



<http://researchspace.auckland.ac.nz>

## ***ResearchSpace@Auckland***

### **Copyright Statement**

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

To request permissions please use the Feedback form on our webpage.

<http://researchspace.auckland.ac.nz/feedback>

### **General copyright and disclaimer**

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the [Library Thesis Consent Form](#) and [Deposit Licence](#).

### **Note : Masters Theses**

The digital copy of a masters thesis is as submitted for examination and contains no corrections. The print copy, usually available in the University Library, may contain corrections made by hand, which have been requested by the supervisor.

**SEMANTIC SPACE MODELS FOR  
CLASSIFICATION OF CONSUMER  
WEBPAGES  
ON METADATA ATTRIBUTES**

---

**Guocai Chen**

*A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of  
Philosophy, The University of Auckland, 2010.*

## **Abstract:**

*A means of dealing with the quantity and quality issues of web-based consumer health resources is the creation of web portals centred on particular health topics and/or communities of users, a strategy that provides access to a more manageably sized corpus of reduced good quality, relevant, information. Breast Cancer Knowledge Online (BCKO) is an example of such a topic-centered portal; it provides a gateway to online information about breast cancer for patients, their families, friends and carers. Such portals are enhanced by metadata elements that help to focus user search, but the maintenance of such information, especially for dynamic Web 2.0 style resources, challenges the sustainability of the portal strategy. This thesis addresses this problem by exploring the feasibility of automated assessment of metadata attributes for consumer health webpages.*

*In this thesis I use Hyperspace Analogue to Language (HAL) to model the language use patterns of webpages as Semantic Spaces. I present and demonstrate methods for automatically inferring non-trivial metadata attributes that have been encoded for BCKO for article tone ('supportive' versus 'medical'), author credentials and disease stage. I introduce a refined use of the classic Decision Forest and a novel Summed Similarity Measure (SSM) to automatically classify online webpages on their Semantic Space models. For the purpose of comparison, I have applied these methods and the well-known SVM algorithm on both BCKO and the popular Reuters21578 dataset. In addition to performance evaluation in terms of random sub-samples, to simulate real use I look at the datasets in their 'natural order' - the order in which the cases occurred chronologically.*

*I find classification accuracy of 90% to 93% for the different BCKO metadata attributes (with SSM always among the top performing methods, and significantly*



*superior to SVM on the author credential attribute) and approximately 98% for distinguishing the two most frequent classes in Reuters21578. In natural order, accuracies reach approximately 90%.*

*These results indicate that language use patterns can be used to automate classification of consumer health webpages with acceptable accuracy. However, our study has been limited to webpages indexed by the BCKO consumer portal and only its metadata attributes. A wider range of websites and metadata attributes needs to be assessed, and the classification results should be compared to end-user feedback.*

## Acknowledgments

I would like to thank my supervisor Professor James Roy Warren, for his supervision, advice and direction. I also would like to thank Peter Bruza and Robert McArthur for their advice on use of Semantic Space models; Joanne Evans, Frada Burstein and the other staff of the Smart Information Portals team at Monash University, Australia, for their advice and support with consumer health metadata; and Vojo Kecman and Pat Riddle for their mentorship on machine learning methods. This work was supported in part by the Australian Research Council Discovery Project DP0665353 and a University of Auckland postgraduate scholarship.

I am deeply indebted to my wife and my family for their generous support and encouragement.

# Table of Contents

Abstract: .....	I
Acknowledgments .....	III
Index of Tables .....	VIII
Chapter 1. Introduction .....	1
1.1. Motivation .....	1
1.2. Background.....	2
1.2.1. Portals.....	2
1.2.2. Breast Cancer Knowledge Online .....	2
1.2.3. Semantic Space Models.....	5
1.3. Research Questions .....	7
1.4. Outline of the thesis .....	8
Chapter 2. Literature Review.....	10
2.1. Foundation Methods.....	10
Singular Value Decomposition (SVD).....	10
Entropy and Information Gain.....	10
Precision, Recall and F-measure.....	11
Distance Measurement methods .....	12
K-Fold Cross Validation and Repeated Random Sub-sampling .....	13
Stop words.....	14
Metadata .....	15
2.2. Semantic Space Models.....	16
2.2.1. Word Frequency (Term-Document Models) .....	16
TF-IDF.....	16
The LSA model .....	17
Probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA) .....	18
2.2.2. Term-Term models (sliding-window) .....	19
Hyperspace Analogue to Language .....	19
Refined HAL .....	21
The COALS and COALS-SVD model .....	22



2.3. Text Classification .....	23
2.3.1. Classification Methods .....	23
ID3 .....	23
Decision Forest .....	23
K-Nearest Neighbour (KNN) .....	24
Adaptive K-Local Hyperplane .....	24
Support Vector Machine .....	25
2.3.2. Previous Examples of Text Classification.....	27
Reuters21578 .....	28
IBM KitCat System .....	29
HAL and emotion and parts of speech .....	30
2.4. Consumer Health Information and Tailoring.....	30
2.4.1. Breast Cancer Knowledge Online .....	31
2.4.2. Recommender Systems .....	32
2.4.3. Computer tailoring .....	33
2.4.4. Comprehensive Health Enhancement Support System (CHESS).....	35
2.4.5. Violet Technology (VT) .....	36
Chapter 3. Methods.....	38
3.1. Data sources .....	38
3.1.1. Transition.....	38
3.1.2. BCKO .....	39
3.1.3. Reuters21578 .....	42
3.2. Algorithms .....	43
3.2.1. Data pre-processing.....	43
3.2.2. Feature Selection.....	43
3.2.3. Classifiers .....	46
Decision Tree .....	47
Round-Robin Feature Allocation .....	50
Summed Similarity Measure on HAL (SSMoHal) .....	51
AKLH .....	51
Support Vector Machine (SVM).....	52
3.3. Procedures.....	52
3.3.1. Exploration .....	52
Transition Dataset .....	52

