 Amazing facts.
 U250 Authorized English ed.
 U260 Alexandria, Va. : Bb Time-Life Books ; Ba Richmond, Va. : Bb School and library distribution by Time-Life Education, &c c1994.
 U300 87 p. : Bb col. ill. ; Bc 31 cm.
 U440 2 A child's first library of learning
 U500 Illustration on lining paper.
 U520 Answers such questions as "why are there seven days in a week?" and "where did chewing gum come from?"
 U650 0 Curiosities and wonders Bx Juvenile literature.
 U650 0 Handbooks, vade-mecums, etc. Bx Juvenile literature.
 U650 1 Curiosities and wonders Bx Miscellanea.
 U650 1 Questions and answers.
 U710 2 Time-Life Books.

Figure 5.2: An example of an OCLC-MARC record from the CatCD™ Recent Books, May 2000

Tag	Ind 1	Ind 2	SFC	Definition
050				Library of Congress Call Number (R)
	␣			Undefined
	0			Item is in LC
	1			Item is not in LC
		␣		No information provided
		0		Assigned by LC
		4		Assigned by agency other than LC
			‡a	Classification number (R)
			‡b	Item number (NR)
			‡d	Supplementary class number
				Note: Subfield obsolete
			‡u	Custodial location (R)
			‡3	Materials specified (NR)

Tag	Ind 1	Ind 2	SFC	Definition
090				Locally Assigned LC-type Call Number (R)
				• OCLC defined
	␣	␣		Undefined
			‡a	Classification number (R)
			‡b	Local cutter number (NR)
			‡e	Feature heading (NR)
			‡f	Filing suffix (NR)

Table 5.3: Interpretations of indicators and sub-field codes for 050 and 090 tags

Tag	Ind 1	Ind 2	SFC	Definition
650				Subject Added Entry-Topical Term (R)
	Ø			No information provided
	0			No level specified
	1			Primary
	2			Secondary
		0		Library of Congress Subject Heading
		1		LC subject headings for children's literature
		2		Medical Subject Heading
		3		National Agricultural Library subject heading
		4		Source not specified
		5		Canadian Subject Heading
		6		Repertoire des vedettes-matiere
		7		Source is specified in subfield ‡2
		8		Sears subject heading <ul style="list-style-type: none"> • OCLC defined
			‡a	Topical term or geographic name as entry element (NR)
			‡b	Topical term following geographic name as entry element (NR)
			‡c	Location of an event (NR)
			‡d	Active dates (NR)
			‡e	Relator term (NR)
			‡v	Form subdivision (R)
			‡x	General subdivision (R)
			‡y	Chronological subdivision (R)
			‡z	Geographic subdivision (R)
			‡2	Source of heading or term (NR) <ul style="list-style-type: none"> • See LC's <i>MARC Code List for Relators, Sources, Description Conventions</i>
			‡3	Materials specified (NR)
			‡6	Linkage (NR)

Table 5.4: Interpretations of indicators and sub-field codes for the 650 tag

LC classification	The total number of records	The number of the records with both LCC and subject heading fields
A	4593	2883
B	79908	66424
C	14592	8070
D	102555	61859
E	16888	13618
F	31183	18724
G	38126	36656
H	172897	161365
J	29865	25500
K	64802	63230
L	31651	30309
M	13323	11987
N	52802	43760
P	248444	141692
Q	77425	76217
R	50450	50274
S	19645	19351
T	75913	75224
U	8404	6808
V	2737	2347
Z	16867	14939
Total	1153070	931237

Table 5.5: Statistics of OCLC-MARC records from OCLC CatCD

OCLC: 28498460 050 00 AG243 βb .A425 1994 650 0 Curiosities and wonders βx Juvenile literature.

Figure 5.3: An example of raw training data

OCLC: 28498460 050 AG243 650 Curiosities and wonders Juvenile literature.

Figure 5.4: An Example with 050 and 650 fields only

OCLC: 28498460 050 AG243 650 curios wonder juvenile literatur

Figure 5.5: An example of the output of the text manipulation process

Dissertation Abstracts

Dissertation abstract is used as the source for both our test data and training data. To build up the training/test data sets, the ProQuest Digital Dissertation (PQDD) online database to which McGill University²³ subscribes was used. The PQDD database is a single comprehensive source for doctoral dissertations and master's theses with the following characteristics and limitations to length, coverage, and language:

1. The Ph.D. dissertations published since July 1980, are guided to provide 350-word abstracts, and the Master theses published since 1988, 150-word abstracts. Both sources are the target data sets for this study.
2. This database's collection consists of the dissertations or theses accepted mainly in North America. More than 90 percent of the dissertation abstracts originating from North America are available from this database.

Our data selection is limited in year to those documents dating from 1980 and later in order to have an equal size of abstracts. The reason for this is that longer documents contain more terms than shorter ones and, as a result, the former may be considered to have more information content. Yet, dissertation abstracts are fairly uniform in length as mentioned above, and can be treated to have an equal amount of information, for the dissertations after 1980 and the theses after 1988, inclusively. As for the language issue, this research is interested in classifying documents written in the English language. Due to the comprehensive coverage of dissertations from North America in the PQDD database, the database is utilized to collect the data for this study.

The primary question in this section is how dissertations are to be selected for each of the subjects determined. Several national- and international- online catalogues and online databases are investigated in order to find the best bibliographic source allowing users to search dissertations under a specific LCC. Most do not provide search functions to

²³ <http://www.library.mcgill.ca/cdroms/PQDD.htm>

support the search requirements needed for obtaining data for this study. For example, AMICUS, the information system of the National Library of Canada, does not support a tool for retrieving theses through the use of LCC subjects. The OCLC WorldCat™ database is the only electronic database close to being ideal, since it allows the document types (thesis in our case) and subjects for a range of LC call numbers to be specified. Figure 5.6 displays the OCLC WorldCat™ database interface, set for several of these search restrictions. The search screen mirrors search limitations in the document type for 'thesis' as a subtype of 'book', the language of document for 'English', and the publication year, for a LCC class specified in the Keyword section. Figure 5.7 shows the results of the search functions formulated in Figure 5.6. There are, however, a few limitations in formulating a LCC for our purposes with the OCLC WorldCat™ interface. The online database does not support the feature of supporting a range of LCC numbers. Therefore, the combination of wild characters and Boolean operators is used to represent a target range of LCC numbers. Another search limitation with the interface is that a number of at least three digits must be provided before using wild characters as in Figure 5.6.

In training and testing a system in the ML approach, neither the optimal size for training and test data sets nor the best ratio between training and test data is theoretically known. In most research concerning the ML approach, the decision regarding size has been limited by practical issues such as time. The same rule applies to the problem of deciding the size of data in this study. For training and test data for this study, the first twenty-five dissertation abstracts obtained from the PQDD database are used, based on the bibliographic data of the relevant list acquired through our OCLC WorldCat™ database search, given a target LC class. Twenty of these dissertations are used for training purposes and the other five are used for testing purposes. Since the rationale for splitting the training and test data sets is concerned with the method of using them, they will be described in the same section later (see Section 6.3.1).

The three top level LCC and their subclasses (see 5.2.2. for the reasons for these choices) are of interest in this experiment: 'Q' for the 'Science' class and its 12 subclasses, 'S' for the 'Agriculture' class and its 6 subclasses, and 'N' for the 'Fine Arts'

class and its 8 subclasses. Table 5.6 displays the list of subclasses of interest for our experiments. For our classification experiments, abstracts (called test data set) for a second level LC subclass are obtained from the PQDD database, and are categorized over the twenty-five²⁴ second-level subclasses.

²⁴ The Subclass NE for print media is not considered as a classification class in our experiments because fewer than 25 abstracts are collected for the subclass.

Search in: WorldCat

Search for: qa2#

Indexed in: Library of Congress Call No. (lc:)

Limit to: 1980- English All

Limit type to: match any of the following

Subtype limits: All types Theses/dissertations Monographs

Limit availability to: match any of the following

Rank by: Default

Figure 5.6: A snapshot of the OCLC WorldCat™ database search screen

List of Records

- Click on a title to see the detailed record.
- Click on a checkbox to mark a record to be e-mailed or printed in Marked Records.






- ☐ 1.  Mathematics in the Enlightenment :
a study of algebra, 1685-1800 /
Author: Rider, Robin E.
Publication: 1981, 1980
Document: English : Book : Thesis/dissertation/manuscript
Libraries: 6
More Like This: [Search for versions with same title and author](#) | [Advanced options ...](#)
- ☐ 2.  The commercial revolution and the beginning of Western mathematics In Renaissance
Florence, 1300-1500 /
Author: Van Egmond, Warren, 1946-
Publication: 1980, 1976
Document: English : Book : Thesis/dissertation/manuscript
Libraries: 8
More Like This: [Search for versions with same title and author](#) | [Advanced options ...](#)
- ☐ 3.  Hermann Weyl, mathematics and physics :
1900-1927 /
Author: Sigurdsson, Skuli.
Publication: 1990
Document: English : Book : Thesis/dissertation/manuscript
Libraries: 2
More Like This: [Search for versions with same title and author](#) | [Advanced options ...](#)

Figure 5.7: An instance of the dissertation list using the search query in Figure 5.6

Class Q - SCIENCE	
Subclass Q	Science (General)
Subclass QA	Mathematics
Subclass QB	Astronomy
Subclass QC	Physics
Subclass QD	Chemistry
Subclass QE	Geology
Subclass QH	Natural history – Biology
Subclass QK	Botany
Subclass QL	Zoology
Subclass QM	Human anatomy
Subclass QP	Physiology
Subclass QR	Microbiology
Class S - AGRICULTURE	
Subclass S	Agriculture (General)
Subclass SB	Plant culture
Subclass SD	Forestry
Subclass SF	Animal culture
Subclass SH	Aquaculture. Fisheries. Angling
Subclass SK	Hunting sports
Class N – FINE ARTS	
Subclass N	Visual arts
Subclass NA	Architecture
Subclass NB	Sculpture
Subclass NC	Drawing. Design. Illustration
Subclass ND	Painting
Subclass NE	Print media
Subclass NK	Decorative arts
Subclass NX	Arts in general

Table 5.6: The selected three main classes and their subclasses

5.2.4 Data Processing

5.2.4.1 Text Analysis

In this section, text processing algorithms will be applied to the extracted text, with the aim of eliminating unnecessary data and retaining only meaningful data. A stream of characters from the classification number and subject heading fields in OCLC-MARC records as well as from text in the dissertation abstracts selected from the PQDD database are sequentially processed in the following ways:

Step 1: Preliminary Processing

There are a few issues relating to the preliminary process. First, non-semantic characters or symbols are removed. For example, punctuation marks such as commas, periods, semicolons, colons, apostrophes, quotation marks are suppressed in text. As shown in Figure 5.3, after the selection of proper fields are performed, a few symbols such as commas (,) and periods (.) still remain in the 650 data field. Since the presence of these symbols is unnecessary in recognizing the subject that the 650 field is supposed to represent, they can be removed from the training data without loss of meaning.

The second issue is the problem of case sensitivity. In this study, the cases of words are all treated as lower case, and the issue is disregarded. As Baeza-Yates & Ribeiro-Neto (1999) state, the problem of case sensitivity can be normally handled by converting the letters to either of the cases, with only a few exceptions. Any significant benefit taken by case sensitivity cannot be expected with the data set for our purposes, in contrast to the fact that if case sensitivity is considered, the same terms will be treated as different ones because the first word of each sub-field in the 650 fields or a sentence always begins with the upper case.

The third issue is how to deal with numbers. A number might pose problems because it sometimes carries semantic information such as the date. In this study, however, numbers are not considered for the following reasons. First, numbers are seldom used as subject headings. Second, our study has an interest in subject

classifications based on only the top two levels of the three main LC classes according to the hierarchical structure of LCC, but the use of years in LC classification become available only at a much deeper level of the structure than the second level. In other words, the information of *date* does not seem to provide much evidence for the classification.

Step 2: Applying a stop-word list

A stop-words list is a list of terms that occur frequently in documents, and that are not supposed to be indexed due to their semantic insignificance. A document consists of a sequence of terms, and is generally represented by selected terms, not by all of them. Thus, in most text processing tasks, a stop-words list has been widely used as a way of eliminating semantically insignificant terms, independent of their contexts. Our system uses the 571-member stop-word list²⁵ Salton (1971) employed in his SMART system.

One might argue that the decision as to whether a word is a useful indexing term is partly conditional on the context in which it is used. Nevertheless, it is commonly believed by most researchers in information retrieval that better benefits can be had by applying a stop-word list, resulting in a significantly reduced number of indexed terms, than that by not utilizing one (Baeza-Yates & Ribeiro-Neto, 1999). Considering that our data consist of subject headings and dissertation abstracts, which are not loosely descriptive as are essays, and that they contain a high percentage of well-selected terms by ones who know the contents well, the application of a stop-word list, mostly comprising articles, pronouns, adjectives, verbs, and adverbs, to our data set is a needed process.

Step 3: Applying Porter's stemming algorithm

A stem is defined as the root of a word that remains the same in its morphologically diverse forms. In other words, it is the minimum unit that

²⁵ The electronic version of Salton's 571 stopword list was obtained from <ftp://ftp.cs.cornell.edu/pub/smart/english.stop>.

conveys the semantic information unique to a word. The objective of a stemming algorithm is to flatten variants of a word so as to result in a single unique form. A stemming technique is a rule-based approach where a list of rules for finding stems is written and applied in order. The performance of information retrieval systems has been improved through the use of stemming algorithms (Baeza-Yates & Ribeiro-Neto, 1999). As consequences of applying a stemming algorithm, the size of the index list is greatly reduced and simultaneously various forms of a word are recognized as a same index term because the words derived from the same root are assigned the same index term. Our system adopts Porter's stemming algorithm²⁶ (Porter, 1980) that is the most popular among similar algorithms due to its simplicity and elegance (Baeza-Yates & Ribeiro-Neto, 1999). The effect of applying Porter's algorithm can be found in Figures 5.4 and 5.5, where the modifications made to terms before and after the application of Porter's algorithm are shown.

5.2.4.2 Feature Selection

So far, several simple techniques for manipulating text have been discussed and applied to our data set. As a consequence of the previous Data Processing steps (see Figure 5.1), given each of the classes of interest, a set of related terms are collected from two different sources and manipulated to reduce the number of indexed terms (the process is generally called *feature selection*). In other words, a list of terms is prepared to represent a concept that corresponds to each of the LCCs of interest for this study.

The TC task is often recognized as a concept learning problem. A concept space is an abstract space representing a subjective class of interest. In our model, the 25 subject classes from the hierarchical structure of LCC are chosen. From the perspective of concept space, the ultimate task targeted in this dissertation is reformulated as the one of statistically constructing concept spaces for the 25 subjects by using a statistical model.

²⁶ The PERL version of Porter's algorithm adopted in building this system is available from <http://www.omsee.com/~martin/stem.html>. The Perl script was encapsulated in the codes used for analyzing training/test data, which is available in Appendix C.

The following sections will provide a description of how concept spaces are statistically built using HMMs.

5.3 Relationship of LCCs and LCSHs

One of the challenges in building a HMM-based classification system is how LCC as a classification platform for digital documents is represented within the TC model. In its use in traditional libraries, a human being has had the task of understanding the LCC and applying its rules appropriately. However, in TC tasks, they must be understood and manipulated by machines (computers), wherein the difficulty of this research lies. Therefore, a conceptual model of LCC for our purposes should support a representation of LCC where machines are able to handle LCC in the same way that a human being does.

The following questions for a machine-readable representation of a LCC class can be raised: how can a LCC class be represented by machines, and what information does a machine need for that task? Exploring the relationship between bibliographic data in MARC records inspires our approach. A MARC record consists of a set of bibliographic data for the various types of library materials, including books, visual materials, and computer files. The data shown on a MARC record includes: 1) classification data, such as a LCC class number, 2) descriptive data or bibliographic description, such as title and author, and 3) subject cataloging data or subject heading, such as LCSH. LCSH is a subject authority list for the subject cataloging of library materials. It has been a standard authorized list in most large general libraries in North America and abroad, including many special libraries and some smaller libraries (Chan, 1994).

The relationship between LCSH and LCC numbers is embedded in MARC records. Given a MARC record for a library material, a LCC number indicating its LCC class may be found in the field of LC Call Number. Also, a list of subject headings is found in the field of Subject Heading, along with their source information, and only the first subject heading among the list is used for making a decision on its corresponding classification number (Larson, 1992). For that reason, only the first LCSH, in the case that multiple LCSHs exist, is considered as a descriptor to the LCC class indicated by the LCC number

assigned. Having that in mind, only the first LCSHs collected from a set of MARC records for a target LCC number are regarded as subject authority descriptors for the LCC class designated by the number. Therefore, repeating this process for all the classes appearing on MARC records brings upon the establishment of a subject authority list for the LCC. Figure 5.8 illustrates the conceptual relationship between LCSHs and LCC subjects and between LCC subjects and LCC numbers appeared in MARC records, based on the framework mentioned. A LCC number has a one-to-one relation to a LCC class, and a group of LCSHs collected for a LCC number also have one-to-one relations to the LCC class. Thus, given a LCC class, a unique pair of corresponding LCC number and a set of subject authority descriptors exist.

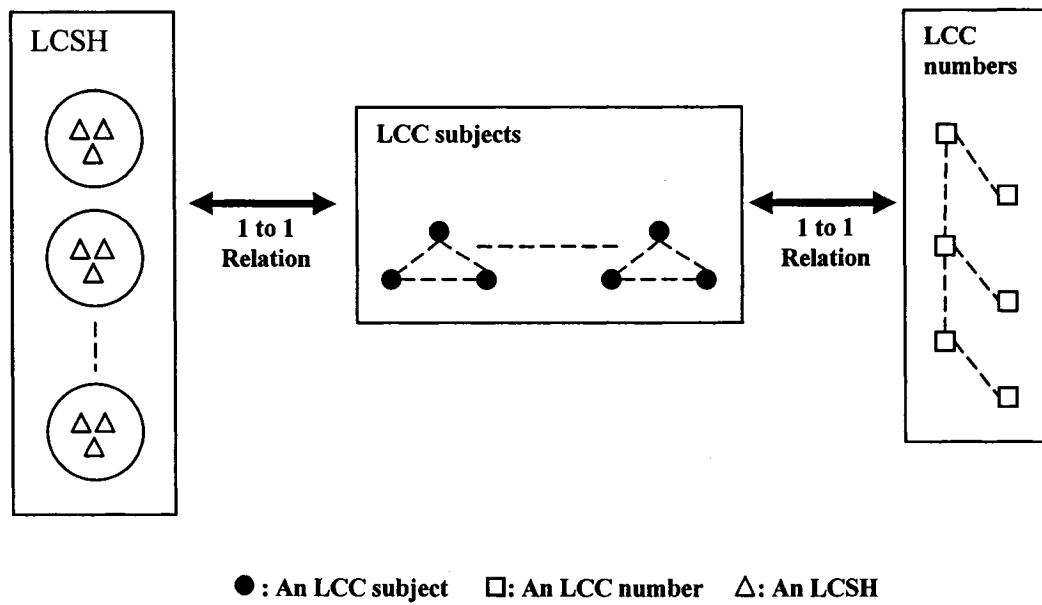


Figure 5.8: Relationships of LCSH and LCC numbers to LCC subjects

5.4 Building a HMM-based Classifier

5.4.1 Introduction

The architecture of a HMM-based classifier, as shown in Figure 5.1, displays the full procedure of a HMM-based text classification, from the initial step of preparing data to the final step of evaluating the classification model. In this section, the second step of the procedure, how a HMM-based classification model is built, will be dealt with, and described from the theoretical to the design and implementation level. In this dissertation, an automatic document classification system using a HMM as its classification framework is proposed, so as to categorize documents into LC classes. Constructing a HMM-based classifier can be divided into three components: (1) Preparing training data, (2) Designing the model, and (3) Training the model. The previous section covered the issues involved with training data, such as how the training data sources were selected and how the selected data was processed. In this section, the theoretical background of a text classifier will be provided, and then the two remaining components for building a HMM-based classifier will be described.

5.4.2 Model Training

Once the topology of a HMM consisting of a set of states and transition flows is determined, the next procedure is to parameterize the model variables such as emission probabilities and state transition probabilities, in a supervised learning approach, which is called a training process. A training process requires training data and a training algorithm.

Two databases, the OCLC CatCDTM Book for Windows database and the PQDD database, have been involved and used for preparing training data sets, as stated in the previous section for this procedure. Five hundred dissertation abstracts covering three top-level LC classes (Science, Agriculture, and Fine Arts) and their twenty-five second-level subclasses are arranged with twenty abstracts per category. Furthermore, the subject-

heading descriptors from the 102,650 MARC records are collected to set up another information source.

The main aim of applying a training algorithm to a model is the realization of the transition probabilities and emission probabilities of the model. In the general case of training models, especially HMMs with incomplete training data referring to the situation of including data unlabeled for their categories, the Baum-Welch algorithm as a special case of the EM method (Dempster et.al., 1977) has been widely used for training the variables in the HMM (Rabiner, 1989). The training data for this study, however, come accompanied by their corresponding class labels, which is the reason this training process is called a supervised learning. With supervised data, the emission probabilities of output symbols in a state can be estimated by the ratio of the number of occurrences of a symbol to the total number of all the output symbols. The estimation of an emission probability for the symbol W_i at the state S_j is computed by:

$$P(W_i / S_j) = \frac{N(W_i, S_j)}{\sum_{k=1}^{|V|} N(W_k, S_j)}, \quad (5.1)$$

where $|V|$ is the total number of distinct terms in the state, and $N(W_i, S_j)$ refers to the frequency of the symbol W_i at the state S_j .

The problem of producing a symbol emission probability using this expression (5.1) lies in the fact that it might generate zero probability if a symbol does not occur in the training data. In order to prevent non-occurring symbols from being assigned zero probability the n-estimate probability (Mitchell, 1997) is adopted that provides a constant probability in proportion to the total number of symbols by:

$$P(W_i / S_j) = \frac{1 + N(W_i, S_j)}{|V| + \sum_{k=1}^{|V|} N(W_k, S_j)}. \quad (5.2)$$

In general, the estimation of state transition probability in HMM can be made by simply counting transition occurrences between states with labeled data, or can be obtained by using the maximum likelihood process led by EM (Rabiner, 1989) with unlabeled data.

Such methods for parameterization, however, will not be effectively applicable to our model. In the methods referred to, the estimation of transition probabilities can be interpreted to be a statistical reflection of the relationship among different states. Since our model uses a non-dummy state to represent an information source, the relationship between states can be rephrased as the one between different information sources. What is considered as training data is two separate data sets from different sources, which does not seem to have relevant information about their relationship.

According to the new proposed method for the estimation of transition probabilities, the role of the transition probability from a state A to a state B is newly defined as the weight to be added to the state B, as measured by the average amount of information output symbols (words) occurring in a state B. In other words, a transition probability from a state A to a state B is regarded as the amount of information that an information source B (the state B) has. The same principle can be applied to initial transition probabilities between dummy states and regular states as well as transition probabilities between regular states. Thus, a central issue related to this scenario is how a quantity of information possessed by an IS can be computed. TF and IDF concepts (Salton & Buckley, 1988) have been popularly used in the IR field as a measurement of a word's degree of importance at both document and corpus levels. These well-known concepts are adopted due to their simple but powerful features. They are used to compute the quantity of information that each IS in a different subject context holds. The expression (5.3) displays the process of calculating initial probabilities, based on TF and IDF:

$$\begin{aligned}
 I(C_i) &= \sum_{w \in C_i} I(w) = \sum_{w \in C_i} TF(w, C_i) IDF(w) \\
 Inormal(C_i) &= I(C_i) / |C_i| \\
 P(C_i) &= \frac{Inormal(C_i)}{\sum_{\forall j} Inormal(C_j)},
 \end{aligned} \tag{5.3}$$

where C_i refers to a collection of information sources, regardless of categories and the function I refers to an estimated quantity of information. According to (5.3), the initial probabilities in all the categories are the same, from a start state to a specific regular state.

Therefore, the initial probability of a state can be rephrased as the entire quantity of information found in an information source corresponding to a state.

The expression (5.4) describing a state transition probability is very similar to that of an initial probability, except that the complexity of categories is added. When the quantity of information in an IS is computed for the estimation of a state transition probability, rather than using an IS for all the categories as in initial probabilities, only a category-relevant IS is considered by (5.4). In either case, they are normalized according to all other types of category-free ISs as in (5.3) and category-relevant ISs as in (5.4).

$$\begin{aligned}
I(C_i^k) &= \sum_{w \in C_i^k} I(w) = \sum_{w \in C_i^k} TF(w, C_i^k) IDF(w) \\
Inormal(C_i^k) &= I(C_i^k) / |C_i^k| \\
P(C_i^k) &= \frac{Inormal(C_i^k)}{\sum_{\forall j} Inormal(C_j^k)}.
\end{aligned} \tag{5.4}$$

Many variations of TF and IDF algorithms have become available, and have been used in many different settings. For our purposes, the TF and IDF version that has been adopted by the popular IR system called Okapi (Roberson *et al.*, 1995) and other several IR systems (Ponte & Croft, 1998; Miller *et al.*, 1999) is used. For the sake of completeness, it is reproduced here:

$$TF(t, d) = \frac{tf(t, d)}{tf(t, d) + 0.5 + 1.5 \frac{l(d)}{AveNum}}$$

$$IDF(t) = \frac{\log\left(\frac{N + 0.5}{df(t)}\right)}{\log(N + 1)}$$

$TF(t, d)$ = the modified version of term frequency

$tf(t, d)$ = number of the term t in document d

$l(d)$ = total number of terms appearing in document d

$AveNum$ = average number of terms of a document in the corpus

$IDF(t)$ = the standard version of inversed document frequency

N = total number of documents in the corpus

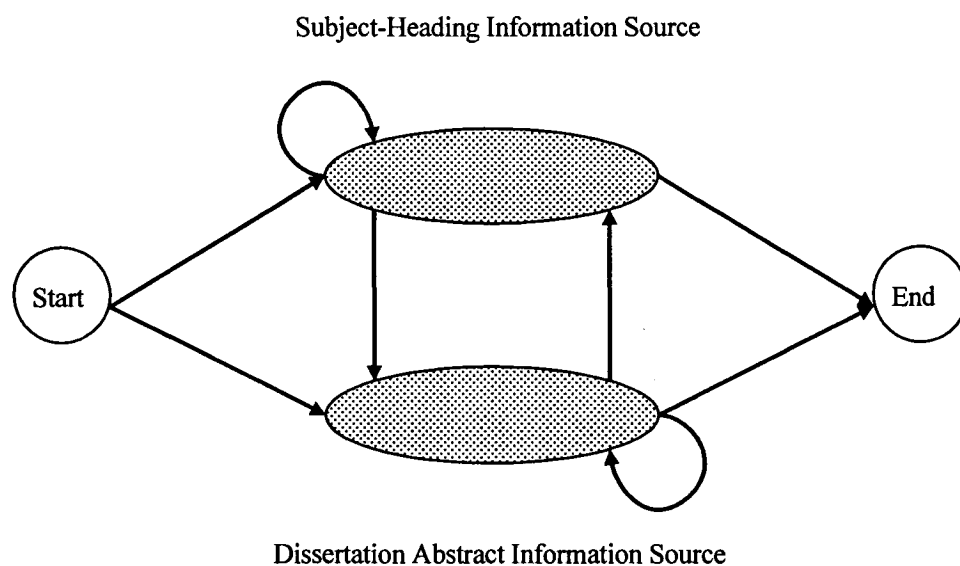
$df(t)$ = number of documents in the corpus having the term t

5.4.3 Experimental Settings

The experiments performed in this study were applied to the proposed HMM-based TC system that supports the implementation of some selected LC classes as its classification basis, with a set of randomly collected dissertation abstracts consisting of 625 documents being evenly distributed amongst three top-level LC classes. A HMM-based TC model was designed and trained in accordance with the procedures and methods discussed in the previous chapter. Figure 5.9 displays the trained HMM for the subclass QR (Microbiology) with initial and transitive probabilities. Also, the data upon which the proposed model will be tested, and the methods of measuring model effectiveness are also described in the previous two sections.

In this study, twenty-five HMMs were trained with each implemented for a specific target class (subject), given a training data set. Thus, the implementation of the 5-fold cross validation requires one hundred twenty-five HMMs to be trained, being 25 multiplied by 5. One remaining issue before implementing the next step is the problem of how to organize the HMMs. Since the set of LC classes implemented by HMMs forms a hierarchical structure, it seems natural that the HMMs should be constructed in the same way that the LC classes are connected. For the experiments used in this study, however, a

flat structure is considered, where the twenty-five HMMs are in the same level. Let the hierarchical classification structure of the twenty-five classes be T_0 . T_0 has a tree structure having a root at the top, which is a starting point of the tree, and leaves at the bottom, each of which refers to the classes of interest. The machine classification performed in the experiments in the following section is performed in a bottom-up way because the classification process results in assigning abstracts to leaves. That is, the extent of inclusiveness for an abstract of each class in the leaves, rather than any intermediate node referring to a broader class, is measured. For that reason, the conceptual distance from an abstract to any broader class is indirectly estimated based on the results to the leaves. In summary, the twenty-five LC subclasses are a hierarchical structure, called a tree, and the tree's leaves are the only classes implemented with HMMs.



- Initial probabilities:
 $\{\text{Start}\} \rightarrow \{\text{SH}\}: 0.74$
 $\{\text{Start}\} \rightarrow \{\text{DA}\}: 0.26$

- Transitive probabilities:
 $\{\text{DA}\} \rightarrow \{\text{SH}\}: 0.61$
 $\{\text{DA}\} \rightarrow \{\text{DA}\}: 0.39$
 $\{\text{SH}\} \rightarrow \{\text{DA}\}: 0.39$
 $\{\text{SH}\} \rightarrow \{\text{SH}\}: 0.61$

[NOTE]

* Transitive probabilities going into the End state are zeroes.

Figure 5.9: A trained HMM for the subclass QR (Microbiology)

5.4.4 Model Inference

So far the design of a TC model and the building of the TC system based on the HMM has been discussed. Next, the principles and procedures of document classification will be described. A set of documents based on the ProQuest digital dissertations (test documents) is prepared for testing our model. Each document in the test dataset, already classified by a professional, is labeled with a LCC category. For testing the proposed TC model, given a test document d , a HMM, designed to represent a LCC category c , produces a probability. This probability is then used to measure the relevance between a document d and a category c , as described in section 3.1. The same procedure is used to calculate relevancy probabilities between documents and LC categories with other HMMs. The result is a list of relevant probabilities for all the documents and categories.

There are two concerns that should be answered in the classification process: how to define the relevancy between a subject category and a document, and how to measure that relevancy. First, as previously explained in section 3.1, in our model a document is viewed as a list of words, and a TC system is considered to be a model, which generates words. Relevancy is defined as the probability of similarity between the terms in the document and the predefined subject categories. The similarity between a document and a subject category is measured by the probability of a list of HMM generated terms in the document. The TC model produces a sequence of words along with the probabilities corresponding to the words. In the HMM, given a list of output symbols, there are numerous possible paths of states producing the same output symbols. Thus, more formally, the probability of a HMM TC model given a list of terms is the summation of all the probabilities from different paths that were taken to produce the terms. Second, the estimation of relevance is the process by which a probability indicating the relevance between a document and a category is obtained. Given N being the number of states and T being the number of terms in a document, the direct method of calculating the probability requires $\Theta(NT)$ multiplications (Rabiner, 1989), which poses logistical problems even with a small number of N and T . For the estimation of the probability, our model uses an efficient method based on a dynamic procedure referred to as the forward algorithm (Baum & Egon, 1967; Rabiner, 1989).

Chapter 6

EXPERIMENTS AND THEIR RESULTS

6.1 Introduction

This chapter primarily covers the analysis and discussion on the performance of the proposed HMM-based classification model. Prior to the discussion, the delimitations and limitations of this study will be outlined. After that, the performance evaluation of the proposed classification system will be described including the following topics: (1) experimental setting, (2) experiment process, and (3) measurement and evaluation. The experimental setting includes the data and type of categories used in model training and testing. The experiment process covers the means of integrating all the experimental components for the experiments. The measurement and evaluation deals with the topics of model performance and tools for its measurement.

6.2 Delimitations and Limitations

The delimitations and limitations are discussed in this section. The delimitations address the factors that confine the scope of this study and the limitations address the factors that provide inherent limitations to this study.

6.2.1 Delimitations

The scope of this study is limited by a combination of the availability of data and various logistical reasons as well as the nature of the machine learning paradigm for the following issues:

- One of the delimitations of this study is in implementing only a few levels of the LC classification hierarchy, rather than all the classes in the LC outline as in the Pharos project. The LC classification outline has a tree structure with a root node. There are twenty-one classes at the top-level in the LC classification trees, two of which may be combined into a single class (class E and class F indicate the same subject, American History). At the second level, there are 222 subject sub-classes. In the third level, it is known that there are 4214 subject classes (Dolin *et al.*, 1998). In this study, the implementation of the LC classification tree is limited to the second level of the LCC tree. This is largely due to practical considerations of the processing time needed for building such a large number of classifiers. The nature of machine learning is partly related to this limitation because building each classifier under this approach requires a relatively significant amount of manual processes compared to IR approaches used by Pharos and Larson's work. However, this study is focused on the construction of a model for automatic classification, rather than on the automatic construction of a model. Since the model is being built manually, the time for training the model increases as the size of the model expands.
- Secondly, it should be noted that this study has no intention of having special interest on classifying dissertation abstracts. The objective in this application is to investigate the performances of machine classification on different subjects to see if it acts as do human beings who might feel various levels of difficulty over subjects. Therefore, a source of data that is pre-classified by professionals who could bring reliability to data had been sought. Dissertation abstracts are believed to be the most reliable source available and that is the only reason why they were selected as the data for this study. Thus, the availability of data was the crucial factor giving a limitation to the choice of data.

6.2.2 Limitations

The limitations associated with this study are summarized as follows:

- Theoretical assumption of our model: This study has introduced HMM as a new TC model for the task of Text Classification. HMM is a statistical model that characterizes a real-world process (TC in our case) as discrete parametric random processes. This parametric stochastic model is based on the underlying assumption that the full statistical description of a system is restricted to the conditional probability of current events to previous events. In other words, considering a current event, all previous history information is disregarded except for the previous one adjacent to the current. This is called the first order Markov condition.
- Limitation on the task of classification: The ML paradigm in which this study resides requires having training/test data sets classified correctly, in the sense that given a set of pre-defined categories, the most relevant category is assigned to a document. Since the nature of the task of assigning a category is highly subjective, the pre-classified data used for building and testing the TC system has an inherent limitation.
- Assumption on the level of classification: LCC has a hierarchical structure from general to specific. A LC class number assigned to a document specifies a certain level of the LC structure. Meanwhile, the second level of LCC was set to be the target level when the systems were built and evaluated in this study. Therefore, there are mismatches between the target level to be considered and the level of classes that are really assigned to documents and used for the experiments. In most cases, the classes assigned to the training/test data sets refer to more specific level of classes than the second level. Thus, it is implicitly assumed that documents in data sets belong to the second level of classes although they were assigned to a deeper level.

6.3 Use of Data Sets and Baseline Classifiers

The method and procedure for the selection and collection of training and test data were previously described. In this section, primary focus will be given to an approach for the proper use of the collected data. This is an important issue in ML algorithms in that an inappropriate use of data may cause an inappropriately trained system resulting in the incorrect evaluation of the system's performance.

6.3.1 V-fold Cross Validation

The V-fold cross validation method is a popular method of estimating the generalization error of a given model, especially for small data sets (Goutte, 1997). In V-fold cross validation, a collected data set is divided into V groups of equal size (or as close as possible.) By eliminating one out of V groups, the rest (V-1 groups) constitute a training data set, and the eliminated set becomes a test data set. By repeating the same process of selecting a different group for elimination, V training and test sets can be prepared. The only concern with this method is the determination of the value of V. Although it is still debatable and under discussion, some works show the better performance with the value of 10, 5, or even 2, than others (Breiman & Spector, 1992; Kohavi, 1995; Shao, 1993).

In our experiments, six hundred twenty-five abstracts were collected for the twenty-five LC categories, resulting in twenty-five dissertation abstracts being placed into each category. In previous studies, a range between 2 and 10 was suggested as a value of V for V-fold cross validation. For the twenty-five documents in our training and test data to be equally divided, five is the only choice between the integers of 2 and 10 for the value of V. The selection of a value for V is more of a practical issue than a theoretical one. Thus, building systems using various options for V values may be experimentally explored and compared. This will be discussed later in the Further Work section. In short, given a LC category, the twenty-five collected abstracts were randomly divided into five groups, due to the choice of five for V, with each group consisting of five abstracts. According to the 5-fold cross validation, five pairs of training and test sets were prepared, depending on the selection of a different group for a test set, with each training set consisting of 20 abstracts

and test sets containing 5 abstracts. Therefore, the proposed HMM-based TC system will be trained as many as five times, based on five different sets, and will be tested accordingly. The following pseudo code explains the procedure of implementing the 5-fold cross validation:

```

For each category  $C_i$ ,  $C = \{C_1, C_2, \dots, C_{25}\}$ 
    CATEGORY =  $C_i$ ;
    SET = 25 documents collected for the category  $C_i$ ;
    V = 1;
    Repeat until size(SET) is equal to zero;
        S = 5 documents randomly selected from SET;
         $C_i(V) = S$ ;
        size(SET) = size(SET) - 5;
        V = V + 1;

```

6.3.2 Baseline Classifier

A version of a NB classifier is employed and applied to the same task as the HMM-based classifier due to our need of producing a baseline performance for comparison purposes. There are several reasons for the selection: (1) A NB-based classifier has been popular in TC due to its low computational complexity and good performance in practice (McCallum & Nigam, 1998), (2) it has been adopted in many TC comparative studies (Dumais *et al.*, 1998; Joachims, 1997; McCallum & Nigam, 1998; Yang & Liu, 1999), and (3) it has performed very well in some cases (Dumais *et al.*, 1998). Among the variations of its versions, the algorithm used in Joachims' (1997) TC work is adopted in this study, which is described in Mitchell's ML book (1994). In his version, the Laplace estimator suggested by Vapnik (1998) is employed for the task of calculating the probability of a term in a category. He claimed this version worked well in practice.