Adaptive K-Local Hyperplane (AKLH) on BCKO.....	53
3.3.2. Comparing standard voting to summed similarity along the Validation Path for RRDF .....	54
3.3.3. Parameter Optimization .....	54
Window size of HAL.....	54
Columns and Rows .....	55
Number of trees forming the decision forest.....	55
Overlap .....	55
Tie Resolution .....	55
3.3.4. Classification Accuracy (Resampling) .....	56
3.3.5. ‘Natural Order’ Experiments .....	56
3.3.6. Improving Performance on Less Frequent Classes.....	57
3.3.7. Non-Exclusive Classes Classification.....	59
3.3.8. Computational Complexity and Performance .....	60
Chapter 4. Implementation and Java API .....	61
4.1. Data Structure .....	61
4.2. Structure of Program.....	63
4.3. Java API.....	64
4.3.1. A simple example of the API in use: .....	66
4.3.2. A larger example – Customised Google interface: .....	67
Chapter 5. Results.....	71
5.1. Exploration and Parameter Optimisation .....	71
5.1.1. Transition Dataset .....	71
5.1.2. AKLH method.....	73
5.1.3. Comparing voting to summed similarity along the VP for RRDF .....	75
5.1.4. Tuning HAL parameter .....	77
Size of window for building HAL.....	77
Columns and Rows .....	78
Number of trees forming a decision forest .....	79
Overlaps.....	80
Ties.....	81
5.2. Performance Assessment .....	83
5.2.1. Resampling .....	83
5.2.2. Natural Order .....	86



5.2.3. Unbalanced Classes .....	90
5.2.4. Non-Exclusive Classes Classifying .....	92
5.3. Computational complexity analysis of SVM and SSM .....	95
Chapter 6. Discussion .....	97
6.1. Significance .....	97
6.2. Comparison to related work.....	98
6.3. What we have learned .....	100
6.3.1. Feasible.....	100
6.3.2. SSM is an elegant semantic space approach.....	100
6.3.3. Semantic Space not demonstrably better on accuracy than other good methods .....	100
6.3.4. SSM gives the implementer a new choice with different performance characteristic to SVM .....	100
6.3.5. Less frequent classes .....	101
6.3.6. Non-exclusive classes .....	101
6.4. Limitation.....	101
6.5. Conclusion and future direction .....	102
References.....	105
Appendix A: Stop Words list .....	113

# Index of Figures

Figure 1 The Breast Cancer Knowledge Online search profile page. ....	3
Figure 2 Medical search results of 'breast reconstruction' .....	4
Figure 3 Supportive search results of 'breast reconstruction' .....	5
Figure 4 from [19] a) concrete nominal concepts, b) grammatical concepts, and c) abstract concepts .....	6
Figure 5 Outline of the thesis .....	9
Figure 6 Singular Value Decomposition.....	10
Figure 7 Prediction table .....	11
Figure 8 a) City block. b) Euclidean. c) Cosine. d) Correlation.....	13
Figure 9 Illustration of 5-fold cross validation.....	14
Figure 10 a) Illustration of pLSA; b) illustration of LDA; where positive-real vector $\alpha$ is the parameter of the uniform Dirichlet prior on the per-document topic distributions, $\beta$ is the parameter of the uniform Dirichlet prior on the per-topic word distribution, $\theta$ is the topic distribution for a document, $z$ is a topic for a word in a document, $w$ is a specific word, $N$ is the number of words in a document and $M$ is the number of documents. ....	18
Figure 11 HAL values for word 'masses' with window size 3 .....	20
Figure 12 Illustration of KNN .....	24
Figure 13 Graphical presentation of the AKLH method .....	25
Figure 14 Illustration of linear SVM. In a) and b) the distance between the hyperplane and the nearest data point on each side is called 'margin'. The margin in a) is smaller than that in b). The hyperplane with the maximum margin forms the SVM classifier.....	26
Figure 15 Plots of micro-averaged F1 (leftmost) and macro-averaged F1 (rightmost) obtained with supervised weighting and a $\xi = 0.0$ reduction factor [100]. Plots indicate results obtained with Rocchio (top), K-NN (middle) and SVMs (bottom). The X axis indicates the three subsets of Reuters-21578 [55].....	27
Figure 16 Excerpt from webpage of type 'medical' .....	40
Figure 17 Excerpt from webpage of type 'supportive' .....	41
Figure 18 An induced decision tree for 70 training websites of each supportive and medical; words (e.g., 'Breast') are the decision words for that node in the tree, with the entropy reduction in parentheses; number pairs at nodes indicate number of medical and supportive documents, respectively. ....	49
Figure 19 Decision tree showing number of 'medical' and 'supportive' cases, decision word (and its entropy gain in parentheses) annotated with validation path arrows and similarity measures. (a) Decision flow for a medical webpage in the decision tree resulting in a correct classification (the page is: <a href="http://www.cancerbacup.org.uk/info/goserelin.htm">http://www.cancerbacup.org.uk/info/goserelin.htm</a> ); (b) Correct classification of a supportive webpage on the same decision tree (see <a href="http://my.webmd.com/content/chat_transcripts/1/103833.htm">http://my.webmd.com/content/chat_transcripts/1/103833.htm</a> ); (c) classification of a medical webpage that is missing the keyword 'Treatment' (see <a href="http://theoncologist.alphamedpress.org/cgi/content/full/5/5/393?maxtoshow=&amp;HITS=10&amp;hits=10&amp;RESULTFORMAT=&amp;titleabstract=b">http://theoncologist.alphamedpress.org/cgi/content/full/5/5/393?maxtoshow=&amp;HITS=10&amp;hits=10&amp;RESULTFORMAT=&amp;titleabstract=b</a> ) .....	49
Figure 20 Stages of Transition in chronic illness (from [1]) .....	53

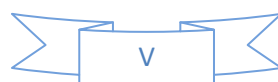


Figure 21 Illustration of the over-sampling for the unbalanced datasets.....	58
Figure 22 Illustration of non-exclusive classes where the overlapping portion of the two classes is flagged as a third class (e.g. Medsup) .....	59
Figure 23 Illustration of the data structures: a) $W_{ij}$   HAL indicates $\text{HashMap}\langle\text{String}, \text{Double}\rangle$ for the non-zero HAL value of the $i$ th word with its $j$ th word - an HMmatrix stores a sparse HAL matrix; b) ID is the id of a webpage, HMmatrix is its HAL matrix; c) a Node stores all the training dataset, each HMclass maps a data group. ....	63
Figure 24 The structure of the software system .....	64
Figure 25 Output of the example program using the API .....	67
Figure 26 Interface for the customised search engine.....	68
Figure 27 Tagged search result for "my story breast cancer" .....	69
Figure 28 Tagged search result for 'tamoxifen' .....	70
Figure 29 Manually produced cluster map of largest sense-of-self terms for Susan (overlaid with subsequently selected projection axes) [2].....	72
Figure 30 Segment of manually-produced cluster map for Cyndi [2] .....	72
Figure 31 Projection of Susan's discussion entries by month [2].....	73
Figure 32 Accuracies for different HAL window sizes for AKLH [141]. ....	74
Figure 33 Classification accuracy by number of cases available of each type (medical and supportive) for AKLH [141].....	75
Figure 34 Comparison of accuracies using different algorithms [141]. ....	75
Figure 35 Comparison between standard voting and summed similarity along the VP for RRDF on earn vs. acq in Reuter21578 (accuracy and its 95% confidence interval) .....	76
Figure 36 Comparison between standard voting and summed similarity along the VP for RRDF on medical vs. supportive (accuracy and its 95% confidence interval) .....	76
Figure 37 Comparison between standard voting and summed similarity along the VP for RRDF on early vs. advanced (accuracy and its 95% confidence interval).....	77
Figure 38 Comparison between standard voting and summed similarity along the VP for RRDF on lay vs. clinician (accuracy and its 95% confidence interval) .....	77
Figure 39 Accuracies affected by the window size of HAL ('reu' – Reuters21578 ; 'ms' – medical v. supportive; 'era' – early v. advanced; 'lc' – lay v. clinical). ....	78
Figure 40 Accuracies affected by number of columns selected ('reu' – Reuters21578 ; 'ms' – medical v. supportive; 'era' – early v. advanced; 'lc' – lay v. clinical). ....	79
Figure 41 Accuracies affected by number of rows selected ('reu' – Reuters21578 ; 'ms' – medical v. supportive; 'era' – early v. advanced; 'lc' – lay v. clinical). ....	79
Figure 42 Accuracies affected by the number of trees in Decision Forest ('reu' – Reuters21578 ; 'ms' – medical v. supportive; 'era' – early v. advanced; 'lc' – lay v. clinical). ....	80
Figure 43 Accuracies affected by Overlaps ('reu' – Reuters21578 ; 'ms' – medical v. supportive; 'era' – early v. advanced; 'lc' – lay v. clinical).....	80
Figure 44 Accuracies after changing the ties to right branch for all four datasets. ....	82
Figure 45 Classification of the two most popular article categories (earn and acq) in Reuters21578 .....	83
Figure 46 Classification of medical vs supportive BCKO articles .....	84
Figure 47 Classification of BCKO articles by disease stage (early vs advanced) .....	84
Figure 48 Classification of BCKO articles by author qualification (clinician vs lay) .....	85

Figure 49 Accumulated accuracy of earn vs acq in Reuters21578 in natural order. b, c and d) Each test case is classified using SSM, RRDF and SVM: 1 is correct, 0 is incorrect. ....	86
Figure 50 a) Accumulated accuracy of medical vs supportive in BCKO in natural order. b, c and d) Each test case is classified using SSM, RRDF and SVM: 1 is correct, 0 is incorrect. ....	87
Figure 51 a) Accumulated accuracy of early vs advanced in BCKO in natural order. b, c and d) Each test case is classified using SSM, RRDF and SVM: 1 is correct, 0 is incorrect. ....	88
Figure 52 a) Accumulated accuracy of lay vs clinician in BCKO in natural order. b, c and d) Each test case is classified using SSM, RRDF and SVM: 1 is correct, 0 is incorrect. ....	89
Figure 53 a) Refined SSM on earn vs. acq in Reuter21578; b) Each test case is classified using refined SSM: 1 is correct, 0 is incorrect.....	90
Figure 54 a) Refined SSM on medical vs. supportive; b) Each test case is classified using refined SSM: 1 is correct, 0 is incorrect.....	91
Figure 55 a) Refined SSM for early vs. advanced; b) Each test case is classified using refined SSM: 1 is correct, 0 is incorrect. ....	91
Figure 56 a) Refined SSM for lay vs. clinician; b) Each test case is classified using refined SSM: 1 is correct, 0 is incorrect. ....	92
Figure 57 Plotting of 1000 cases by the difference between the supportive and medical HAL values for RRDF.....	93
Figure 58 Plotting of 1000 cases by the difference between the early and advanced HAL values for RRDF.....	93
Figure 59 Plotting of 1000 cases by the difference between the supportive and medical HAL values for SSM (note there are ~230 occurrences of medical at the left boundary and ~140 occurrence of supportive at the right boundary of the graph).....	94
Figure 60 Plotting of 1000 cases by the difference between the early and advanced HAL values for SSM (note there are ~95 occurrences of early at the left boundary and ~81 occurrence of advanced at the right boundary of the graph).....	94
Figure 61 Illustration of the computation complexity of SSM and SVM .....	95

## Index of Tables

Table 1 A list of Stop words starting with 'h' .....	14
Table 2 HAL matrix for the sample text for window size of 3 .....	20
Table 3 Refined HAL matrix for window size of 3.....	22
Table 4 Comparison of three technologies for matching consumers to content. ....	31
Table 5 Largest HAL vector sums for text corpora based on the Medical and the Supportive web sites.....	44
Table 6 Highest-value components of HAL matrices for 80 Medical and 80 Supportive web sites.....	45
Table 7 Largest values for HAL vectors of projection axes [2] .....	73
Table 8 BCKO and Reuters21578 classification accuracy (based on 100 resamples).....	85

## Chapter 1. Introduction

### 1.1. Motivation

When confronted with a healthcare situation, people are increasingly turning to the Internet for information to aid in understanding diagnoses, deciding on treatment options and seeking psychosocial support for themselves, their family and their friends [3].