6.4 Model Effectiveness

6.4.1 Introduction

In this section, a popular method widely used on IR for the measurement of TC model effectiveness will be explained, which will be also adopted for our TC model:

classification accuracy. The technique for evaluating the performance of IR systems was invented and mainly used for a binary-valued class problem making a decision based upon two possible choices, such as relevance or non-relevance to a given query by web search engine systems. This technique can be extended to cover a multi-valued class problem, having more than two possible values, as this study has been proceeded using the twenty-five categories under three top-level LC classes.

Given a multi-valued TC problem, let's assume $C = \{C_1, C_2, \dots, C_n\}$ to be a set of n target classes. Table 6.1 shows a distribution of *predicted* data (examples or instances) into n *actual* target classes. The first row shows a list of n *actual* target classes. In each of the following rows, the number of test data *predicted* as a certain class by a TC model are shown and distributed into the corresponding *actual* target classes. For example, in the row for the predicted class of $C_i, 1 \leq i \leq n$, the total number of test data *predicted* for the class C_i is equal to the sum of the numbers over all the columns, denoted by $\sum_{j=1}^n D_{ij}$, and the value D_{ii} is the number of data that are correctly classified for the *actual* class C_i . Thus, the sum of the other cells in the same row is equal to the number of data incorrectly classified, denoted by $\left(\sum_{j=1}^n D_{ij}\right) - D_{ii}$. Using Table 6.1, a description of the model estimation methods will be presented in the following sections.

	C_1	C_2	...	C_n
The number of documents predicted as C_1	D_{11}	D_{12}		D_{1n}
The number of documents predicted as C_2	D_{21}	D_{22}		D_{2n}
The number of documents predicted as C_n	D_{n1}	D_{n2}		D_{nn}

Table 6.1: Contingency table for the multi-valued classes $C = \{C_1, C_2, \dots, C_n\}$

The second-level classes under three LC classes (Q, S, and N)	HMM-based TC system	NB-based TC system
Q – Science (General)	1.7	16.9
QA – Mathematics	1.2	20.7
QB – Astronomy	2.4	22.7
QC – Physics	2.2	18.6
QD – Chemistry	1.7	18.3
QE – Geology	4.2	21.0
QH – Natural history – Biology	5.5	20.6
QK – Botany	5.6	22.4
QL – Zoology	8.9	15.0
QM – Human anatomy	1.6	14.8
QP – Physiology	3.4	19.8
QR – Microbiology	1.6	24.8
S – Agriculture (General)	1.8	9.8
SB – Plant culture	3.4	23.5
SD – Forestry	2.4	16.2
SF – Animal culture	4.9	21.9
SH – Aquaculture. Fisheries. Angling	3.9	19.0
SK – Hunting sports	2.1	22.2
N – Visual arts	2.4	16.2
NA – Architecture	2.8	20.0
NB – Sculpture	1.9	21.2
NC – Drawing. Design. Illustration	1.8	13.3
ND – Painting	2.6	18.6
NK – Decorative arts	4.9	24.7
NX – Arts in general	1.1	14.4

Table 6.2: Comparison of the actual category rankings from two TC systems

6.4.2 Classification Accuracy

Classification accuracy is a method used for the estimation of model effectiveness in terms of the correctness of classification. It is measured by the probability of the number of correct classifications over the total number of classifications. Classification accuracy of a model M on a test data set T is measured with the formula below (Mladenic, 1998):

$$\begin{aligned} Accuracy(M, T) &= \sum_{e \in T} P(e) \times Correct(e) \\ Correct(e) &= \begin{cases} 1 & \text{if } C(e) = \hat{C}(e) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (6.1)$$

where $P(e)$ is a probability of a document (example or instance) e that is usually equally considered for all documents, that is, $1/N$ with $N = |T|$ being the number of documents in the test data set T , $C(e)$ is the *actual* class of a document e , and $\hat{C}(e)$ refers to the *predicted* class by the model M . It is assumed that $P(e)$ is a constant to all documents. If the model M perfectly estimates the actual classes of all the test documents, then the value of $Accuracy(M, T)$ in (6.1) becomes 1, $\sum_N \left(\frac{1}{N} \times 1 \right) = 1$. On the contrary, if the model does not correctly estimate the actual class of a single document at all, the classification accuracy becomes 0, $\sum_N \left(\frac{1}{N} \times 0 \right) = 0$. As a result, a classification accuracy value based on (6.1) is between 0 for the worst expectation and 1 for the perfect expectation.

Given a multi-valued class problem and its results as shown in Table 6.2, with the test data T , the classification accuracy calculated by the expression of (6.1) can be visually seen as the sum of the values on the diagonal line in Table 6.1., and is equivalently represented in a formula, as shown in (6.2):

$$Accuracy(M, T) = \frac{\sum_{i=1}^N D_{ii}}{\sum_{i=1}^N \sum_{j=1}^N D_{ij}} \quad (6.2)$$

6.4.3 Classification Accuracy Confidence

As mentioned in the previous section, classification accuracy is a method that measures a TC system's capability of finding the most relevant category, with the assumption that it has been identified among pre-set categories, and pre-assigned as a label to each of the test documents. A question concerning the evaluation of this measurement may be raised: How can the classification accuracy method consider the ranking of the actual category in a system-generated ranking list?

Given a document in a test set, the TC system generates a list of possibly relevant categories with relative rankings. Suppose a test document d_i has its category, $C(d_i)$, pre-assigned to the document. When the *actual* category, $C(d_i)$, is top-ranked in the list, this is considered to be an indication that the system has correctly found the most relevant category for this document. There is no question about the interpretation in this situation. This being said, how does classification accuracy measure the performance of a text classification system? When a TC system classifies a number of documents, and none of the *predicted* categories for the documents is top-ranked, the classification accuracy of the system is calculated as zero if only top-ranked *predicted* categories are considered to be correct, even in the case that they are all second-ranked. For instance, consider two TC systems, A and B, with five test documents and ten target categories. Let $L(A)$ be the list of rankings of the *predicted* categories for the performance of the system A, and let $L(B)$ be the same for the system B. If $L(A) = \{1, 10, 10, 10, 10\}$ and $L(B) = \{2, 2, 2, 2, 2\}$, then the classification accuracy of the system A is $\frac{1}{5} + 0 + 0 + 0 + 0 = \frac{1}{5}$, and that of the system B is $0 + 0 + 0 + 0 + 0 = 0$, with the rule of the top-ranked. In the example above, only the top-ranked predicted categories are taken into consideration towards the measurement of classification accuracy, and other rankings of the system-generated list are disregarded though non-top ranked categories need to be considered for the accuracy method. Therefore, the concept of confidence tolerance will be considered in measuring the classification accuracy in order to expand the ranking range of interest, rather than concentrating on top-ranked categories only. The confidence N in classification accuracy may be defined as a window of top- N ranking such that if an actual category appears

within the window, then it is considered to be accurate. Consequently, when a predicted category is ranked within a range of confidential tolerance, it is said to be correctly classified within the boundary of the value of a confidential tolerance which will be referred to as the ‘alpha value’ later. For example, the alpha value is set to 3, which means that the third-ranked rule will be applied, rather than the top-ranked rule.

6.5 Experimental Results

6.5.1 Introduction

Given a test document (a dissertation abstract) two TC systems involved in experiments - HMM- and NB-based systems – classify it according to the twenty-five categories, and produce a ranked list of the considered categories in decreasing order of relevance (the ranking of *predicted* categories). The NB algorithm implemented in other TC task (Joachims, 1997) is used for the NB classifier implemented for this study. Since the *actual* category of a test document is pre-known, the location of the *actual* category can be found on the ranking of the predicted categories, thereby being a barometer for the accuracy of the system’s performance. The higher the position of an *actual* category, the better the system has performed. According to the 5-fold cross validation rule, five different test sets, each consisting of five documents, are tested, and in total, twenty-five documents are used for the purpose of testing for each category.

The experiments in this study will be explored with two main purposes in mind: (1) How does the proposed HMM-based TC system perform, in comparison to a NB-based TC system? (2) How differently does the HMM-based system work in classifying documents from two distinct disciplines? The experiments performed will be discussed in the following sections.

6.5.2 HMM-based system vs. NB-based system

As the first step of the performance analysis, the probabilistic distributions of the rankings of the actual categories by two different systems are displayed in Figure 6.1 and their

cumulative versions are in Figure 6.2. Figure 6.1 plots the probabilities of the ranking frequencies, rankings referring to the values of the locations where actual categories are proposed by systems. As for the output of the HMM-based system, more than 70 percent of all the cases are ranked within the rankings of the 1st, 2nd, and 3rd, whereas for the NB-based, greater than 55 percent falls out of the 20th rankings. Figure 6.2 depicts the cumulative effects of Figure 6.1. The two systems' completely different trends are shown. The density function for the HMM-based system quickly approaches a value of 80% at the alpha value of 4, whereas the same value is attained at the alpha value of 24 for the NB-based system, which clearly shows that the HMM-based system outperforms the NB-based system.

Table 6.2 shows a chart comparing the performances of two TC systems. The figures in the second and third columns of the table refer to the ranking of the actual categories, averaged for twenty-five documents, by the HMM-based and NB-based systems, respectively. When reading Table 6.2, it is worth remembering that a rank of 1 indicates a perfect classification. Overall, the averaged rankings by the HMM-based system are scattered between 1.1 and 8.9, whereas those by the NB-based system are between 9.8 and 24.8. The worst case in the HMM-based system performs better than the best case in the NB-based system, indicating that the HMM system outperforms the NB system in all of the twenty-five classes. With the HMM system, actual categories appear in top-third rankings in about 64% of the classes (16 out of 25), and top-fifth rankings in about 88% (22 out of 25). Among the three classes residing out of the boundary of the top-fifth, two actual rankings are very close to the fifth (5.5 and 5.6), and only one actual ranking is far away from the boundary. Shifting the focus to the NB system, the best ranking among the twenty-five is 9.8. A majority number of the actual rankings (56%; 14 out of 25 cases) falls in between 15.0 and 20.0 inclusively. As described, in terms of the averaged actual ranking, the performance of the HMM system by far exceeds that of the NB system in all the classes considered.

Next, the analysis of two TC systems' performance, based on classification accuracy, is studied according to three aspects: (1) second-level class, (2) top-level class, and (3) model. Our definition of classification accuracy, as explained in section 6.3.2, refers to a

real number between 0 for the worst case and 1 for the best case. Thus, getting closer to 1 indicates better performance. Figure 6.3 displays classification accuracy graphs over the twenty-five second-level LC classes, performed by the HMM system. Five different accuracy graphs are outlined by applying five different alpha values (refer to 6.3.3 for the definition). As the alpha value increases in number, the corresponding accuracy graph quickly moves close to the line of $y=1$. In the accuracy graph for the alpha value of 3, an accuracy rate of at least 0.8 is obtained in about 64% (16 out of 25) of the twenty-five second-level classes. In the accuracy graph for the alpha value of 5, it is achieved in about 72% (18 out of 25), and for the alpha value of 10, about 96% (24 out of 25). Figure 6.4 displays accuracy graphs for the NB system using the same alpha values as in Figure 6.3. In the accuracy graphs for the NB, an accuracy of at least 0.8 is never attained with the alpha values of 3, 5, or 10. In addition, the perfect accuracy of 1 is not a reachable point even with the alpha value set at 20.

Figure 6.5 shows classification accuracy graphs over the three top-level LC classes for both TC systems. The accuracy data for a top-level class, T_i , are obtained from an average of all the accuracies for the second-level classes belonging to the T_i . That is, let $T_i = \{T_{i1}, T_{i2}, \dots, T_{in}\}$ be a set of second-level classes of the top-level class i , and the classification accuracy for T_i at Figure 6.5 is obtained, as shown in (6.3):

$$Accuracy(T_i) = \frac{\left(\sum_{j=1}^n Accuracy(T_{ij}) \right)}{n} \quad (6.3)$$

In Figure 6.5, the first five graphs on the top indicate accuracy graphs for the HMM-based system, and the remaining bottom graphs refer to those for the NB-based system. In all the three top-level classes, the performance of the HMM-based system exceeds that of the NB-based system in all alpha values. Particularly, the HMM-based performance begins with about 0.7 accuracy and goes up, as alpha values are increased, whereas the NB-based performance does not pass beyond the line of 0.5 accuracy in any class even with the highest alpha value.

Figure 6.6 shows the overall accuracy graphs for the two systems, according to alpha values. Similar to the accuracy for a top-level class, the overall accuracy of a system is calculated by averaging the accuracies for the top-level classes. Let $T = \{T_1, T_2, \dots, T_t\}$ be a set of top-level classes involved in this experiment. The overall classification accuracy for a TC system, M , in Figure 6.6 is obtained by the following expression in (6.4):

$$Accuracy(M) = \frac{\left(\sum_{i=1}^t Accuracy(T_i) \right)}{t} \quad (6.4)$$

According to the overall accuracy graph for the HMM system, the system performance lies in between 0.77 and 0.99. The other graph shows that the overall performance of the NB system is measured between 0.24 and 0.46. In addition to the different starting accuracy points, the two accuracy graphs have different trends in slope changes. In the graph for HMM, a relatively sharp change in slope occurs in the changes of alpha values from 3 to 5 and 5 to 10. However, in the NB accuracy graph, steeper slopes occur in the range of 10 to 15 and 15 to 20. Having a steeper slope in an alpha value range means having more documents whose actual category rankings are in the same alpha value range. Therefore, it can be concluded that, in the HMM system, the actual categories of a significant number of test documents occur in top-three rankings, and those of a relatively small number of documents are in top-five or top-ten ranking. In the NB system, there are a considerable number of test documents for which actual categories are found in top-fifteen and top-twenty rankings, and it is relatively rare to see documents whose categories are in top-three or top-five rankings.

The performance of the two TC systems (HMM and naïve Bayesian) has been compared according to individual class and broader disciplines. The results unearthed in this experiment have shown that the performance of the HMM-based TC system is superior to that of the NB-based. One factor for this difference may be attributed to the training data used in both systems. It has been mentioned in the previous section outlining the experimental setting that the same set of labeled dissertation abstracts were used in training both systems, and that additional data (subject headings), were applied only to

the training process of the HMM system and not to that of the NB system. The HMM theory has a description of the principles and regulations on model design. One of the purported strengths of HMM is the flexibility in model structure, which leads to the ability of incorporating multiple sets of data from different sources. Therefore, we added this extra data set to see if the feature of integrating different data sets would contribute to improve the model's effectiveness. In this study, the experimental results with a combination of multiple data sets only are reported. Further experiments, without the addition of the subject heading data to the HMM, may show the effects of this inclusion. Also, further work remains to be done in order to confirm the relationship between the inclusion of more data sources and the effectiveness of the model using the HMM. Therefore, it is not clearly unveiled the extent of contribution with the additional training data for the HMM model to the enhancement of the performance at this point.

The NB-based TC model is a TC framework solely based on a simple assumption that the attribute values of an example are independent of the class of the example. As the structural design of NB-based model is concerned, any standard rules, constraints or restrictions do not exist so that there is no directive of formulating multiple data sets in a sense that data from distinct sources are processed and handled systematically, leading to the fact that multiple source data are not adopted in the NB-based system implemented for this experimental study.

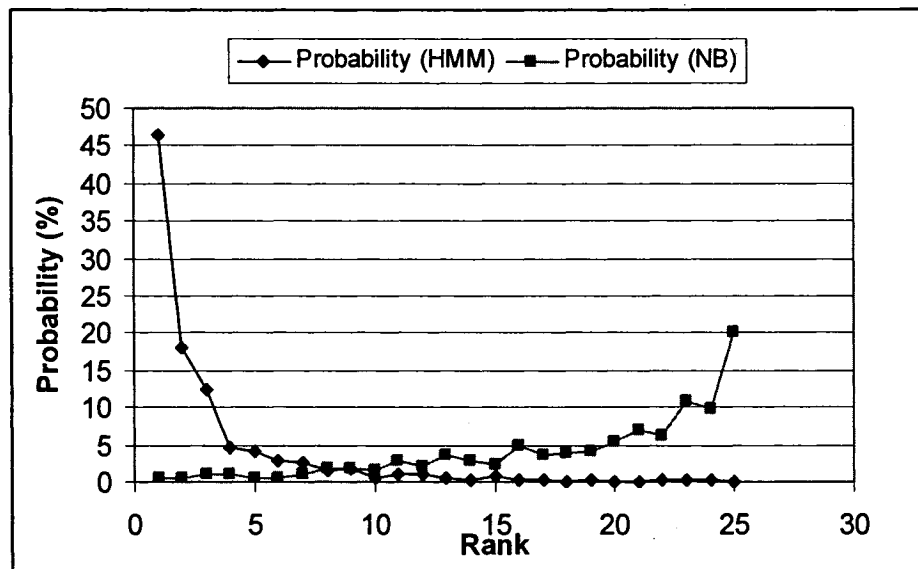


Figure 6.1: Probability mass functions of classification accuracy for the HMM-based and NB-based systems

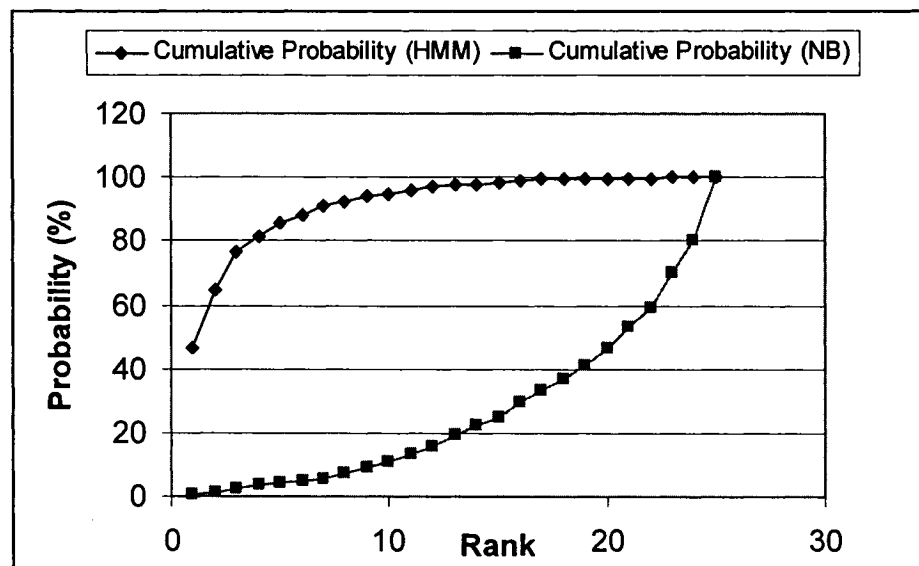


Figure 6.2: Cumulative probability functions of classification accuracy for the HMM-based and NB-based system

6.5.3 Hard vs. Soft Disciplines

Disciplines can be located on a spectrum of hard and soft disciplines and the semantic ambiguity of a document might relate to the discipline from which a document is derived (see Section 5.2.2). This statement seems to be convincing in that the semantic scope of terms used in softer disciplines appears to be broader, and that of terms in harder disciplines to be narrower. An interesting question related to this is whether Harter's claim can hold up when only a machine plays the role of understanding the contents of documents and deciding their relevant subjects. For these experiments, the 'Science' LC class (Q) and 'Fine Art' LC class (N) are selected to make representations of the hardest and softest disciplines, respectively, and the 'Agriculture' LC class (S) is perceived to be a discipline in between the previous two.

Figure 6.3 and 6.5 show classification accuracy graphs for the HMM system in the second-level and top-level classes, respectively. In Figure 6.3, the classification accuracies vary among second-level classes of a top-level class. In the 'Science' class (Q) with the alpha value (confidence value) equal to 3, the difference between the highest accuracy (1 for QA class) and the lowest accuracy (0.2 for QL class) is 0.8. In the 'Agriculture' class (S) with the same confidence value, the difference is 0.36 between SD class (0.88) and SF class (0.52), and in the 'Fine Art' class (N), it is 0.6 between NX class (1) and NK class (0.4). With the alpha value equal to 5, the differences are reduced to 0.64 for Q class, 0.2 for S class, and 0.4 for N class. Thus, the variation in accuracies becomes relatively small within the second-level classes of the 'Agriculture' class, compared to the other two classes. In Figure 6.5, various accuracy graphs for top-level classes, depending on diverse alpha value and top-level classes, are illustrated. A classification accuracy for a top-level class is calculated by the expression of (6.3). As seen in the graphs for a HMM system, in terms of averaged accuracy for a top-level class, the accuracy for the N class is found to be slightly higher than for the other two classes, when the confidence values are equal to 3 or 5. Also, classification accuracy has been almost equally increased over the top-level classes, when different confidence values have been applied from 3 to 5.

In comparing the accuracy of the three top-level classes, some evidences for supporting Harter's statements have not been found. On the contrary, the averaged accuracy for the N class is reported to be slightly higher than those for other two classes. What is the rationale behind this contradiction between the semantic ambiguity of terms from different disciplines and the obtained experimental results? It is generally believed by Harter that hard-disciplinary terms, such as specific technical ones, seem to be much less ambiguous in semantics than soft-disciplinary terms, such as ones describing feeling or emotion. I have observed that for a semantically less ambiguous term, it is presumably more difficult to find a semantically identical term, than it is for a more ambiguous term. For example, as an extreme case of a hard-disciplinary term, it seems impossible to find an alternative term to describe a specific gene name because the term is uniquely defined and used for a specific gene. Accordingly, the presence and absence of terms used for testing in the training data set might more sensitively influence hard-disciplinary terms, rather than soft-disciplinary ones. Since a document in training and test sets has an average of 311 terms, such an observation is applicable in seeking to understand the experimental results in this study.

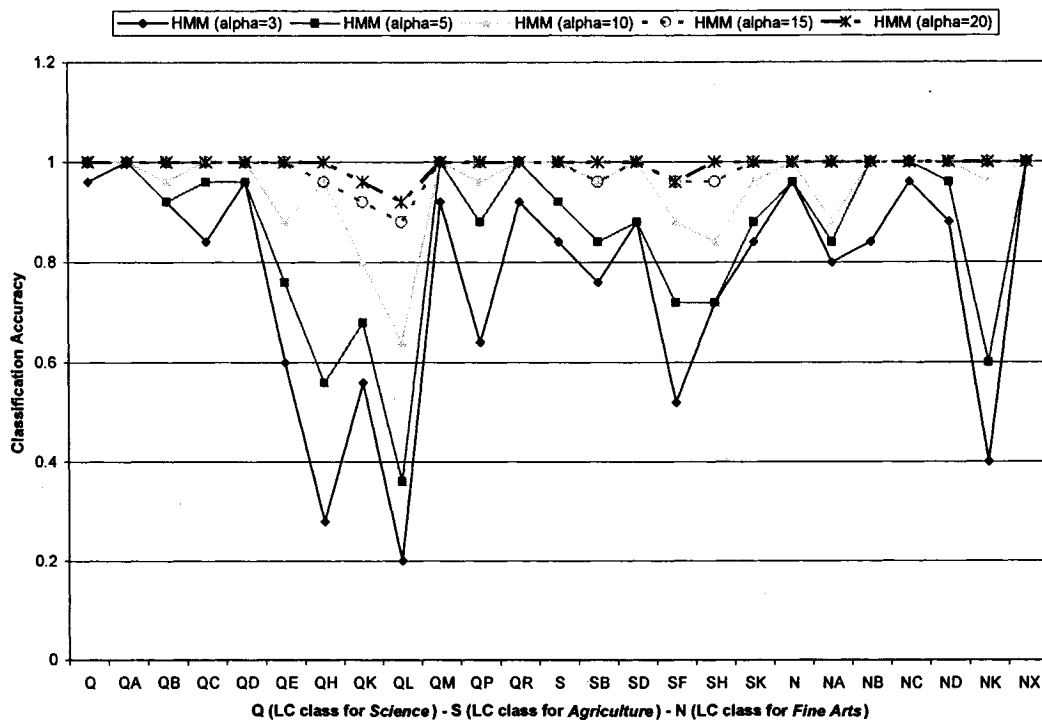


Figure 6.3: Classification accuracy graph for the HMM-based TC system

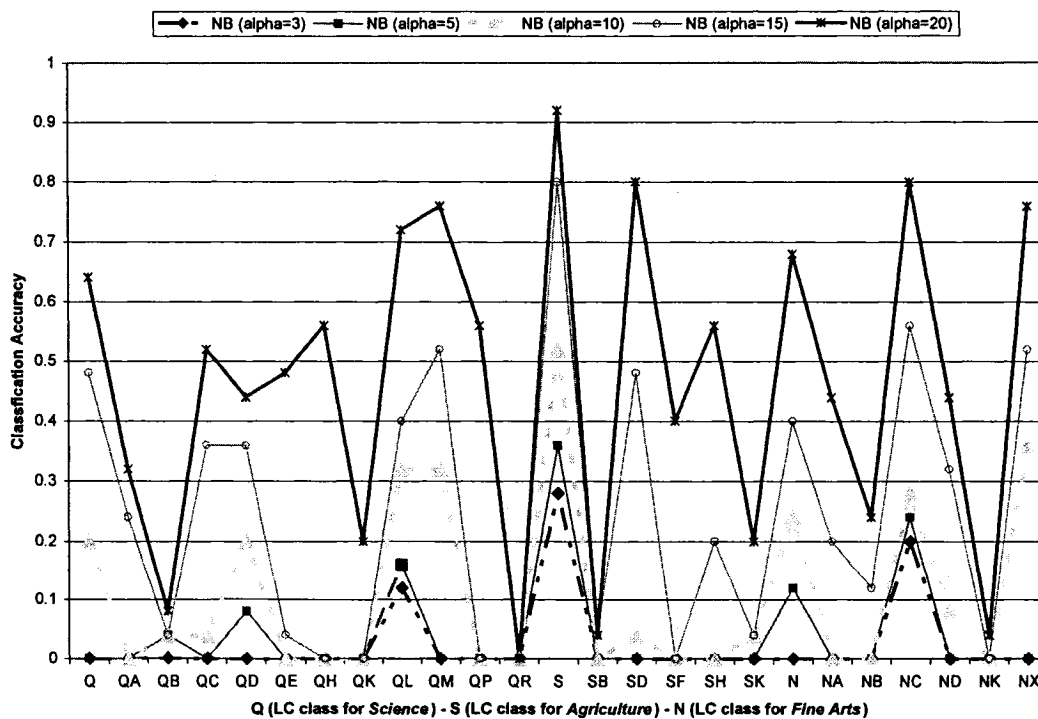


Figure 6.4: Classification accuracy graph for the NB-based TC system

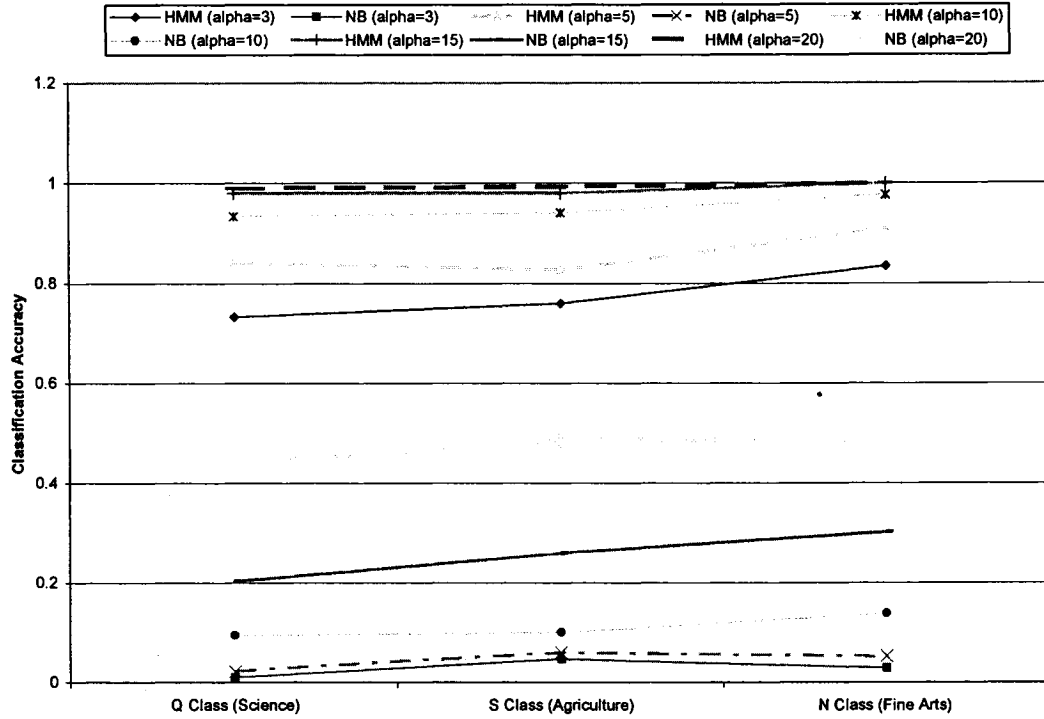


Figure 6.5: Classification accuracy graph for three top-level LC classes (Q, S, and N)

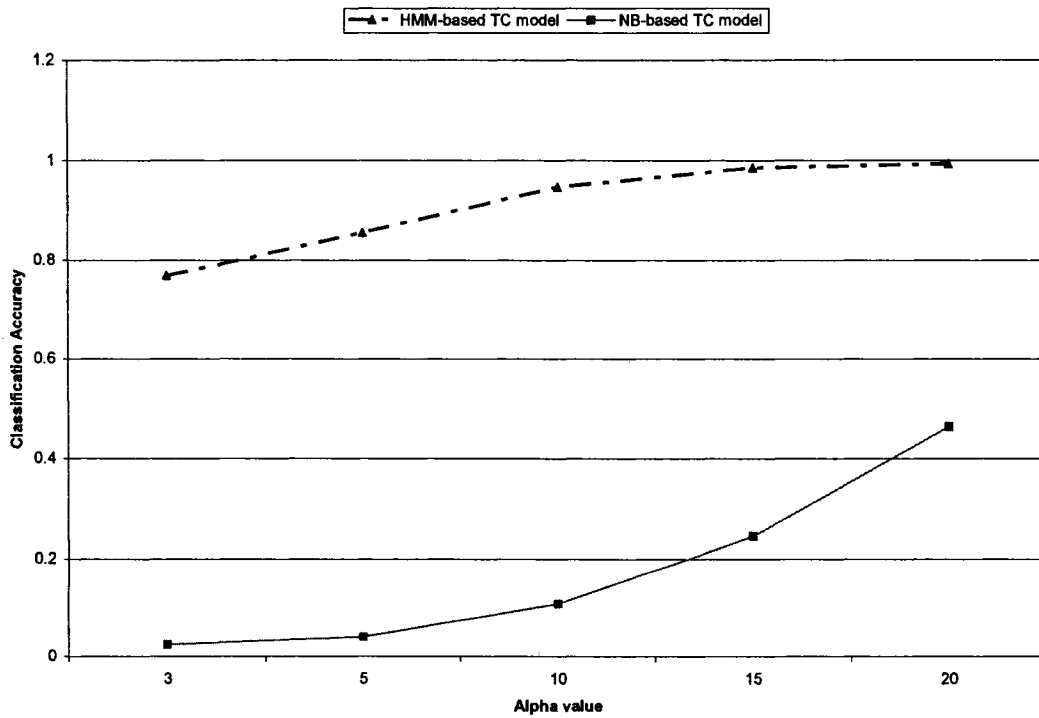


Figure 6.6: Trends of classification accuracies and alpha values

6.5.4 Analysis of Experimental Results

The experiments performed in this study have focused on exploring evidences related to two questions: the evaluation of the performance of the HMM-based TC system and the way in which a machine classification system, in this case, a HMM-based system, tackles the issue of semantic ambiguity inherited in documents. In the previous section, a summary of the experimental results was reported. In this section, the results of all six hundred twenty-five experiments involved are analyzed to investigate the factors that lead to these results.

With regards to the system's performance, the purpose of this analysis is to reveal factors affecting system performance in general. Fundamentally, the process of determining the most relevant category for a document starts with identifying whether or not terms in the document have been used in a training data set. If a term is not used for training, its role as a contributor to the classification process is of little significance. Its presence or absence, therefore, can be considered to be one of the factors affecting the performance of the system. Furthermore, though it is desirable to measure the extent to which a term contributes to a category being relevant, it is not measurable with the HMM-based system due to the complexity of its contribution. TF and IDF have long been used in IR fields as discriminating indicators of a term. They are adopted here to measure the extent to which a term is relevant to a category.

In each cross validation test, 5 documents are tested for each category and 125 documents in total are tested for all categories. Thus, for the 5-cross validation, 25 documents are involved in tests for a category and 625 documents (equivalent to the number of all the documents for testing) are involved for all categories. There are approximately 194,375 (625×311) terms in 625 documents (each document has an average of 311 terms). The resulting data from all experiments involved are summarized in Table 6.3. They are categorized into five performance levels, depending upon the ranking of actual categories appearing in the predicted category list. For example, if the actual category of a test document is presented among the top 5 in the system-predicted list, the test document is placed under the '1-5' column. Among these five levels, the '1-5' column in 'ranking of actual category' indicates the highest performance. About 86%

cases (535 abstracts) fall into this category. The ‘Number of matched terms’ row in Table 6.3 indicates the number of terms that occurred in both test and training documents. The data in this row displays a trend that the more common terms they have, the lower the ranking of actual categories. Also, a simple version of TF and IDF, term frequencies divided by document frequencies, was applied on the common terms to see if there is a relationship between TF and IDF and ranking. The averaged TF and IDF of the common terms for the test documents falling in the 1-5 ranking range is 1.42. This indicator represents how valuable the common terms are in terms of TF and IDF. By considering the two last rows, it is interesting that the TF and IDF value of a term in documents ranked higher is larger than in documents ranked lower. The data in Table 6.3 show that the system’s performance in ranking improves constantly, as not only the TF and IDF per common term has increased, but so has the total number of common terms.

Regarding the issue of semantic ambiguity, this analysis explores the observation described in section 6.4.3 as to how a machine interprets different levels of semantic ambiguities. A similar study to the previous work was done; the results are shown in Table 6.4. Regardless of the disciplines, as the ranking increases, it is common to see a reduction in the TF and IDF value and the number of hit terms, except in a few cases. The drop rate of TF and IDF (written in parentheses) for lower rankings is relatively high in the S and N classes when compared to the Q class. In fact, it seems that more meaningful terms in TF and IDF are concentrated on the highest ranking in the Q class, which can be linked to the alleged observation that the presence of a term affects performance more sensitively for the hardest discipline. The table shows the number of terms missed in each class (terms in test data which are not matched with terms in training data), depending upon different rankings. In the case of TF and IDF, however, the missed terms are equally treated as insignificant. The alleged observation for absence of terms is not traced in terms of TF and IDF.

NB-based classifiers were chosen as a classifier to provide a baseline performance for comparison in this study, mainly due to their popularity in many comparative TC research projects (Dumais *et al.*, 1998; Joachims, 1997; McCallum & Nigam, 1998; Yang & Liu, 1999). In the previous section, the performance yielded by NB-based classifiers was

reported to be much lower than the one by HMM models and thus was not comparable over all categories concerned. This is experimental evidence of the performance difference between two different models in TC in a controlled environment provided in this study.

A simple TF and IDF-based classifier may be anticipated to be compared with the HMM-based system, due to the fact that TF and IDF techniques were used in training HMM such as in obtaining emission probabilities (see Section 5.4.2). However, since the HMM theory has been applied to the proposed TC system as a principal foundation, and other IR techniques such as TF and IDF are employed as auxiliary methods in order to implement the HMM framework, the additional performance supported by the techniques can not be interpretable as an all-embracing power of the model but as a way of fine-tuning model parameters. Nevertheless, the performance of the system with the link of TF/IDF is discussed later and interesting results are represented in the Table 6.3 and 6.4.

In Chapter 2, some TC research projects that use library classification schemes have been reviewed. Of the related work discussed earlier, three projects are similar to our study in that, although their goals and methodologies are different, LCCs are used as organizational tools for classification and LCSHs extracted from MARC records serve as a primary data source for building their TC systems: (1) Larson's work (1992), (2) Pharos project (Dolin, 1998), and (3) Frank and Paynter's study (2004).

From the perspective of application, the biggest difference between our study and these three related works is in the different types of objects to which the use of LCSH and LCC has been applied. In related works, MARC records were classified based mainly on *title* and *subject headings* information, whereas in this study, Dissertation Abstracts, which are more lengthy and descriptive than *title* and *subject headings* in MARC records, were the target objects of classification. The nature of classification work with MARC records seems to be easier than the work in this study due to the homogeneity of training and test sets. Also, considering the nature of objects, terms used in Dissertation Abstracts are more diverse than those in subject headings since terms from subject headings come from a controlled source, which makes the classification of MARC records a much easier task. In the Pharos project, Dolin (1998) conducted an experiment to classify 42 randomly

selected news articles from Newsgroups into the LCC Outline. However, the results were inconclusive. In Frank's work (2004), they sought to predict the LC classes of the INFOMINE records where classification data are absent, but they were unable to measure performance. From the perspective of classes in test, Pharos and Frank's work have a common characteristic in that they both implemented the 4214 categories of the LCC Outline. However, Larson's classification was focused on the LC class Z only where 5,765 clusters were generated. In this study, the three LC classes N (Fine Arts), S (Agriculture), and Q (Science) were considered at the second-level of the LCC Outline where there are all 26 subclasses together. However, it should be repeated that the selection of these three LC classes was carefully made to reflect our intention to cover the subjects representing extremes and the middle of a subject spectrum, which refers to the fact that they can cover all 21 top-level LC classes in breadth. Since the evaluation metric is the average of rankings, it seems to be a disadvantage when a larger number of classes are considered. The summary of experimental results from the studies are as follows: Larson's experiments with 283 MARC records reported the best performance to be 22.22 in ranking average, and the Pharos project reported the best average mean of 76 with 7214 MARC records. According to Frank's experiments, 42.54% (which is a percentage indicating perfect prediction with its mean ranking unknown) out of 50,000 MARC records were correctly predicted. This study reports 2.97 of the mean value of rankings with 625 dissertation abstracts. When two different perspectives are taken into consideration, direct comparisons among different experiments as well as indirect comparison by scaling up or down according to the number of classes used do not seem to be reasonable. Nevertheless, it suffices to say that the results show that the HMM-based TC system used in this study is at least comparable to other systems.