Escalating healthcare costs are one of the key drivers of increasing interest in the provision of health information on the web from a health consumer perspective [4]. The potential way for patients to reduce costs is to improve their abilities to help themselves by searching out relevant online health information and/or sharing health information with other patients; they can thus make informed choices and more actively participate in decisions surrounding treatment choices, better monitor their condition, and have more efficient and effective interactions with medical professionals [5].

The wealth of information on the Internet not only provides abundant choices to users, but also engenders a problem: which source of information is the most appropriate? Vast quantities of health information are being made available online by a number of providers including government agencies, pharmaceutical companies and other commercial enterprises, charity organisations, community groups and individuals, to service the information needs of medical professionals and healthcare consumers. As a result, a keyword search using any of the major search engines on most healthcare topics will bring up thousands, hundreds of thousands, and even millions of hits of varying quality and relevance to a person's particular health and life situation. The resulting information overload, where the amount of information exceeds a person's ability to process it [6], can often add stress to an already stressful situation. Consequently, there is much interest in how the quality, relevance, authority and accuracy of online information can be assessed in a timely manner by both healthcare consumers and medical professionals alike [7-8].

## 1.2. Background

### 1.2.1. Portals

Many projects have been devised to address information overload and investigate ways in which timely, differentiated access to quality online healthcare resources can be provided. One strategy is to create comprehensive repositories of high-quality health information (or links to such information). The US National Library of Medicine's MedlinePlus (<http://medlineplus.gov>), Australian HealthInsite (<http://www.healthinsite.gov.au>) and the Geneva-based Health on the Net (<http://www.hon.ch>) are examples of such portals. The provision of web portals centred on particular health topics and/or communities of users is another strategy [9-10] which aims to provide access to a reduced corpus of information resources that meet quality and relevance criteria. In charge of the selection and classification of resources for the portal is the curator, an administer of the web portal; coding the index of the resources for the portal users; and updating of the metadata for those resources to validate that the links and contents of the resources are valid and appropriate. Portals can be further augmented by capturing and creating descriptive metadata about resources selected for inclusion. This structured, value-added information can then be used by portal users in searching, filtering, ranking, and in making judgements about what information is relevant to their needs and in which they wish to place their trust.

### 1.2.2. Breast Cancer Knowledge Online

Breast Cancer Knowledge Online (BCKO), developed through collaboration between Monash University, BreastCare Victoria and the Breast Cancer Action Group, is an example of a topic-centered portal, providing a gateway to online information about breast cancer of relevance to patients, their families, friends and carers. In response to the user studies and needs analysis undertaken in the initial stages of the BCKO project, the portal incorporates metadata that describe relevant resources from a user-centred perspective [11]. Included in the description of resources are metadata about the type and style of information, the stage of breast cancer to which it relates, and the credentials of the author. The search interface allows portal users to indicate their information preferences along these lines.

## Introduction

While usability studies have shown a high degree of satisfaction with the resultant portal, questions as to its scalability have been raised. Reliance on manual methods of metadata creation are problematic given the volume of information available online and its volatile, dynamic and complex nature [12]. In the case of BCKO, user information needs analysis identified the desire for more access to personal stories of breast cancer experiences, which are often buried deep in the result sets of the major search engines. Better tools to support metadata coders in identification and labelling of web resources to suit their user communities are needed. Thus, the desire to increase the sustainability and quality of such portals motivates investigation into automated support for the generation of metadata describing relevant resources from a user-centred perspective. Figure 1 shows the BCKO search preferences page: the two groups of categories circled in blue and red are of particular interest for this project.

Home Personalised Search Breast Cancer Topics Simple Search Help

Personalised Search

Home > Personalised Search

**You may select all, some or none of the following categories**

**I want information for a woman aged**

☐ Under 40 ☐ 40-49 ☐ 50-70 ☐ Over 70

**I want information on**

☐ Early Breast Cancer ☐ Recurrent Breast Cancer ☐ Advanced Breast Cancer

**I want information for**

☐ Self ☐ Partner-Spouse ☐ Friend ☐ Parent ☐ Child

**I want information which is**

☐ Plain Brief ☐ Plain Detailed ☐ Scientific Brief ☐ Scientific Detailed

**The type of information I'm looking for is**

☐ MEDICAL ☐ SUPPORTIVE ☐ PERSONAL

Enter Search Term

e.g: breast reconstruction

Search Clear Search

Copyright 2005 © Monash University, Melbourne, Australia.  
All content and works posted on this website are owned and  
copyrighted by the Monash University, Australia. All rights reserved

Figure 1 The Breast Cancer Knowledge Online search profile page.



# Introduction



The screenshot shows the BCKO Online website interface. At the top, there is a navigation bar with links: Home, Personalised Search, Breast Cancer Topics, Simple Search, and Help. Below this is a large green button labeled 'Personalised Search' with a magnifying glass icon. The main content area displays search results for the query 'Medical AND breast AND reconstruction'. The results are numbered 1 to 10 of 26. The first four results are listed, each with a title, description, quality report, and full resource details link.

Results 1- 10 of 26 for "Medical AND breast AND reconstruction" [NEXT >](#)

- [Autoogenous breast reconstruction with the deep inferior epigastric perforator flap](#)**  
Description: **The abstract of a medical journal article which outlines the differences between DIEP and tram flap breast reconstruction. (American)**  
Quality report: This material was created by clinicians and published by a commercial group. The content has undergone editorial review. References are not included and the subject matter represents current personal opinion. The purpose is educational/informative and is of a non-controversial nature.  
Full Resource Details
- [Diep flap \(lower abdomen\)](#)**  
Description: **A flash interactive presentation of the DIEP flap reconstruction technique. Text accompanies the visuals (diagrams) as well as audio. The information includes an explanation of which muscles and tissues are used in the breast reconstruction. (American)**  
Quality report: This material was created by clinicians and published by a commercial group. The content has not undergone review. References are not included and the subject matter represents current consensus opinion. The purpose is educational/informative and is of a non-controversial nature.  
Full Resource Details
- [DIEP flap breast reconstruction](#)**  
Description: **Information on the breast reconstruction technique known as 'DIEP Flap'. Included are diagrams depicting the parts of the body affected; explanations of the surgical procedure involved and a step-by-step explanation of the sequence involved in the surgery. (American)**  
Quality report: This material was created by clinicians and published by a medical organisation. The content has undergone editorial review. References are not included. The subject matter represents current consensus opinion. The purpose is educational/informative and is of a non-controversial nature.  
Full Resource Details
- [The reconstruction of breasts](#)**  
Description: **An overview of the various types of breast reconstructive surgery including: immediate and delayed surgery; implant surgery and autologous surgery (use of the patient's tissues). Photographs of reconstructions are included. (Canadian)**

Figure 2 Medical search results of 'breast reconstruction'

Figure 2 shows the 'medical' search results of 'breast reconstruction'. From the short descriptions, we can see that those first four linked articles are all technique resources about breast reconstruction.