In the studies reviewed above, the nature of the hierarchical structure of classification classes is largely ignored in building systems, except in Frank's work. As specified in Chapter 1, the primary focus of this dissertation is on the design of a new framework for TC, rather than application-level concerns such as category structure, although the utilization of category structure is probably one of the future issues to be dealt with. There has been only a handful of research conducted on this issue in relation to the improvement

of performance (Koller & Sahami, 1997; McCallum *et al.*, 1998). Frank & Paynter (2004) demonstrated that classification accuracy decreases as hierarchical levels are deeper. Due to decreasing performance, they pointed out the inheritance of errors made at higher levels into lower levels and sparseness of training data available at lower levels. As the influencing factors causing the phenomenon are independent of algorithms or frameworks taken for TC, it is predictable that the performance of a HMM-based TC system, while taking hierarchical structure into consideration, may reflect the same trend reported in their studies.

Ranking of actual categories	1-5	6-10	11-15	16-20	20-25
Number of test documents	535	57	24	5	4
(percentage)	(85.6%)	(9.1%)	(3.8%)	(0.8%)	(0.6%)
Number of matched terms	110	93	81	75	73
(percentage)	(35.4%)	(29.9%)	(26.0%)	(24.1%)	(23.5%)
Averaged TF/IDF per matched term	1.42	0.65	0.50	0.34	0.34

Table 6.3: Analysis of experimental results by ranking of actual categories for the HMM system

Ranking of actual categories		1-5	6-10	11-15	16-20	21-25
Q class (Science)	Number of test documents	252	28	14	3	3
	Number of matched terms	108	93	72	73	85
	Averaged TF/IDF per matched term	1.10	0.63	0.56	0.34	0.33
	(Difference from the higher ranking next)		(-0.47)	(-0.07)	(-0.22)	(-0.01)
S class (Agriculture)	Number of test documents	124	17	6	2	1
	Number of matched terms	107	82	98	78	35
	Averaged TF/IDF per matched term	1.95	0.71	0.46	0.34	0.38
	(Difference from the higher ranking next)		(-1.24)	(-0.25)	(-0.22)	(+0.04)
N class (Fine Art)	Number of test documents	159	12	4	0	0
	Number of matched terms	117	107	87	0	0
	Averaged TF/IDF per matched term	1.52	0.59	0.35	0	0
			(-0.93)	(-0.24)		

Table 6.4: Analysis of experimental results by ranking of actual categories and LC classes

Chapter 7

CONCLUSIONS AND DISCUSSION

7.1 Summary and Conclusions

Recently TC has become a popular IR-related research field as the importance of the management and organization of digital information is growing. The TC task is recognized by IR researchers as a research task that, given pre-set categories, provides relevant categories for digital documents automatically by identifying their textual contents. In the current approach to TC, a TC model is viewed as an object that learns relevant information needed for making a decision on relevant categories. Thus, extensive effort has been made of investigating various conceptual models such as Naïve Bayesian, Classification Trees, and Support Vector Machines as learning models applicable to the TC task. This study introduces a hidden Markov model to be a learning model to tackle TC problems as well as a conceptual prototype for learning.

This dissertation explores the validity of a HMM as a TC model, especially with LC classes as a platform to classify text documents. This study primarily aims to develop a HMM-based TC system, and to investigate the validity of its ability to learn subjects (corresponding to categories) as a learning model as well as for discriminating relevant categories as a TC model. This study presented a new model prototype of HMM, algorithms, and experimental results showing a highly achieved performance for the HMM-based TC system in a LCC-based TC task. The conclusions of this study are summarized regarding the three primary research questions set earlier (Section 1.4).

The first research question is how the model represents the classification scheme employed. This question arose in the model training stage. As a classification framework, a subset of the LCC is chosen for various reasons (in Section 2.6). Previous TC research

using LCC, including this study, were primarily distinguished through the use of different models. Thus, the LCC submission to which HMMs are applied is unique to this TC research. As for the question of representing LCC by HMMs, a major concern is how to design and construct a HMM structure to represent the LC classes considered. A HMM structure is proposed as an abstract topology for a class (in Section 4.3), where a model state of the structure is defined as a source of information for a class. Our approach is to provide a way of collecting information from multiple sources. In training, model parameters including transition probabilities are determined. In this model, a transition probability connecting two states is interpreted as a relative weight of the information source referred to by a destination state. The proposed model can be characterized by its capability of merging various information extracted from diverse sources. In the current, implemented version of the HMM-based TC system, only two information sources are incorporated. However, the concept of merging multiple sources has been attempted in this study for the vision that as more sources are integrated into a HMM-based model, the model may be incapable of doing automatic classification with additional various types of document sources.

The second research question is the issue of how relevant categories for documents are determined, including the evaluation of the system's performance. This question was raised in the model testing stage. While being tested with a test data set, the classifier trained for its corresponding category (a LC class in this case) yields a score indicating the degree of similarity of the category to the tested document. Our approach uses the popular dynamic method referred to as the forward algorithm for the estimation of probability. One of the major advantages of HMM is that efficient solutions (see Chapter 2) for the challengeable problems related to the use of HMMs were already known, including the forward algorithm. Consequently, given a test document, the probably relevant categories by ranking are generated as the output of the system. That is, the system guesses the top-ranked category as the most probably relevant one. The issue raised in this question influences the system performance. Thus, the evaluation of performance is indirectly related to this issue. Performance is assessed in accuracy: how the system anticipates the relevant categories correctly. The overall accuracy is measured

as high as 0.768 with the alpha value equal to 3, and 0.856 with the alpha value being 5. Considering that the perfect accuracy indicator is 1, and other models used for comparative purposes have produced less than 0.5 even with the highest alpha value, the proposed HMM-based system's performance can be considered to outperform the others. Judging from the overall performance, this study of the use of HMMs demonstrates evidence that HMMs are promising models for TC.

The third research question focuses on the flexibility for system expansion. With the availability of a great amount of new digital information in a short period of time, the expansion and updating of TC systems is an essential system component. The system expansion in question aims to add new categories to a system, and adding new sources to existing categories. From the model prototype standpoint, with new categories being added, the same model prototype can be still used. In the case that new sources are added, the existing prototype can be easily expanded to accommodate it, according to the general model prototype proposed (see Figure 4.2). From the model efficiency standpoint (running time), since the running time in testing for the general model prototype stays in $\Theta(NT)$ multiplications, N for the number of states and T for the number of terms in a document, the system expansion stays in the same boundary as the running cost. Although no experiment for expanding the system was attempted in this study, the same procedures and methodologies used could be applied to those for system expansion.

With LCC as the classification outline for this task, the difficulty of classifying text documents containing different complexities in semantic ambiguity has been experimented. To the contrary, the result does not reveal any relationship between the complexity of semantic ambiguity and the classification difficulty, as one would one generally assume.

In conclusion, the experimental results show that our HMM model, methods, and algorithms provide favorable results for a TC task, within the limited settings of this experiment, and although much work is needed in order to produce a mature model, this study shows that a positive direction has been taken with this model.

7.2 Contributions

In this section, the contributions of this dissertation to the fields of library and information science as well as computer science are summarized. The crux of the research on text classification is to make a categorical decision where subjective prejudice is inherently included. In library science, such activities as determining the subjects related to an item and assigning proper categories among the given classes, have long been accomplished by human professionals. Recently, the automatic methods of doing similar tasks have attracted interest as demand has increased with the availability of a large volume of digital documents. Consequently, a group of computer scientists has indulged in research on automated text classification. Their interests are rooted in the study of information retrieval, which is a major research field in information science. As these three disciplines have intertwined themselves in the subject of text classification, their contributions are self-guided towards the subjects related to their particular disciplines.

The contributions of their presence in this learning model can be described in the following three branches:

Contributions in theory

- Development of a HMM-based TC model:

HMM has been recognized as a successful model in signal processing applications since the late 1970s, and it recently began to be considered for text-based and IR-related tasks. A handful of experimental level research of this sort has been published. The uniqueness of this work can be found in the relatively successful introduction of HMM to the automated TC field using a traditional library classification scheme.

- Ability of incorporating multiple resources and experiments with new text data sets for automated classification:

The characteristics of this corpus may have been an influential factor in explaining the performance inconsistencies of a same model in diverse experimental environments. In this study, a specific type of document, dissertation abstracts, was used in the training and testing of the model, but were not utilized in the

automated classification system. Also, the proposed model prototype was devised for the integration of heterogeneous data stemming from multiple sources towards a target subject.

Contributions in application

- A framework for the automatic organization of text documents:
The proposed model can be viewed as a systematical tool to analyze textual documents and organize them into a certain structural arrangement. This framework can be used to build an entire system or may be used as an integrated component of any information system.
- Extension of the use of LCC:
The invention and the use of LCC has been limited to the manual classification system mainly for organizing and accessing tangible items collected in libraries. Prior to this work, some laboratory works had been made to apply the concept of LCC towards the organization of non-traditional items. In this study, however, LCC has been systematically implemented in the model and employed as the basis for automatically determining the relevant categories of a document and organizing them accordingly. Therefore, this model opens a new possibility for the development of an integrated system that would organize and retrieve both digital information and non-digital items under one umbrella. This work can be applicable to OPAC, digital library, and information management systems.
- A method for automatic multi-faceted classification:
This model is not limited to a certain type of classification scheme and moreover is widely open to the diverse features of classifying text. Such flexibility can extend its use to multi-faceted classifications whose importance has increased in current information systems.
- Others related applications:

The HMM-based model can be used as a statistical prototype for concept representation including but not -limited to data mining, concept learning, and document representation.

In summary, this study contributes to *text classification* and *machine learning* by providing a prototype for a HMM-based learning model and to the related research areas of information and library science such as *concept representation* and *subject tools*:

7.3 Further Research

The following future directions for further research and improvements of our proposed methods are proposed:

- Further extensions can be made toward having a generalized model that is capable of dealing with various sources. To move in such a direction, the TC system needs to be tested with more diverse text sources such as newspaper articles and Web pages, and from controlled sources such as thesauri, if possible. An extension to incorporate deeper and wider levels of LCC: Use of LCC hierarchical structure; Construction of numerous models
- Fine-tuning of the HMM model prototype and internal parameters: Estimation of different information sources
- Application to common TC tasks for a direct comparison of the performance of the HMM model for TC to those of other models.
- Study of the approach to switch to other classification systems such as the Dewey Decimal Classification scheme through the use of a conversion table, possibly in the interface level without affecting any design or implementation components.
- Development of an automatic re-training process with new training data. Acquiring new training data can be another challenge. By applying a TC system

to unlabeled training data set, they can be converted to a new set of labeled training data.

- Analysis and experimental comparison of different learning TC algorithms.
- Extension of the user interface by adding communication tools with users and visualization tools for navigating all contents and presenting relevant results.

Appendix A

LIST OF THE ABBREVIATIONS USED

AI	Artificial Intelligence
DDC	Dewey Decimal Classification
EI	Engineering Information
EM	Expectation Maximization
ERIC	Educational Resources Information Center
ESS	Editorial Support System
GCT	General Classification Theory
HMM	Hidden Markov Model
IDF	Inversed Document Frequency
IS	Information Source
k-NN	k-Near Neighbor
LC	Library of Congress
LCC	Library of Congress Classification
LCSH	Library of Congress Subject Headings

LM	Language Model
LSI	Latent Semantic Indexing
MARC	MAchine Readable Cataloging
MeSH	Medical Subject Headings
ML	Machine Learning
NB	Naïve Bayesian
NLM	National Library of Medicine
NN	Neural Network
OCLC	Online Computer Library Center
OPAC	Online Public Access Catalog
POS	Part-of-Speech
PQDD	ProQuest Digital Dissertation
SFC	sub-field code
SVD	Singular Value Decomposition
SVM	Support Vector Machines
TC	Text Classification or Text Categorization
TDT	Topic Detection and Tracking
TF	Term Frequency

UDC	Universal Decimal Classification
URL	Universal Resource Locator
VSM	Vector Space Model
WWlib	Wolverhampton Web Library

Appendix B

LIST OF HARD/SOFT DISCIPLINES²⁷

Rank	Discipline	Rank	Discipline
1 (Hardest)	Mathematics	33	Ind and Tech Educ
2	Chemistry	34	Voc Agric Educ
3	Electrical Engineering	35	Business Communication
4	Chemical Engineering	36	General Business
5	Physics	37	Economics
6	Mechanical Engineering	38	Home Economics
7	Civil Engineering	39	Management
8	Statistics	40	History
9	Spanish	41	English
10	German	42	Voc Ind Educ
11	Microbiology	43	Music
12	Petroleum Engineering	44	Reading
13	Computer Science	45	Marketing
14	Latin	46	Architecture
15	General Engineering	47	Speech
16	Biology	48	Physical Education
17	Accounting	49	Law Enforcement
18	Geology	50	Health Education
19	French	51	Journalism
20	Horticulture	52	Art and Architecture
21	Animal Husbandry	53	Special Education
22	Agricultural Engineering	54	Adult Education
23	Dairy Husbandry	55	Applied Arts
24	Nursing	56	Dance
25	Geography	57	Education
26	Agronomy	58	Psychology
27	Medical Record Science	59	Sociology
28	Library Science	60	Political Science
29	Aerospace Studies	61	Philosophy
30	Agriculture	62	Recreation
31	Finance	63 (Softest)	Fine Arts
32	Office Administration		

²⁷ This list is cited from (McGraph, 1978, p. 22).

Appendix C

SELECTED PERL SCRIPTS

TITLE: DF Count for Distinct Words

```
#!/usr/bin/perl -w

# Kwan Yi writes this script to count the document frequencies (DF) of distinct word appearing in all the
# corpus.

$total_terms = 0;
%frequencies = ();

while ($filename = <TF*.txt>) {
    open (INPUT, $filename) || die "Can't open $filename: $!\n";
    print "Processing $filename \n";
    while (defined($read = <INPUT>)) {
        chomp($read);
        if ($read =~ /\s+\d+\s+\S+/) {
            $cnt = $read;
            $cnt =~ s/\s+(\d+)\s+\S+/$1/;
            $total_terms += $cnt;

            $read =~ s/\s+\d+\s+(\S+)/$1/;
            $frequencies{$read}++;
        }
    }
    close INPUT;
}

# declare a file to record Term Frequencies
$outfilename = "DF_full_V51_tr.txt";
open (DFOUT, "> $outfilename") || die "Can't open $outfilename: $!\n";

$max_val = 0;
$max_term = "";

foreach $term (keys %frequencies) {
    #print DFOUT "$term $frequencies{$term}\n";

    select("DFOUT");
    $num = $frequencies{$term};
    $listofcha = $term;
    $~ = 'OUTFORMAT';
```


TITLE: TF Count for Distinct Words

```
#!/usr/bin/perl -w
```

Kwan Yi write this script to count the term frequencies (TF) of distinct word appearing in a file.

```
while ($infilename = <STEM*.txt>) {
    open (INPUT, $infilename) || die "Can't open $infilename: $!\n";
    $logfile = $outfilename = $infilename;

    # declare a file to record Term Frequencies
    $outfilename =~ s/STEM(.*)/TF$1/;
    open (TFOUT, "> $outfilename") || die "Can't open $outfilename: $!\n";

    # declare a file to record Length of a document related to a subclass
    $logfile =~ s/STEM(.*)/LEN$1/;
    open (LENOUT, "> $logfile") || die "Can't open $logfile: $!\n";

    # set the filehandle STDOUT for output
    select("STDOUT");
    print "Processing $infilename \n";

    while (defined($read_line = <INPUT>)) {
        @sh_list = split(/\s+/, $read_line);
        foreach $sh (@sh_list) {
            $frequencies{$sh}++;
        }
    }

    foreach $sh (keys %frequencies) {
        select("TFOUT");
        $num = $frequencies{$sh};
        $listofcha = $sh;
        $~ = 'OUTPUTF';
        write;

        # print OUTPUT "$frequencies{$sh} $sh\n";
        $totalWords += $frequencies{$sh};
        $distinctWords++;
    }

    print LENOUT "[TOTAL LEN] $totalWords \n";
    print LENOUT "[DISTINCT LEN] $distinctWords \n";

    $totalWords = 0;
    $distinctWords = 0;
    %frequencies = ();

    close LENOUT;
    close TFOUT;
    close INPUT;
}

format OUTPUTF =
```