# Introduction

The screenshot shows the BCKOnline website interface. At the top, there is a navigation bar with links: Home, Personalised Search, Breast Cancer Topics, Simple Search, and Help. Below this is a large green button labeled 'Personalised Search' with a magnifying glass icon. The main content area displays search results for the query 'Supportive AND breast AND reconstruction'. The results are numbered 1 to 6. The first three results are visible:

- 1. [Breast reconstruction](#)**  
Description: The first page of a series of webpages reviewing the various options available to women considering breast reconstruction. Included in the discussion is information on the various surgical procedures such as, implants, DIEP flap reconstructions and immediate versus delayed reconstruction. Also included is a section on 'selecting the right option'. (American) NOTE TO USERS: IN ORDER TO VIEW THIS WEBPAGE THE PUBLISHERS REQUIRE THAT YOU FIRST REGISTER WITH THEM. REGISTRATION IS FREE AND PERSONAL DETAILS WILL NOT BE COLLECTED. YOU WILL AUTOMATICALLY BE TAKEN TO THE REGISTRATION PAGE ONCE YOU CLICK ON THE TITLE LINK.  
Quality report: This material was created by clinicians and published by a commercial organisation. The content has undergone editorial review. References are included. The subject matter represents current case studies. The purpose is educational/informative and is of a non-controversial nature.  
Full Resource Details
- 2. [Show me](#)**  
Description: The home page of a series of links to individual case studies of women of varying ages who have undergone either mastectomy, lumpectomy, and/or reconstruction. The linked pages provide the user with the individual's personal story as well as photographs of the results of surgery. (American)  
Quality report: This material was originally created and published by a consumer organisation. The content has undergone editorial review. References are not included and the subject matter represents current personal opinion. The purpose is educational/informative and is of a non-controversial nature.  
Full Resource Details
- 3. [Breast reconstruction?](#)**  
Description: The home page of a series of articles on breast reconstruction. Links are provided to the following topics: what is breast reconstruction; what are the most common methods of breast reconstruction; what about nipple reconstruction; how to check my breasts after reconstruction; questions to ask the plastic surgeon; and recommended reading. (Canadian)

Figure 3 Supportive search results of 'breast reconstruction'

Figure 3 shows the 'supportive' search results of 'breast reconstruction'. These first three webpages contain various kinds of support information. While the substantive topic of the articles is much the same in both the 'medical' and 'supportive' results, one can see the difference in the tone of the articles, with the former being more about the facts of relevant procedures per se and the latter more directly focused on patient choice and patients' experiences.

## 1.2.3. Semantic Space Models

A high-dimensional Semantic Space model is a mathematical representation of a large corpus of textual material which can numerically represent the meanings of words.

## Introduction

Such a model is based on the frequency distribution of words that are immediately adjacent and/or nearby to any given target word, and is computed over a large language corpus (possibly containing millions of words) [13-14]. Words that occur in similar contexts are deemed to be ‘contextually similar,’ and tend to be of like or related meaning. There is a large family of ‘distributional semantics’ [15], ‘semantic vector’ [16] or ‘semantic space’ [17] representations that quantify similarity of meaning of terms or of whole documents.

Hyperspace Analogue to Language (HAL) is one specific sliding-window (see Section 2.2.2) model/approach for automatic construction of a Semantic Space model from a corpus of text [18]. In previous research, Burgess and Lund [18-20] demonstrated that abstract word categories (e.g., their part of speech – noun, verb, etc.) and emotional connotation of words could be represented with HAL matrices. By using Multi-Dimensional Scaling (MDS), the distance between two HAL vectors representing two words could be simulated by their respective lower dimensional points (notably, in 2D). In simulations plotting the clustering of words, they found words were compellingly categorized: nouns close to other nouns, emotionally positive words close to other positive words, negative words close to other negative words. Given these properties of HAL, we sought to apply HAL as a source of classification features for the problem of identifying metadata values for topic-centred consumer portals such as BCKO.

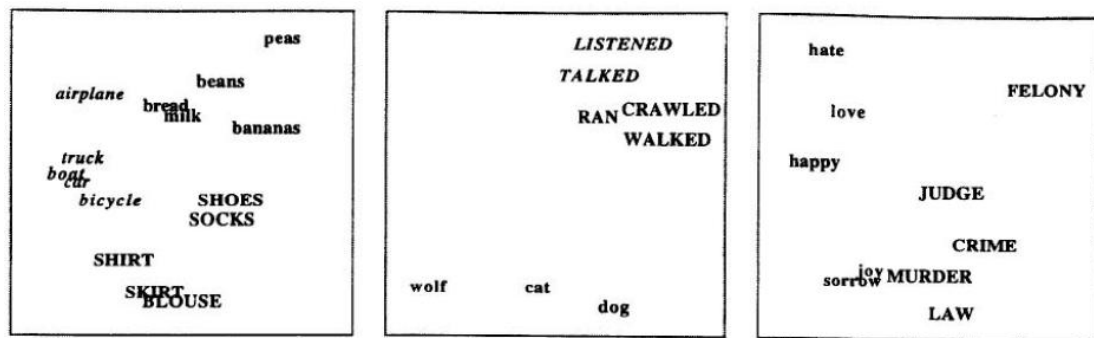


Figure 4 from [19] a) concrete nominal concepts, b) grammatical concepts, and c) abstract concepts

Figure 4 is extracted from [19]. In this figure, the high-dimensional vectors have been plotted into 2D using the MDS algorithm, and we can see that vectors (i.e. positions on the graph) are grouped by their domains. Moreover, a previous study [21] has shown that sliding-window based semantic space models tend to encode strong

# Introduction

---

associations between terms of the same part-of-speech, and often outperform term-document based models in synonym and antonym tests. On the other hand, term-document based models tend to correlate better with free association norms.

This intriguing performance of HAL in distinguishing a wide range of relationships among words (as per above) encourages us to explore the development of HAL as the basis of a document classifier.

## 1.3. Research Questions

One of the key questions this project is addressing is how the metadata, which enrich the user experience and allow differentiated access to resources based on personal information needs, can be created in sustainable and scalable ways. Manual methods of metadata creation cannot keep up with the vast quantities of healthcare information being made available online, and cannot easily respond to their increasing dynamism, complexity and volatility [12]. In addition, those responsible for selecting resources for inclusion in a portal's knowledge repository of metadata descriptions need more sophisticated tools for discovering potential resources of relevance. Further development of the portal therefore requires investigation into how the generation of metadata describing relevant resources from a user-centred perspective can be automated. Moreover, with the continued growth of Web 2.0 content – e.g., blogs, newsfeeds, wikis – the challenge of maintaining accurate metadata will only be increasing.

Semantic Space models in general, and HAL models in particular, appear to have good potential as the basis for identification of subtle aspects of webpage tone that relate to relevant consumer health meta-data attributes.

In this context, the research question for this thesis is:

*Can we use Semantic Spaces to identify relevant metadata values for consumer health webpages?*

Or, more specifically:

*Can we develop an accurate classifier based on attributes of HAL models to identify the membership of consumer health webpages with respect to metadata attribute values such as those used in BCKO?*

## 1.4. Outline of the thesis

In this thesis I describe novel document classifiers tailored to exploitation of HAL's Semantic Space model, particularly a Round-Robin Decision Forest (RRDF) classifier and what I call a Summed Similarity Measurement (SSM) classifier. The chief novelty of our approach is in finding methods to work with the HAL matrix relatively directly, without using computationally expensive dimensional reduction techniques such as Singular Value Decomposition (SVD) and exploiting the HAL matrix structure, as well as simply in our choice of application of Semantic Spaces to requirements derived from experience with consumer health portals. I assess these algorithms based on BCKO's metadata attributes and, to provide a well-known basis of comparison, the widely-used Reuters21578 data set. In view of its known good performance, widespread application and openly available implementation, we compare our algorithms to a Support Vector Machine (SVM) [22] utilising both HAL attributes and the more traditional word frequencies. Our objective in this research is to determine the feasibility of making such automated classifications, including the relationship of accuracy to the amount of training data provided for the various classes of algorithms we consider, with an eye to utilisation of such methods to support consumer health informatics.

In Chapter 2, I review the existing methods for classification and a few semantic models, as well as previous works on text classification. In Chapter 3, I describe the algorithms and related methods I have applied to this classification problem. In Chapter 4, I introduce the data structure and the structure of the program I have used in this project; I also introduce the interface of the Java API and two examples applying this API. Chapter 5 presents the classification results and time complexity analysis. Chapter 6 discusses the practical utility of the research and makes comparison with related work, as well as describing limitations and future work. This is followed by a bibliography of references cited in the chapters of the thesis.

Figure 5 depicts the thesis structure.