$(a) \gg \gg \gg \gg (a) \ll \ll \ll \ll \ll \ll \ll \ll \ll \ll \ll \ll \ll$

\$num, \$listofcha

TITLE: HMM Probabilities

```
#!/usr/bin/perl

# AUTHOR: Yi, Kwan
# PURPOSE:
# Implement initial and transition probabilities of HMM, not based on the EM method,
# but based on the method of viewing transition probability as a weight factor to a destination state
# NOTE:
# 1. The subclass NE is not considered, due to the lack of training data
#
# CHANGE:
# 1. Change all the filenames for input

# change the following variables
$training_data_version = "V51";
$exp_version = "2S";

# define the set of class this experiment is considering
@class_list = qw(N NA NB NC ND NK NX Q QA QB QC QD QE QH QK QL QM QP QR S SB SD SF
SH SK);
@IS_list = qw(ABSTRACT OCLC);

# declare constant global variables
%full_DF = ();
%oclc_DF = ();
%oclc_Info_Quantity = ();
%oclc_Term_Count = ();
%oclc_Info_Quantity_N = ();
%full_Info_Quantity = ();
%full_Term_Count = ();
%full_Info_Quantity_N = ();

# -- read all_DF
$sdf_abstract_file = "DF_full_" . $training_data_version . ".tr.txt";
open (DF_ABSTRACT, $sdf_abstract_file) || die "Can't open $sdf_abstract_file: $!\n";
while (defined($read_line = <DF_ABSTRACT>)) {
    chomp($read_line);
    if ($read_line =~ /\s+\d+\s\S+/) {
        $df_term = $df_val = $read_line;
        $df_term =~ s/\s+\d+\s\S+/$1/;
        $df_val =~ s/\s+(\d+)\s\S+/$1/;
        $full_DF{$df_term} = $df_val;
    } else {
        if ((length $read_line) > 0) {
            die "Finish here: $!\n";
        } else {
            print "$read_line <- If this is not empty, stop the execution \n";
        }
    }
}
}
close DF_ABSTRACT;
```

```

$df_oclc_file = "OCLC_TRAIN_DF.txt";
open (DF_OCLC, $df_oclc_file) || die "Can't open $df_oclc_file: $!\n";
while (defined($read_line = <DF_OCLC>)) {
    chomp($read_line);
    if ($read_line =~ /\S+\s\d+/) {
        $df_term = $df_val = $read_line;
        $df_term =~ s/(\S+)\s\d+/$1/;
        $df_val =~ s/\S+\s(\d+)/$1/;
        $oclc_DF{$df_term} = $df_val;
    } else {
        if ((length $read_line) > 0) {
            die "Finish here: $!\n";
        } else {
            print "$read_line <- If this is not empty, stop the execution \n";
        }
    }
}
close DF_OCLC;

# -- read all _TF and store them into internal structure
foreach $single_IS (@IS_list) {
    foreach $single_class (@class_list) {

        if ($single_IS eq "ABSTRACT") {
            $tf_file = "TF_full-Subclass" . $single_class . "_" . $training_data_version . "_tr.txt";
        } else {
            $tf_file = $single_class . "_OCLC_TF.txt";
        }

        open (TF_F, $tf_file) || die "Can't open $tf_file: $!\n";
        print "read and process with $tf_file ..\n";
        while (defined($read_tf = <TF_F>)) {
            chomp($read_tf);
            if ($read_tf =~ /\s+\d+\s\S+/) {
                $tf_term = $tf_val = $read_tf;
                $tf_term =~ s/\s+\d+\s\S+/$1/;
                $tf_val =~ s/\s+(\d+)\s\S+/$1/;

                if ($single_IS eq "ABSTRACT") {
                    $full_Info_Quantity{$single_class} += $tf_val / $full_DF{$tf_term};
                    if (! $full_DF{$tf_term}) {
                        die "Wrong: [tf_term] $tf_term [df] $full_DF{$tf_term} \n";
                    }
                    $full_Term_Count{$single_class} += 1;
                } else {
                    $oclc_Info_Quantity{$single_class} += $tf_val / $oclc_DF{$tf_term};
                    if (! $oclc_DF{$tf_term}) {
                        die "Wrong: [tf_term] $tf_term [df] $oclc_DF{$tf_term} \n";
                    }
                    $oclc_Term_Count{$single_class} += 1;
                }
            } else {
                print "TF: $read_tf: If it is not empty, stop the execution: $!\n";
            }
        }
    }
}

```

```

    close TF_F;
}
}

# -- Write results
$result_file = "TRANSITION_PROB_" . $exp_version . "_" . $training_data_version . ".txt";
open (TP_RESULT, "> $result_file") || die "Can't open $result_file: $!\n";

$sum_full_Info_Quantity = $sum_full_Term_Count = $sum_oclc_Info_Quantity =
$sum_oclc_Term_Count = 0;

foreach $single_IS (@IS_list) {
    print TP_RESULT "[IS]$single_IS";
    foreach $single_class (@class_list) {
        if ($single_IS eq "ABSTRACT") {
            $stemp = $full_Info_Quantity{$single_class} / $full_Term_Count{$single_class};
            $full_Info_Quantity_N{$single_class} = $stemp;
            $sum_full_Info_Quantity += $full_Info_Quantity{$single_class};
            $sum_full_Term_Count += $full_Term_Count{$single_class};
        } else {
            $stemp = $oclc_Info_Quantity{$single_class} / $oclc_Term_Count{$single_class};
            $oclc_Info_Quantity_N{$single_class} = $stemp;
            $sum_oclc_Info_Quantity += $oclc_Info_Quantity{$single_class};
            $sum_oclc_Term_Count += $oclc_Term_Count{$single_class};
        }
        print TP_RESULT "[CLASS]{$single_class}-$stemp ";
    }
}

if ($single_IS eq "ABSTRACT") {
    $stemp = $sum_full_Info_Quantity / $sum_full_Term_Count;
} else {
    $stemp = $sum_oclc_Info_Quantity / $sum_oclc_Term_Count;
}
print TP_RESULT "[TOTAL]$stemp\n";
}

# The normalized Information Quantity for different classes
print TP_RESULT "[NORMALIZED_OCLC]";
foreach $single_class (@class_list) {
    $stemp = $oclc_Info_Quantity_N{$single_class} / ($oclc_Info_Quantity_N{$single_class} +
$full_Info_Quantity_N{$single_class});
    print TP_RESULT "{$single_class}-$stemp ";
}

# The normalized Information Quantity for different information sources
$sum_full_Info_Quantity_N = $sum_full_Info_Quantity / $sum_full_Term_Count;
$sum_oclc_Info_Quantity_N = $sum_oclc_Info_Quantity / $sum_oclc_Term_Count;
$stemp = $sum_oclc_Info_Quantity_N / ($sum_oclc_Info_Quantity_N + $sum_full_Info_Quantity_N);
print TP_RESULT "[TOTAL]$stemp\n";

close TP_RESULT;

```

TITLE: Forward Algorithm

```
#!/usr/bin/perl

use Math::BigFloat;

# Kwan Yi writes this script to implement the forward algorithm to calculate the probability of a
# observation sequence.
#
# In this version, TF/IDF-based transition probability is considered.
# For the estimation of emission probability, the n-estimate probability is adopted
# [Machine Learning. Tom M. Mitchell. pp. 179] &&
# [HMMs for TC in multipage documents. pp. 202]
#

# -----
# MAKE SURE THAT THE VALUES OF THE VARIABLES BELOW ARE CORRECT
# -----
$training_data_version = "V51";
$exp_version = "2S";

# -----
# declare global variables
# -----
# define the set of class this experiment is considering
@class_list = qw(N NA NB NC ND NK NX Q QA QB QC QD QE QH QK QL QM QP QR S SB SD SF
SH SK);

# declare constant global variables
%EMISSION_FULL_DISTINCT_LEN = ();
%EMISSION_FULL_TOTAL_LEN = ();
%EMISSION_OCLC_DISTINCT_LEN = ();
%EMISSION_OCLC_TOTAL_LEN = ();
# ++++++
@TERM_MATCH = ();
# ++++++

# -----
# READ TRANSITION PROBABILITIES
# -----
$transition_file = "TRANSITION_PROB_" . $exp_version . "_" . $training_data_version . ".txt";
open (TP_F, $transition_file) || die "Can't open $transition_file: $!\n";
@transition_lines = <TP_F>;
close TP_F;

# -----
# read all _TF and _LEN
# -----
foreach $single_class (@class_list) {

    print "read and process with _TF and _LEN of $single_class class ..\n";

    # -----
    # define symbolic reference to hash names
```

```

# -----
$Emission_FULL_ref = "EP_FULL_" . $single_class; # This is a hash name for loading TF info of
FULL
$Emission_oclc_ref = "EP_OCLC_" . $single_class; # This is a hash name for loading TF info of
OCLC

# -----
# "title" IS
# -----
# --read TF
%{$Emission_FULL_ref} = ();
$FULL_tf_file = "TF_full-Subclass" . $single_class . "_" . $straining_data_version . "_tr.txt";
open (AB_TF_F, $FULL_tf_file) || die "Can't open $FULL_tf_file: $!\n";
while (defined($read_tf = <AB_TF_F>)) {
    chomp($read_tf);
    if ($read_tf =~ /^s+\d+s\S+/) {
        $tf_term = $tf_val = $read_tf;
        $tf_term =~ s/^s+\d+s\S+/$1/;
        $tf_val =~ s/^s+(\d+)s\S+/$1/;
        $Emission_FULL_ref->{$tf_term} = $tf_val;
    } else {
        print "TF: $read_tf: If it is not empty, stop the execution: $!\n";
    }
}
close AB_TF_F;

# --read LEN
$FULL_len_file = "LEN_full-Subclass" . $single_class . "_" . $straining_data_version . "_tr.txt";
open (AB_LEN_F, $FULL_len_file) || die "Can't open $abst_len_file: $!\n";
# read info about the total number of terms in documents of the current class
if (defined($read_line = <AB_LEN_F>)) {
    chomp($read_line);
    $read_line =~ s/^[TOTAL LEN] (\d+)/$1/;
    $EMISSION_FULL_TOTAL_LEN{$single_class} = $read_line;
} else {
    die "Wrong Format [TOTAL LEN] in $FULL_len_file: $!\n";
}
# read info about the number of distinct terms in documents of the current class
if (defined($read_line = <AB_LEN_F>)) {
    chomp($read_line);
    $read_line =~ s/^[DISTINCT LEN] (\d+)/$1/;
    $EMISSION_FULL_DISTINCT_LEN{$single_class} = $read_line;
} else {
    die "Wrong Format [DISTINCT LEN] in $FULL_len_file: $!\n";
}
close AB_LEN_F;

# -----
# "oclc" IS
# -----
# --read TF
%{$Emission_oclc_ref} = ();
$oclc_tf_file = $single_class . "_OCLC_TF.txt";
open (OC_TF_F, $oclc_tf_file) || die "Can't open $oclc_tf_file: $!\n";
while (defined($read_tf = <OC_TF_F>)) {

```

```

chomp($read_tf);
if ($read_tf =~ /\s+\d+\s\S+/) {
    $tf_term = $tf_val = $read_tf;
    $tf_term =~ s/\s+\d+\s\S+/$1/;
    $tf_val =~ s/\s+(\d+)\s\S+/$1/;
    $Emission_oclc_ref->{$tf_term} = $tf_val;
} else {
    print "TF: $read_tf: If it is not empty, stop the execution: $!\n";
}
}
close OC_LEN_F;

# --read LEN
$oclc_len_file = $single_class . "_OCLC_LEN.txt";
open (OC_LEN_F, $oclc_len_file) || die "Can't open $oclc_len_file: $!\n";
# read info about the total number of terms in documents of the current class
if (defined($read_line = <OC_LEN_F>)) {
    chomp($read_line);
    $read_line =~ s/[TOTAL LEN] (\d+)/$1/;
    $EMISSION_OCLC_TOTAL_LEN{$single_class} = $read_line;
} else {
    die "Wrong Format [TOTAL LEN] in $oclc_len_file: $!\n";
}
# read info about the number of distinct terms in documents of the current class
if (defined($read_line = <OC_LEN_F>)) {
    chomp($read_line);
    $read_line =~ s/[DISTINCT LEN] (\d+)/$1/;
    $EMISSION_OCLC_DISTINCT_LEN{$single_class} = $read_line;
} else {
    die "Wrong Format [DISTINCT LEN] in $oclc_len_file: $!\n";
}
close OC_LEN_F;
}

# -----
# define filename for results
# -----
$log_file = "LOGbigfloat_" . $exp_version . "_" . $training_data_version . ".txt";
open (LOGDATA, "> $log_file") || die "Can't open $log_file: $!\n";
$experiment_file = "RESULTbigfloat_" . $exp_version . "_" . $training_data_version . ".txt";
open (EXPERIMENTDATA, "> $experiment_file") || die "Can't open $experiment_file: $!\n";
$test_file = "TEST_" . $training_data_version . ".txt";
open (TESTDATA, $test_file) || die "Can't open $test_file: $!\n";
#+++++
+++++
$matching_report = "RESULTbigfloat_matching-report_" . $exp_version . "_" . $training_data_version . ".txt";
open (MR_F, "> $matching_report") || die "Can't open $matching_report: $!\n";
#+++++
+++++

# -----
# implement the Forward algorithm
# -----

```

```

$testdata_line = 1;
while (defined($read_test = <TESTDATA>)) {
    print "Processing the $testdata_line line of the test data ... \n";
    print LOGDATA "$read_test";

    chomp($read_test);
    $target_class = $read_test;
    $target_class =~ s/^\[.{1,2}\]\s+(.*)/$1/;
    $read_test =~ s/^\[.{1,2}\]\s+(.*)/$1/;
    @test_data = split(/\s+/, $read_test);

    $time = 1;
    #+++++
    $TERM_COUNT = @test_data;
    foreach $single_class (@class_list) {
        $rec = {};
        $rec->{$class} = 0;
        push @TERM_MATCH, $rec;
    }
    #+++++

    foreach $test_term (@test_data) {
        foreach $cnt_class (@class_list) {
            # state 0 indicates "FULL" information source
            # state 1 indicates "OCLC" information source
            if ($time == 1) {
                # -- assign initial transition probabilities
                for $state ( 0 .. 1 ) {
                    $BF_initial = Math::BigFloat->new("0");
                    $BF_emission = Math::BigFloat->new("0");

                    $BF_initial->fadd(sub_cal_initial_transition_prob($state));
                    $BF_emission->fadd(sub_cal_emission_prob($cnt_class, $state, $test_term));

                    $alpha{$cnt_class}[$time][$state] = $BF_initial * $BF_emission;
                }
            } else {
                # -- calculate a forward variable of the current state, based on a forward variable of the previous
state
                $sum_of_pre_forward = 0;
                for $current_state ( 0 .. 1 ) {
                    for $previous_state ( 0 .. 1 ) {
                        $BF_transition = Math::BigFloat->new("0");
                        $BF_alpha = Math::BigFloat->new("0");

                        $BF_alpha->fadd($alpha{$cnt_class}[$time - 1][$previous_state]);
                        $BF_transition->fadd(sub_cal_transition_prob($cnt_class, $current_state));

                        $sum_of_pre_forward += $BF_alpha * $BF_transition;
                    }
                    $BF_emission = Math::BigFloat->new("0");
                    $BF_emission->fadd(sub_cal_emission_prob($cnt_class, $current_state, $test_term));

                    $alpha{$cnt_class}[$time][$current_state] = $sum_of_pre_forward * $BF_emission;
                }
            }
        }
    }
}

```



```

    }
  }
  $time++;
}
#close LOGDATA;

# calculate the probability of a given sequence observation
$time--;
%result_info = ();
foreach $cnt_class (@class_list) {
  for $state ( 0 .. 1 ) {
    $result_info{$cnt_class} += $alpha{$cnt_class}[$time][$state];
    print LOGDATA "[CLASS] $cnt_class [TIME] $time [STATE] $state [VALUE]
$alpha{$cnt_class}[$time][$state] [TOTAL] $result_info{$cnt_class} \n";
  }
}

# print out the result according to different class
print EXPERIMENTDATA "[TARGET_CLASS] $target_class [RANKING]";
$i = 1;
foreach $class (sort sub_numerically keys %result_info) {
  if ($target_class eq $class) { $target_rank = $i; }
  print EXPERIMENTDATA "({i})${class}-$result_info{$class}";
  $i++;
}
print EXPERIMENTDATA "[RANK_OF_TARGET_CLASS]$target_rank\n";

#+++++
# Write analytic result for the term matching
#+++++
foreach $single_class (@class_list) {
  print MR_F "\n[${testdata_line}-${single_class}] $TERM_COUNT ";
  for $IS ( 0 .. 1 ) {
    if ($IS == 0) {
      print MR_F "[FULL] ";
    } else {
      print MR_F "[OCLC] ";
    }
  }

  $tmp_matched = $TERM_MATCH[$IS][$single_class];
  $tmp_ratio = int (($tmp_matched / $TERM_COUNT) * 100);
  print MR_F "$tmp_matched [$tmp_ratio %]";
}
}
@TERM_MATCH = ();
#+++++
$testdata_line++;
}

close LOGDATA;
close TESTDATA;
close EXPERIMENTDATA;
#+++++
close MR_F;
#+++++

```

```

# -----
# list of subroutines
# -----
sub sub_numerically { $result_info{$b} <=> $result_info{$a} }

sub sub_cal_initial_transition_prob {
    my ($to_state) = shift;
    my ($tran_prob);

    $tran_prob = $transition_lines[2];
    $tran_prob =~ s/[TOTAL](\S+)/$1/;

    if ($to_state == 0) {
        return (1 - $tran_prob);
    } elsif ($to_state == 1) {
        return $tran_prob;
    } else {
        return 0;
        die "Wrong state in sub_cal_initial_transition_prob: $state : $!\n";
    }
}

sub sub_cal_transition_prob {
    my ($class, $to_state) = @_;
    my ($tran_prob);

    $tran_prob = $transition_lines[2];
    $tran_prob =~ s/${class}-(\S+)/$1/;

    if ($to_state == 0) {
        return (1 - $tran_prob);
    } elsif ($to_state == 1) {
        return $tran_prob;
    } else {
        return 0;
        die "Wrong state in sub_cal_transition_prob: $state : $!\n";
    }
}

sub sub_cal_emission_prob {
    my ($class, $state, $term) = @_;
    my ($array_type, $IS_type, $total_len, $distinct_len);

    if ($state eq 0) {
        $IS_type = "FULL";
        $total_len = $EMISSION_FULL_TOTAL_LEN{$class};
        $distinct_len = $EMISSION_FULL_DISTINCT_LEN{$class};
    } elsif ($state eq 1) {
        $IS_type = "OCLC";
        $total_len = $EMISSION_OCLC_TOTAL_LEN{$class};
        $distinct_len = $EMISSION_OCLC_DISTINCT_LEN{$class};
    } else {
        die "State should be 0 or 1: $!\n";
    }
}

```

```

$array_type = "EP_" . $IS_type . "_" . $class;

# ++++++
if ($array_type->{$term}) {
    $TERM_MATCH[$state][$class]++;
}
# ++++++
return (1 + $array_type->{$term}) / ($total_len + $distinct_len);
}

```

BIBLIOGRAPHY

- Allan, James, Jaime Carbonell, George Doddington, Jonathan Yamron, & Yiming Yang. "Topic Detection and Tracking Pilot Study Final Report." Proceedings of the 1998 Defense Advance Research Projects Agency (DARPA) Broadcast News Transcription and Understanding Workshop, Lansdowne, Virginia, 8-11 February 1998. 194-218.
- Amini, Massih-Reza & Patrick Gallinari. "The Use of Unlabeled Data to Improve Supervised Learning for Text Summarization." Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 11-15 August 2002. 105-112.
- Andresen, D. *et al.* "The WWW Prototype of the Alexandria Digital Library." IEEE Computer 29.5 (May 1996): 54-60.
- Androutsopoulos, Ion *et al.* "Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach." Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases' workshop on Machine Learning and Textual Information Access, 2000. 1-13.
- Baeza-Yates, Ricardo, & Berthier Ribeiro-Neto. Modern Information Retrieval. New York, NY: ACM Press, 1999.
- Baldi, P. & S. Brunak. Bioinformatics, the Machine Learning Approach. Cambridge, MA: MIT Press, 1998.
- Bareiss, E. R., B. Porter, & C. C. Wier. "PROTOS: An Exemplar-Based Learning Apprentice." Proceedings of the Fourth International Workshop on Machine Learning, Irvine, CA, 1987. 12-23.
- Bates, Marcia J. "Indexing and Access for Digital Libraries and the Internet: Human, Database, and Domain Factors." Journal of the American Society for Information Science 49.13 (1998): 1185-1205.
- Baum, L. E. & J. A. Egon. "An Inequality with Applications to Statistical Estimation for Probabilistic Functions for a Markov Process and to a Model for Ecology." Bull. Amer. Meteorol. Soc. 73 (1967): 360-363.
- Baum, L. E. & G. R. Sell. "Growth Functions for Transformations on Manifolds." Pac. J. Math. 27.2 (1968): 211-227.

- Bead, Charles C. "The Library of Congress Classification: Development, Characteristics, and Structure." The Use of the Library of Congress Classification: Proceedings of the Institute on the Use of the Library of Congress Classification. Ed. Richard H. Schimmelpfeng & C. Donald Cook, Chicago: American Library Association, 1968.
- Belew, R. K. (1989). "Adaptive information retrieval." Proceedings of the 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Cambridge, MA, 25-28 June 1989. 11-20.
- Bengio, Yoshua. "Markovian Models for Sequential Data." Neural Computing Surveys 12 (1999): 129-162.
- Biglan, A. "Characteristics of subject matter in different academic areas." J. Appl. Psychol. 57.1 (1973): 195-203.
- Bikel, D. M., S. Miller, R. L. Schwartz, & R. M. Weischedel. "Nymble: a High-Performance Learning Name-Finder." Proceedings of the 5th Conference on Applied Natural Language Processing, Washington, D.C., 31 March – 3 April 1997. 194-201.
- Bikel, D. M., R. L. Schwartz, & R. M. Weischedel. "An Algorithm that Learns What's in a Name." Machine Learning Journal 34 (1999): 211-231.
- Blei, D. M. & Pedro J. Moreno. "Topic Segmentation with an Aspect Hidden Markov Model." Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, 9-13 September 2001. 343-348.
- Bloomberg, Marty & Hans Weber. An Introduction to Classification and Number Building in Dewey. Littleton, Colorado: Libraries Unlimited, 1976.
- Blosseville, M. J., G. Hebrail, M. G. Monteil, & N. Penot. "Automatic document classification: Natural language processing, statistical analysis, and expert system techniques used together." Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, 21-24 June 1992. 51-57.
- Blum, A. & T. Mitchell. "Combining labeled and unlabeled data with co-training." Proceedings of the Workshop on Computational Learning Theory (COLT-98), 1998. 92-100.
- Breiman, L. & P. Spector. "Submodel selection and evaluation in regression: The X-random case." International Statistical Review 60 (1992): 291-319.
- Buchanan, Brian. Theory of Library Classification. London: Clive Bingley, 1979.

- Buntine, Wray. "A theory of learning classification rules." PhD Dissertation. University of Technology, Australia, February 1990.
- Buntine, Wray. "Introduction to IND and recursive partitioning." Technical Report. RIACS/NASA Ames Research Center, September 1991.
- Campbell, K. K. "A NET.CONSPIRACY SO IMMENSE : Chatting with Martha Siegel of the Internet's Infamous Canter & Siegel." Oct. 1, 1994.
<[http://www.eff.org/Legal/Cases/Canter Siegel/c-and-s_summary.article](http://www.eff.org/Legal/Cases/Canter_Siegel/c-and-s_summary.article)> visited on 23 October 2003.
- Carreras, Xavier & L. Márquez. "Boosting trees for anti-spam email filtering." Proceedings of the 4th International Conference on Recent Advances in Natural Language Processing (RANLP-2001), Tzigov Chark, Bulgaria, 2001.
- Chan, Lois Mai. "Inter-indexer Consistency in Subject Cataloging." Information Technology and Libraries 8 (1989): 349-358.
- Chan, Lois Mai. Cataloging and Classification: an Introduction. New York, NY: McGraw-Hill, 1994.
- Chan, Lois Mai. "Dewey Decimal Classification Edition 21 and International Perspectives." Dewey Decimal Classification: Edition 21 and International Perspectives. Ed. Lois Mai Chan & Joan S. Mitchell, Albany, New York: Forest Press, 1996.
- Chan, Lois Mai. A Guide to the Library of Congress Classification. 5th ed. Englewood, Colorado: Libraries Unlimited, 1999.
- Chakrabarti, S., B. E. Dom, & P. Indyk. "Enhanced hypertext categorization using hyperlinks." Proceedings of the ACM SIGMOD1998, Seattle, WA, 1998. 307-318.
- Charniak, Eugene. Statistical Language Learning. Cambridge, Massachusetts: The MIT Press, 1993.
- Chen, Hsinchun. "Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms." Journal of the American Society for Information Science 46.3 (1995): 194-216.
- Chen, Hsinchun, T. D. Ng, J. Martinez, & B. R. Schatz. "A Concept Space Approach to Addressing the Vocabulary Problem in Scientific Information Retrieval: an Experiment on the Worm Community System." Journal of the American Society for Information Science 48.1 (1997): 17-31.

- Chen, Hsinchun. "Semantic Research for Digital Libraries." D-Lib Magazine. Oct. 1999. <<http://www.dlib.org/>> visited on 13 Jan. 2003.
- Chickering, D., D. Heckerman, & C. Meek. "A Bayesian approach for learning Bayesian networks with local structure." Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence, Providence, RI, August 1997. 80-89.
- Chuang, W.T. & J. Yang. "Extracting Sentence Segments for Text Summarization: A Machine Learning Approach." Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, 24-28 July 2000. 152-159.
- Cohen, W. W. "Fast effective rule induction." Proceedings of the 12th International Conference on Machine Learning (ICML-95), Lake Tahoe, CA, 1995. 115-123.
- Cohen, William W. "Learning rules that classify e-mail." Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access, Stanford, CA, 1996. 18-25.
- Comaromi, J. P., *et al.* Dewey Decimal Classification and Relative Index : Devised by Melvil Dewey. 20th ed. Albany, New York: Forest Press, 1989.
- Conroy, John M. & Dianne P. O'Leary. "Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition." Technical report CS-TR-4221. University of Maryland at college park, Department of Computer Science, February 2001a.
- Conroy, John M. & Dianne P. O'Leary. "Text Summarization via Hidden Markov Models." Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, 9-13 September 2001b. 406-407.
- Cooper, G., *et al.* "An evaluation of machine-learning methods for predicting pneumonia mortality." Artificial Intelligence in Medicine 9.2 (1997): 107-138.
- Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest. Introduction to Algorithms. Cambridge, Massachusetts: MIT Press, 2001.
- Cranor, L. F. & B. A. LaMacchia. "Spam!" Communication of the ACM 41.8 (1998): 74-83.
- Craven, M. & S. Slattery. "Relational learning with statistical predicate invention: Better models for hypertext." Machine Learning 43.1-2 (2001): 97-119.

- Crawford, Stuart L., Robert M. Fung, Lee A. Appelbaum, & Richard M. Tong. "Classification Trees for Information Retrieval." Proceedings of the Eight International Workshop on Machine Learning, Evanston, IL, 27-29 June 1991. 245-249.
- Cui, Hong, P. B. Heidorn, & Hong Zhang. "An approach to automatic classification of text for information retrieval." Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries, Portland, Oregon, 96-97. 2002.
- Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, & R. Harshman. "Indexing by Latent Semantic Indexing." Journal of the American Society for Information Science 41.6 (1990): 391-407.
- DeJong, G. F., & R. J. Mooney. "Explanation-Based Learning: An Alternative View." Machine Learning 1 (1986): 145-176.
- Dempster, A. P., *et al.* "Maximum Likelihood from Incomplete Data via the EM Algorithm." J. Roy. Stat. Soc. 39.1 (1977): 1-38.
- DeRose, S. J. "Grammatical Category Disambiguation by Statistical Optimization." Computational Linguistics 14 (1988): 31-39.
- Dewey, Melvil. Dewey Decimal Classification and Relative Index. 20th ed. Vol. 1. Albany, New York: Forest Press, 1989.
- Dittman, Helena & Jane Hardy. Learn Library of Congress Classification. Lanham, Maryland: Scarecrow Press, 2000.
- Dolin, Ron A. "Pharos: Scalable Distributed Architecture for Locating Heterogeneous Information Sources." PhD Dissertation. University of California, Santa Barbara, June 1998.
- Dolin, Ron A., D. Agrawal, A. El Abbadi, & L. Dillon. "Pharos: A Scalable Distributed Architecture for Locating Heterogeneous Information Sources." Proceedings of the 6th International Conference on Information and Knowledge Management, Las Vegas, Nevada, November 1997.
- Dolin, Ron A., D. Agrawal, A. El Abbadi, & J. Peralman. "Using Automated Classification for Summarizing and Selecting Heterogeneous Information Sources." D-Lib Magazine. Jan. 1998. <<http://www.dlib.org/dlib/january98/dolin/01dolin.html>> visited on 13 Jan. 2003.
- Dolin, Ron A., D. Agrawal, & A. El Abbadi. "Scalable collection summarization and selection." Proceedings of the Fourth ACM International Conference on Digital Libraries. Aug. 1999, Berkeley.

- Drabenstott, Karen Markey, & Diane Vizine-Goetz. Using Subject Headings for Online Retrieval. San Diego, CA: Academic Press, 1994.
- Drucker, Harris, Vladimir Vapnik, & Dongui Wu. "Support vector machines for spam categorization." IEEE Transactions on Neural Networks 10.5 (1999): 1048-1055.
- Duda, R., P. Hart, & D. Stork. Pattern Classification. New York, NY: Wiley, 2000.
- Dumais, Susan, John Platt, David Heckerman, & Mehran Sahami. "Inductive learning algorithms and representations for text categorization." Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM-98), Bethesda, Maryland, 1998. 148-155.
- Dumais, Susan & Hao Chen. "Hierarchical classification of Web content." Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, 24-28 July 2000. 256-263.
- Elke, M. & Peter Schäuble. "Document and Passage Retrieval Based on Hidden Markov Models." Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 3-6 July 1994. 318-327.
- Finni, John J. & Peter J. Pauson. "The Dewey Decimal Classification Enters the Computer Ages: Developing the DDC Database™ and Editorial Support System." International Cataloguing 16.4 (1987): 46-48.
- Forney, G. D. "The Viterbi Algorithm." Proceedings of the IEEE. 61 (March 1973): 268-278.
- Frank, Eibe & Gordon W. Paynter. "Predicting Library of Congress Classifications From Library of Congress Subject Headings." Journal of the American Society for Information Science and Technologies 55.3 (2004): 214-227.
- Frasconi, Paolo, Giovanni Soda, & Alessandro Vullo. "Hidden Markov Models for Text Categorization in Multi-Page Documents." Journal of Intelligent Information Systems 18.2-3 (2002): 195-217.
- Freitag, Dayne & Andrew McCallum. "Information Extraction with HMMs and Shrinkage." Workshop Technical Report (WS-99-11) of the 16th National Conference on Artificial Intelligence (AAAI-99), Orlando, FL, 18-22 July 1999. 31-36.
- Freitag, Dayne & Andrew McCallum. "Information Extraction with HMM Structures Learned by Stochastic Optimization." Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, 30 July – 3 August 2000. 584-589.

- Freund, Y., R. Schapire, Y. Singer, & M. Warmuth.. "Using and combining predictors that specialize." Proceedings of the 29th Annual ACM Symposium on Theory of Computing, 1997. 334-343.
- Fürnkranz, Johannes. "Exploiting structural information for text classification on the WWW." Proceedings of the 3rd Symposium on Intelligent Data Analysis (IDA 99), Amsterdam, NL, 1999. 487-498.
- Ghani, R., R. Jones, D. Mladenic, K. Nigam, & Slattery, S. "Data mining on symbolic knowledge extracted from the Web." Workshop on Text Mining at the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000.
- Glover, Eric J., Kostas Tsioutsoulis, Steve Lawrence, David M. Pennock, & Gary W. Flake. "Using Web structure for classifying and describing Web pages." Proceedings of the 11th WWW, Honolulu, Hawaii, 7-11 May 2002. 562-569.
- Gordon, M. "Probabilistic and genetic algorithms for document retrieval." Communications of the ACM 31 (1988): 1208-1218.
- Goutte, C. "Note on free lunches and cross-validation." Neural Computation 9 (1997):1211-1215.
- Guenther, Rebecca S. "The Development and Implementation of the USMARC Format for Classification Data." Information Technology and Libraries 11.2 (1992): 120-131.
- Guenther, R. S. "Automating the Library of Congress Classification Scheme: implementation of the USMARC Format for Classification Data." Cataloging & Classification Quarterly 21.3/4 (1996): 177-203.
- Gwynne, S. & J. Dickerson. "Lost In The E-Mail." Time Magazine 21 April 1997.
- Hahn, Udo and Inderjeet Mani. "The challenges of automatic summarization." IEEE Computer 33.11 (2000): 29-36.
- Harter, Stephen P. Online Information Retrieval. Orlando, FL: Academic Press, 1986.
- Hayes, P. J. & S. P. Weinstein. "CONSTRUE/TIS: A System for Content-Based Indexing of a Database of News Stories." Proceedings of the 2nd Conference on Innovative Applications of Artificial Intelligence (IAAI-90), Washington, D.C., 1-3 May 1990. 1-5.
- Hersh, William, Chris Buckley, T. J. Leone, & David Hickam. "OHSUMED: an interactive retrieval evaluation and new large test collection for research." Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 03-06 July 1994. 192-201.

- Hidalgo, J. G., M. Maña López, & E. Puertas Sanz. "Combining Text and Heuristics for Cost-Sensitive Spam Filtering." Proceedings of the Fourth Computational Natural Language Learning Workshop, CoNLL-2000, Lisbon, Portugal, 2000. 99-102.
- Hidalgo, J. M. G. "Evaluating cost-sensitive unsolicited bulk email categorization." In Proceedings of the 2002 ACM Symposium on Applied Computing (SAC), Madrid, Spain, March 10-14, 2002. 615-620.
- Hu, J., M. K. Brown, & W. Turin. "HMM based on-line handwriting recognition." IEEE Trans. Pattern Analysis and Machine Intelligence 18.10 (1996): 1039-1045.
- Hughey, R. & A. Krogh. "Hidden Markov Model for sequence analysis : extension and analysis of the basic method." Computer Applications in the Biosciences 12 (1996): 95-107.
- Hull, David. "Improving Text Retrieval for the Routing Problem using Latent Semantic Indexing." Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 3-6 July 1994. 282-291.
- Ittner, D. J., D. D. Lewis, & D. D. Ahn. "Text categorization of low quality images." Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, 1996. 301-315.
- Jebara, T., & A. Pentland. "Action Reaction Learning : Automatic Visual Analysis and Synthesis of interactive behavior." Proceedings of the 1st International Conference On Computer Vision Systems (ICVS'99), Las Palmas de Gran Canaria, Spain, 13-15 January 1999. 255-272.
- Jelinek, F. "Markov Source Modeling of Text Generation." Impact of Processing Techniques on Communications. Ed. J. K. Skwirzinski. Nijhoff: Dordrecht, 1985.
- Jenkins Charlotte, Mike Jackson, Peter Burden, & Jon Wallis. "The Wolverhampton Web Library (WWLib) and Automatic Classification." Proceeding of the 1st International Workshop on Libraries and WWW, Brisbane, Queensland, Australia, April 14, 1998. <<http://preprints.cern.ch/conf/brisbane/abstract/WWLib.html>> visited on May 27, 2004.
- Joachims, T. "A probabilistic analysis of the rocchio algorithm with tfidf for text categorization." Proceedings of the 14th International Conference on Machine Learning (ICML-97), Nashville, TN, 1997. 143-151.
- Joachims, T. "Text Categorization with Support Vector Machines: Learning with Many Relevant Features." Proceedings of the 10th European Conference on Machine Learning (ECML-98), Chemnitz, Germany, 21-24 April 1998. 137-142.

- Joachims, T. 1999. "Transductive inference for text classification using support vector machines." Proceedings of the 16th International Conference on Machine Learning (ICML-99), Bled, Slovenia, 1999. 200-209.
- Kessler, Brett, Geoffrey Numberg, & Hinrich Schutze. "Automatic Detection of Text Genre." Proceedings of the 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-97), 07-12 July 1997. 32-38.
- Koch, Traugott & Michael Day. "Specification for Resource Description Methods, Part 3: The Role of Classification Schemes in Internet Resource Description and Discovery." Feb. 1997. <<http://www.lub.lu.se/desire/radar/reports/D3.2.3/>> Retrieved on April 30, 2004.
- Koch, Traugott & Anders Ardö. "Automatic classification of full-text HTML-documents from one specific subject area: DESIRE II D3.6a, Working Paper 2." Feb. 11, 2000. <<http://www.lub.lu.se/desire/DESIRE36a-WP2.html>> visited on January 2001.
- Kohavi, R. "A study of cross-validation and bootstrap for accuracy estimation and model selection." Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1995. 1137-1143.
- Kolcz, A. & Joshua Alspector. "SVM-based Filtering of E-mail Spam with Content-specific Misclassification Costs." Proceedings of the TextDM'01 Workshop on Text Mining, the 2001 IEEE International Conference on Data Mining, San Jose, CA, November 29, 2001.
- Koller, Daphne, & Mehran Sahami. "Hierarchically Classifying Documents Using Very Few Words." Proceedings of the 14th International Conference on Machine Learning (ICML-97), Nashville, TN, 1997. 170-178.
- Kubat, Miroslav, Ivan Bratko, & Ryszard S. Michalski. "A Review of Machine Learning Methods." Machine Learning and Data Mining: Methods and Applications. Ed. Ryszard S. Michalski, Ivan Bratko, & Miroslav Kubat. Chichester, England: John Wiley & Sons LTD, 1999.
- Kundu, A. & L. Bahl. "Recognition of handwritten script: a hidden Markov model based approach." Proceedings of the International Conference on Acoustics, Speech and Signal Processing, New York, NY, April 1988. 928-931.
- Lame, Guiraud.. "A categorization method for French legal documents on the Web." Proceedings of the 8th International Conference on Artificial Intelligence and Law, St. Louis, Missouri, 2001. 219-220.