# Introduction

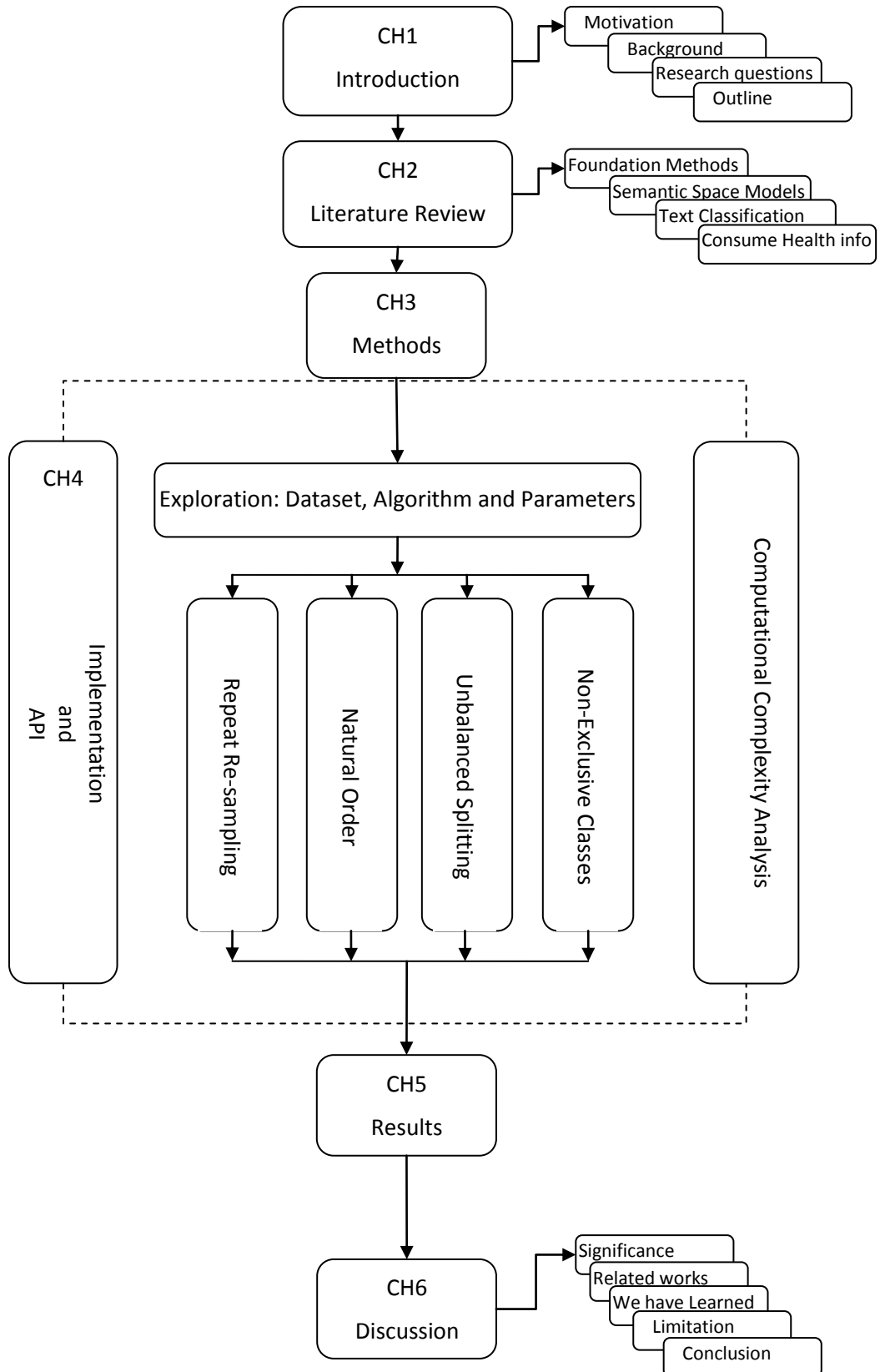


Figure 5 Outline of the thesis



## Chapter 2. Literature Review

### 2.1. Foundation Methods

#### Singular Value Decomposition (SVD)

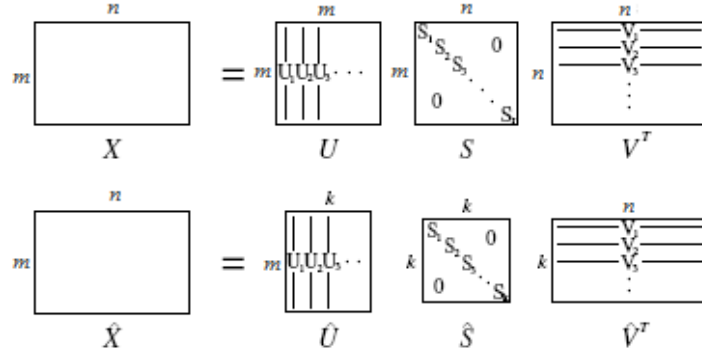


Figure 6 Singular Value Decomposition

An  $m \times n$  matrix  $X$  can be written using a product of three matrices,  $X = USV^T$ , where  $U$  is an  $m \times m$  matrix,  $S$  is an  $m \times n$  diagonal matrix containing the singular values and  $V^T$  is an  $n \times n$  matrix,  $U$  and  $V$  are called left and right singular vectors respectively (as shown in the top diagram in Figure 6). Such a factorisation is called a singular-value decomposition of  $X$ .

If the three matrices comprising the SVD are resequenced such that the singular values in matrix  $V$  are in decreasing order, they can be truncated to smaller size (see  $U$  is  $m \times k$ ,  $S$  is  $k \times k$  and  $V$  is  $n \times k$ , where  $k < n$ ). It can be shown that the product of these reduced matrices is the best rank  $k$  approximation (bottom diagram in Figure 6), in terms of sum of squared error, to the original matrix  $X$ .

#### Entropy and Information Gain

ID3 identified the criteria to split the training dataset in data classification. Two terms are involved in ID3: ‘entropy’ and ‘information gain’, where entropy represents the level of disorder of the dataset, and information gain measures the reduction in entropy after splitting [23]. For a dataset  $S$ , the entropy is

## Literature Review

---

$$Entropy(S) = - \sum p_i \log_2 p_i \quad [2-1]$$

Where  $p_i$  is the proportion of the  $i^{th}$  subset in S ( $p_i = \frac{|S_i|}{|S|}$ ).

For a binary split, the information gain should be

$$Gain(S) = Entropy(S) - \frac{|S_1|}{|S|} Entropy(S_1) - \frac{|S_2|}{|S|} Entropy(S_2) \quad [2-2]$$

Where  $S = S_1 + S_2$ .

### Precision, Recall and F-measure

		Predicted class	
		P	N
Actual class	T	TP	FN
	F	FP	TN

**Figure 7 Prediction table**

Figure 7 illustrates the possible outcomes for a classifier which predicts (P = positive or N = negative) when an attribute applies to a case (where it is in fact T = true or F = false for that case).

‘Precision’ and ‘recall’ are widely used to measure how accurate a statistical classification is. Precision is a measure of accurateness or correctness, and recall is a measure of completeness. Precision and recall are then defined as:

$$\text{Precision} = \frac{tp}{tp+fp} \quad [2-3]$$

$$\text{Recall} = \frac{tp}{tp+fn} \quad [2-4]$$

A measure that combines precision and recall is the harmonic mean of precision and recall, the traditional ‘F-measure’ or balanced F-score:

$$F = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad [2-5]$$



## Literature Review

Common refinements of the **F-measure** includes the Macro-averaged F-Measure and the Micro-averaged F-Measure [24]. In micro-averaging, the F-measure is computed globally over all category decisions, giving equal weight to each case irrespective of the category to which it belongs. Precision and recall are obtained by summing over all individual decisions:

$$\text{Precision} = \frac{\sum_{i=1}^M tp_i}{\sum_{i=1}^M (tp_i + fp_i)} \quad [2-6]$$

$$\text{Recall} = \frac{\sum_{i=1}^M tp_i}{\sum_{i=1}^M (tp_i + fn_i)} \quad [2-7]$$

In macro-averaging, the F-measure is computed locally over each category, after which the average is taken over all categories.