- Lancaster, F. Wilfrid. Information Retrieval Today. Arlington, VA: Information Resources, 1993.
- Landauer, T. D. Laham, B. Rehder, & M. Schreiner. "How well can passage meaning be derived without using word order? A comparison of latent semantic analysis and humans." Proceedings of the 19th Annual Conference of the Cognitive Science Society, 1997. 412-417.
- Lang, K. "NEWSWEEDER: learning to filter netnews." Proceedings of the 12th International Conference on Machine Learning (ICML-95), Lake Tahoe, CA, 1995. 331-339.
- Langley, Pat. Elements of Machine Learning. San Francisco, CA: Morgan Kaufmann, 1996.
- Larkey, L. S. & W. B. Croft. "Combining classifiers in text categorization." Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland, 18-22 August 1996. 289-297.
- Larkey, Leah S. "Automatic essay grading using text categorization techniques." Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 1998. 90-95.
- Larson, Ray R. "Experiments in Automatic Library of Congress Classification." Journal of the American Society for Information Science 43.2 (1992): 130-148.
- Lee, Young-Bae & Sung Hyon Myaeng.. "Text genre classification with genre-revealing and subject-revealing features." Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 2002. 145-150.
- Leek, T. R. "Information Extraction using Hidden Markov Models." Master thesis. University of California at San Diego, 1997.
- Levinson, S. E., *et al.* "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition." Bell Syst. Tech. J. 62.4 (1983): 1035-1074.
- Levinson, S., L. Rabiner, & M. Sondhi. "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition." Bell System Technical Journal 64.4 (1983): 1035-1074.
- Lewis, D. D. "An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task." Proceedings of the Fifteenth Annual International ACM/SIGIR

- Conference on Research and Development in Information Retrieval, Copenhagen, June 21-24, 1992. 37-50.
- Lewis, D. D. & M. Ringuette. "A Comparison of Two Learning Algorithms for Text Categorization." Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, April 1994. 81-93.
- Li, Y. H. & A. K. Jain. "Classification of Text Documents." The Computer Journal 41.8 (1998): 537-546.
- McGrath, William E. "Relationship between Hard/Soft, Pure/Applied, and Life/Nonlife Disciplines and Subject Book Use in a University Library." Information Processing and Management 14.1 (1978): 17-28.
- Maltby, Arthur. Sayers' Manual of Classification for Librarians. 5th ed. London: Andre Deutsch, 1975.
- Mani, Inderjeet. "Recent Developments in Text Summarization." Proceedings of the 10th International Conference on Information and Knowledge Management, Atlanta, Georgia, 5-10 November 2001. 529-531.
- MARC 21 Concise Format for Bibliographic Data. Library of Congress, Network Development and MARC Standards Office. 5 April 2002.
<<http://lcweb.loc.gov/marc/bibliographic/>> visited on July 20, 2002.
- Markey, Karen & Anh N. Demeyer. "Dewey Decimal Classification Online Project: Evaluation of a Library Schedule and Index Integrated into the Subject Searching Capabilities of an Online Catalog." Research Report: OCLC/OPR/RR-86/1. OCLC Online Computer Library Center, Inc., February 28, 1986.
- Markov, A. A. Translation. "An Example of Statistical Investigation in the Text of 'Eugene Onyegin' Illustrating Coupling of 'Tests' in Chains." Proceedings of the Academy of Sciences, St. Petersburg VI Series 1913. 153-162.
- Maron, M. "Automatic Indexing: An Experimental Inquiry." Journal of the Association for Computing Machinery 8.3 (1961): 404-417.
- McCallum, Andrew & K. Nigam. "A comparison of event models for naïve bayes text classification." Proceedings of AAAI-98 Workshop on Learning for Text Categorization, 1998. 137-142.
- McCallum, Andrew, Ronald Rosenfeld, Tom Mitchell, & Andrew Y. Ng. "Improving Text Classification by Shrinkage in a Hierarchy of Classes." Proceedings of the 10th European Conference on Machine Learning (ECML-98), Chemnitz, Germany, 21-24 April 1998. 359-367.

- McCallum, Andrew, Dayne Freitag, & Fernando Pereira. "Maximum Entropy Markov Models for Information Extraction and Segmentation." Proceedings of 17th International Conference on Machine Learning (ICML-00), Stanford, CA, 29 June – 02 July, 2000. 591-598.
- McCallum, Andrew, K. Nigam, J. Rennie, & K. Seymore. "Automating the Construction of Internet Portals with Machine Learning." Information Retrieval Journal 3 (2000): 127-163.
- Miller, D. R. H., T. Leek, & R. M. Schwartz. "A Hidden Markov Model Information Retrieval System." Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, 15-19 August 1999. 214-221.
- Miller, D. R. H., T. Leek, & R. M. Schwartz. "BBN at TREC7: Using Hidden Markov Models for Information Retrieval." Proceedings of the Seventh Text Retrieval Conference, TREC-7 NIST Special Publications 500-242, 1999. 133-142.
- Mitchell, T. M. Machine Learning. New York, NY: McGraw-Hill, 1997.
- Mitchell, T. M., R. M. Keller, & S. T. Kedar-Cabelli. "Explanation-Based Generalization: A Unifying View." Machine Learning 1 (1986): 47-80.
- Mladenec, Dunja. "Machine learning on non-homogeneous, distributed text data." PhD Dissertation. University of Ljubljana, Slovenia, October 1998.
- Mock, Kenricj. "An experimental framework for email categorization and management." Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, 9-13 September 2001. 392-393.
- Mooney, R. J., P. N. Bennett, & L. Roy. "Book recommending using text categorization with extracted information." Proceedings of the AAAI-98 Workshop on Recommender Systems, Madison, WI, 1998. 70-74.
- Mortimer, Mary. Learn Dewey Decimal Classification (Edition 21). Lanham, Maryland: Scarecrow Press, 2000.
- van Mulbregt, P., I. Carp, L. Gillick, S. Lowe, & J. Yamron. "Text Segmentation and Topic Tracking on Broadcast News via a Hidden Markov Model Approach." Proceedings of the 5th International Conference on Spoken Language Processing, Volume VI, Sydney, Australia, 30 November – 4 December 1998. 2519-2522.

- Natarajan, B. K. Machine learning: A theoretical approach. San Francisco, CA: Morgan Kaufmann, 1991.
- Ng, Tou Hwee, Wei Boon Goh, & Kok Leong Low, "Feature selection, perception learning, and a usability case study for text categorization." Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, PA, 27-31 July 1997. 67-73.
- Nigam, Kamel, Andrew McCallum, Thrun Sebastian, & Tom Mitchell. "Text Classification from Labeled and Unlabeled Documents Using EM." Machine Learning 39(2/3) (2000): 103-134.
- Nilsson, Nils J. "Introduction to Machine Learning." 1996.
<<http://robotics.stanford.edu/people/nilsson/mlbook.html> visited 5 September 2003>
visited on 20 August 2003.
- Nottelmann, H. & Norbert Fuhr. Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM-01). Atlanta, Georgia, 2001. 387-394.
- OCLC. "OCLC CatCD for Windows." <<http://www.purl.org/oclc/catcd>> visited on 24 June 2002.
- OCLC WebDewey. <<http://www.oclc.org/dewey/resource/tutorial>> Retrieved on May 9, 2004.
- Oh, H.-J., S. H. Myaeng, & M.-H. Lee. "A practical hypertext categorization method using links and incrementally available class information." Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, 24-28 July 2000. 264-271.
- Olson, Hope A., & John J. Boll. Subject Analysis in Online Catalogs. Englewood, Colorado: Libraries Unlimited, 2001.
- Orwig, R. E., *et al.* "A Graphical, Self-Organizing Approach to Classifying Electronic Meeting Output." Journal of the American Society for Information Science 48.2 (1997): 157-170.
- Palmer, Bernard I. Itself an Education: Six Lectures on Classification. 2nd ed. London: the Library Association, 1971.
- Pannu, A. S. "Using genetic algorithms to inductively reason with cases in the legal domain." Proceedings of the 5th International Conference on Artificial Intelligence and Law. College Park, Maryland, 1995. 175-184.

- Platt, J. "Fast training of SVMs using sequential minimal optimization." Advances in Kernel Methods – Support Vector Learning. Ed. B. Scholkopf, C. Burges, & A. Smola. Cambridge, MA: MIT Press, 1998. 185-208.
- Pommerleau, D. A. "ALVINN: An autonomous land vehicle in a neural network." Technical Report CMU-CS-89-107. Pittsburgh, PA: Carnegie Mellon University. 1989.
- Ponte, J. & W. B. Croft. "A language modeling approach to information retrieval." Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 1998. 275-281.
- Porter, M. F. "An Algorithm for Suffix Stripping." Program 14.3 (1980): 130-137.
- Quinlan, J. R. "Learning logical definitions from relations." Machine Learning 5 (1990): 239-266.
- Quinlan, J. R. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993.
- Rabiner, Lawrence R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." Proceedings of the IEEE 77.2 (1989): 257-286.
- Rabiner, Lawrence R. & B. Juang. "An introduction to hidden Markov models." IEEE ASSP Magazine January 1986: 257-285.
- Rafferty, Pauline. "The Representation of Knowledge in Library Classification Schemes." Knowledge Organization 28.4 (2001): 180-191.
- Rennie, Jason D. M. "Ifile: An application of machine learning to e-mail filtering." Proceedings of KDD-2000 Workshop on Text Mining, Boston, MA, 2000.
- Richmond, Phyllis. "General Theory of Classification." Classification of Library Materials. Ed. Betty G. Bengston & Janet S. Hill, New York: Neal-Schuman Publishers, 1990.
- van Rijsbergen, C. J. Information Retrieval. London, England: Butterworths, 1979.
- Rissland, E. & K. Ashley. "A Case-Based System for Trade Secrets Law." Proceedings of the First International Conference on Artificial Intelligence and Law, Boston, MA, 1989. 60-66.
- Rita, Marcella & Robert Newton. A New Manual of Classification. Hampshire, England: Gower, 1994.

Robertson, S. E., S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford. "Okapi at TREC-3." Proceedings of the Third Text Retrieval Conference, TREC-3 NIST Special Publications 500-226, 1995. 109-126.

Rocchio, J. "Relevance feedback information retrieval." The Smart Retrieval System—Experiments in Automatic Document Processing. Ed. G. Salton. Englewood Cliffs, NJ: Prentice-Hall, 1971. 313-323.

Sahami, Mehran, Susan Dumais, David Heckerman, & Eric Horvitz. "A Bayesian Approach to Filtering Junk E-Mail. Learning for Text Categorization." AAAI Technical Report WS-98-05. Madison, Wisconsin, 1998.
<<http://robotics.stanford.edu/users/sahami/papers-dir/spam.ps>> visited on 23 October 2003.

Sakkis, Georgios *et al.* "Stacking Classifiers for Anti-Spam Filtering of E-Mail." Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP-01), Pittsburgh, PA, 2001.
<<http://www.arxiv.org/abs/cs.CL/0106040>> visited on 23 October 2003.

Salton, Gerard. The SMART Retrieval System-Experiments in Automatic Document Retrieval. Englewood Cliffs, NJ: Prentice Hall, 1971.

Salton, Gerard, A. Wong, & C. Yang. "A Vector Space Model for Automatic Indexing." Communications of the ACM 18.11 (1975): 613-620.

Salton, Gerard. "Developments in automatic text retrieval." Science 253 (1991): 974-979.

Satija, M. P. "Classification: Some Fundamentals, Some Myths, Some Realities." Knowledge Organization 25.1/2 (1998): 32-35.

Schatz, Bruce R. "Information Analysis in the Net: The Interspace of the Twenty-First Century" A Forum, Committee on Information and Communications (CIC) of the National Science and Technology Council, CIC Forum White Paper for America in the Age of Information July 1995. <http://www.hpcc.gov/cic/forum/CIC_Cover.html> visited on 10 July 2002.

Schatz, Bruce R. "Building the Interspace." <<http://csl.ncsa.uiuc.edu/interspace.html>> visited on 15 July 2002.

Schatz, Bruce R. "Information Retrieval in Digital Libraries: Bringing Search to the Net." Science 275 (17 January 1997): 327-334.

Schatz, Bruce R. & Hsinchun Chen. "Digital Libraries: Technological Advances and Social Impacts." IEEE Computer 32.2 (February 1999): 45-50.

- Schneider, Karl-Michael. "A Comparison of Event Models for Naive Bayes Anti-Spam E-Mail Filtering." Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03), 2003. <<http://www.phil.unipassau.de/linguistik/mitarbeiter/schneider/pub/eacl2003.pdf>> visited on 23 October 2003.
- Schütze, H., D. Hull, & J. O. Pedersen. "A Comparison of Classifiers and Document Representations for the Routing Problem." Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, 9-13 July 1995. 229-237.
- Schweighofer, Erich & Dieter Merkl. "A learning technique for legal document analysis." Proceedings of the 7th International Conference on Artificial Intelligence and Law. Oslo, Norway, 1999. 156-163.
- Scott, Mona L. Dewey Decimal Classification, 21st Edition: A Study Manual and Number Building Guide. Englewood, Colorado: Libraries Unlimited, 1998.
- Sebastiani, Fabrizio. "Machine Learning in Automated Text Categorization." ACM Computing Surveys 34.1 (2002): 1-47.
- Seymore, Kristie, Andrew McCallum, & Ronald Rosenfeld. "Learning Hidden Markov Model Structure for Information Extraction." The 16th National Conference on Artificial Intelligence (AAAI-99), Orlando, FL, 18-22 July 1999. AAAI Workshop Technical Report (WS-99-11), 1999. 37-42.
- Shafer, Keith. "A Brief Introduction to Scorpion." 1997a. <<http://orc.rsch.oclc.org:6109/bintro.html>> visited January 2001.
- Shafer, Keith. "Scorpion helps catalog the Web." 1997b. <<http://orc.rsch.oclc.org:6109/basis.html>> visited on 26 August 2003.
- Shafer, Keith. "Evaluating Scorpion Results." 1998. <<http://orc.rsch.oclc.org:6109/eval-sc.html>> visited on 26 August 2003.
- Shafer, K, S. Subramanian, & J. Fausey. "Measures for Evaluating Automatic Subject Assignment of Electronic Resources." OCLC Online Computer Library Center, Inc. 1999. <<http://orc.rsch.oclc.org:6109/measures.html>> visited October 2000.
- Shao, J. "Linear model selection by cross-validation." Journal of the American Statistical Association 88 (1993): 486-494.
- Soper, Mary Ellen, Larry N. Osborne, & Douglas L. Zweizig. "The Librarian's Thesaurus Chicago: a Concise Guide to Library and Information Terms." Chicago, Illinois: American Library Association, 1990.

- Subramanian, Srividhya & Keith Shafer. "Clustering." 1998.
<<http://orc.rsch.oclc.org:6109/clustering.html>> visited on 26 August 2003.
- Stamatatos, E., N. Fakotakis, & G. Kokkinakis. "Text Genre Detection using common word frequencies." Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000), Luxembourg, 31 July – 04 August 2000. 808-814.
- Sullivan, Danny. "Web Directory Sizes." January 1, 2003.
<<http://searchenginewatch.com/reports/article.php/2156411>> visited on 18 July 2003.
- Sun, Aixin, Ee-Peng Lim, & Wee-Keong Ng. "Web classification using support vector machine." Proceedings of the WIDM'02, McLean, Virginia, 08 November 2002.
- Sutton, R. S. "Learning to Predict by the Methods of Temporal Differences." Machine Learning 3 (1988): 9-44.
- Svenonius, Elaine. "Use of Classification in Online Retrieval." Library Resources and Technical Services 27.1 (Jan./Mar. 1983): 76-80.
- Svenonius, Elaine. The Intellectual Foundation of Information Organization. Cambridge, Massachusetts: The MIT Press, 2000.
- Taylor, Arlene G. Introduction to Cataloging and Classification. Englewood, Colorado: Libraries Unlimited, 1992.
- Taylor, Arlene G. The Organization of Information. Englewood, Colorado: Libraries Unlimited, 1999.
- Tesauro, G. "Temporal difference learning and TD-gammon." Communications of the ACM 38.3 (1995): 58-68.
- Thompson, Paul. "Automatic categorization of case law." Proceedings of the 8th International Conference on Artificial Intelligence and Law, St. Louis, Missouri, May 2001. 70-77.
- Thompson, Roger, Keith Shafer, & Diane Vizine-Goetz. "Evaluating Dewey Concepts as a Knowledge Base for Automatic Subject Assignment." February 1997.
<http://www.orc.rsch.oclc.org:6109/eval_dc.html> visited on 26 August 2003.
- Viterbi, A. J. "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm." IEEE Transactions on Information Theory IT.13 (April 1967): 260-269.

Vizine-Goetz, Diane. "Using Library Classification Schemes for Internet Resources." Proceedings of the OCLC Internet Cataloging Colloquium, San Antonio, Texas, 19 January 1996. OCLC Internet cataloging project colloquium position paper, 10 April 2002. 01 July 2002 <<http://staff.oclc.org/~vizine/Intercat/vizine-goetz.htm>>.

Vizine-Goetz, Diane. "OCLC Investigates Using Classification Tools to Organize Internet Data." Proceedings of the 34th Annual Clinic on Library Applications of Data Processing, Illinois University at Urbana-Champaign, Illinois, March 2-4, 1997. 93-105.

Vizine-Goetz, Diane. "Popular LCSH with Dewey Numbers." Annual Review of OCLC Research 1997. (1998).
<<http://digitalarchive.oclc.org/da/ViewObject.jsp?objid=0000003449>> Retrieved on April 30, 2004.

Vizine-Goetz, Diane & Karen Markey. "Characteristics of Subject Heading Records in the Machine-Readable Library of Congress Subject Headings." Information Technology and Libraries 8.2 (1989): 203-209.

Vizine-Goetz, Diane, Carol Hickey, Andrew Houghton, & Roger Thompson. "Vocabulary Mapping for Terminology Services." Journal of Digital Information 4.4 (2004). <<http://jodi.ecs.soton.ac.uk/Articles/v04/i04/Vizine-Goetz/>> Retrieved on April 25, 2004.

Waibel, A., T. Hanazawa, G. Hinton, K. Shikano, & K. Lang. "Phoneme recognition using time-delay neural networks." IEEE Transactions on Acoustics, Speech and Signal Processing 37.3 (1989): 328-339.

Wallis, Jon & Peter Burden. "Towards a Classification-based Approach to Resource Discovery on the Web." Presented at 4th International W4G Workshop on Design and Electronic Publishing, Abingdon, England, 20-22 November 1995.
<<http://www.scit.wlv.ac.uk/wwlib/position.html>> visited on 26 August 2003.

Widrow, B. & S. D. Stearns. Adaptive Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1985.

Williams, Ken. "An Introduction to Machine Learning with Perl." February 3, 2003. O'Reilly Bioinformatics Conference.
<<http://www.campstaff.com/~ken/talks/MLPerl.pdf>> visited on 5 September 2003.

Williamson, Nancy J. The Library of Congress Classification: A Content Analysis of the Schedules in Preparation for Their Conversion into Machine-Readable Form. Washington, D. C.: Library of Congress, Cataloging Distribution Service, 1995.

- Williamson, Nancy J. Knowledge Structures and the Internet. Proceedings of the 6th International Study Conference on the Classification Research, University College London, June 16-18, 1997. 23-27.
- WWLib. "Automatic Classification of Web Resources Using Java and Dewey Decimal Classification." <<http://www.scit.wlv.ac.uk/~ex1253/classifier/>> visited on 26 August 2003.
- Wynar, B. S. Introduction to Cataloging and Classification. 8th ed. Englewood, Colorado: Libraries Unlimited, 1992.
- Yang, Yiming. "Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval." Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 3-6 July 1994. 13-22.
- Yang, Yiming & C. G. Chute. "An example-based mapping method for text categorization and retrieval." ACM Transaction on Information Systems (TOIS) 12.3 (1994): 252-277.
- Yang, Yiming & X. Liu. 1999. "A re-examination of text categorization methods." Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, 15-19 August 1999. 42-49.
- Yang, Yiming, Seán Slattery, & Rayid Ghani. "A study of approaches to hypertext categorization." Journal of Intelligent Information Systems 18.2-3 (2002): 219-241.
- Yamron, J. I., L. Gillick, S. Lowe, & P. van Mulbregt. "A Hidden Markov Model Approach to Text Segmentation and Event Tracking." Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, 12-15 May 1998. 333-336.