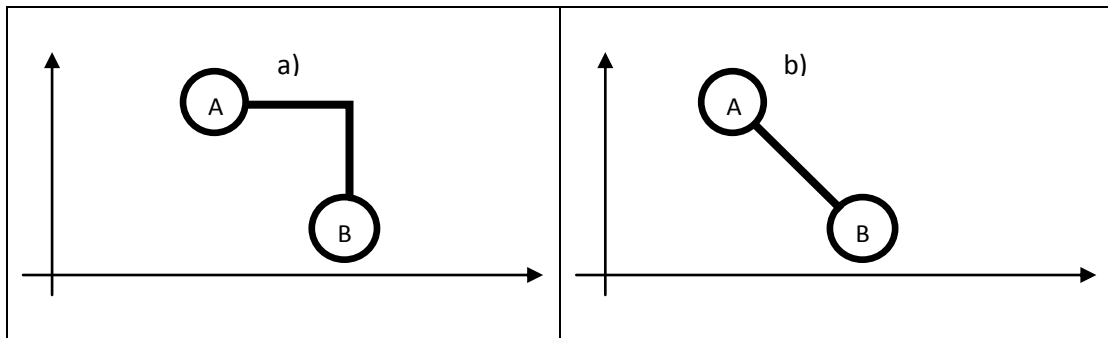
$$F = \frac{\sum_{i=1}^M F_i}{M} \quad [2-8]$$

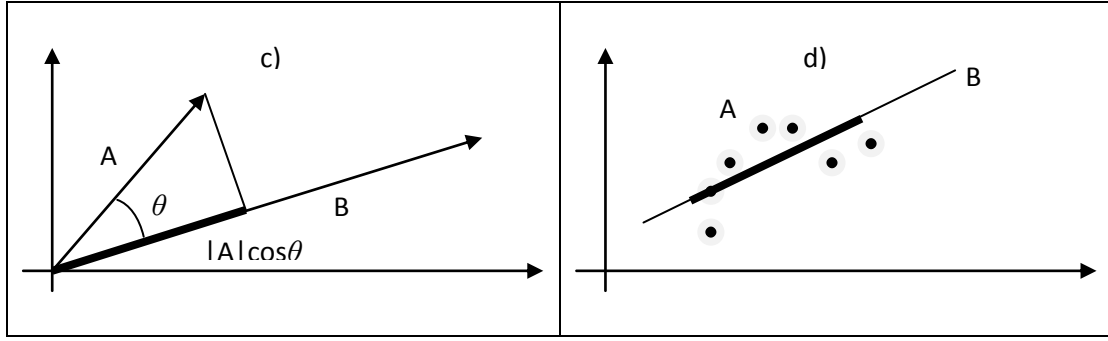
where M in 2-6, 2-7 and 2-8 is the number of categories, and  $tp_i$ ,  $fp_i$  and  $fn_i$  are the frequencies of true positives, false positives and false negatives respectively for class  $i$ .

Notably, when all classes have equal frequency, micro-averaged F is simply the accuracy.

### Distance Measurement methods

Distance Measurements can serve to measure the similarity of two vectors. There are a few common measurements [25-27]:





**Figure 8 a) City block. b) Euclidean. c) Cosine. d) Correlation.**

As shown in Figure 8, the distance between A and B is:

$$\text{City Block: } D(A, B) = (\sum |A_i - B_i|)^2 \quad [2-9]$$

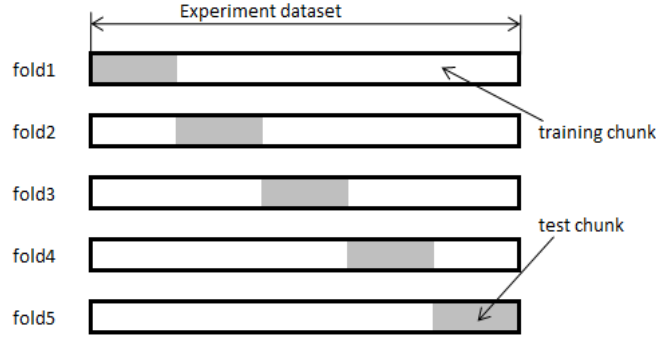
$$\text{Euclidean: } D(A, B) = \sum (A_i - B_i)^2 \quad [2-10]$$

$$\text{Cosine: } D(A, B) = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}| \times |\mathbf{B}|} = \frac{\sum (A_i B_i)}{\sqrt{\sum A_i^2 \sum B_i^2}} \quad [2-11]$$

$$\text{Correlation: } D(A, B) = \frac{\Delta \mathbf{A} \cdot \Delta \mathbf{B}}{|\Delta \mathbf{A}| \times |\Delta \mathbf{B}|} = \frac{\sum ((A_i - \bar{A})(B_i - \bar{B}))}{\sqrt{\sum ((A_i - \bar{A})^2 \sum (B_i - \bar{B})^2)}} \quad [2-12]$$

## **K-Fold Cross Validation and Repeated Random Sub-sampling**

K-Fold Cross Validation [28] is a method to reduce systematic errors in an experiment. In K-Fold Cross Validation each class of data is divided into K slots, and each slot in a data class in turn acts as the test set, with the rest of each data class used as the training set. Figure 9 visually depicts the allocation of data in a cross-fold validation for k=5. Refer to [29-33] for some applications of this methods.



**Figure 9 Illustration of 5-fold cross validation**

Repeated Random Sub-sampling [34] is a type of cross validation. Rather than splitting the dataset equally into K folds, Repeated Random Sub-sampling randomly selects a group of samples from the dataset as validation data and the rest is training data which forms the model of a classifier. The classification accuracy for this splitting is then computed by predicting the class memberships of the validation data with this classifier. The result is averaged over a number of splits [35-36].

## Stop words

‘Stop words’ are extremely common words with high frequency which are not considered meaningful in a given context [37-40]. Stop words are normally removed in information retrieval or searching to save space and speed search engine. A list of common stop words starting with ‘h’ based on [41] (see <http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop>) is shown in Table 1.

**Table 1 A list of Stop words starting with 'h'**

<i>h</i>	<i>had</i>	<i>hadn't</i>
<i>hardly</i>	<i>has</i>	<i>hasn't</i>
<i>haven't</i>	<i>having</i>	<i>he</i>
<i>hello</i>	<i>help</i>	<i>hence</i>
<i>here</i>	<i>here's</i>	<i>hereafter</i>
<i>herein</i>	<i>hers</i>	<i>howbeit</i>
<i>herself</i>	<i>hi</i>	<i>him</i>

## Literature Review

---

<i>his</i>	<i>hither</i>	<i>hopefully</i>
<i>have</i>	<i>happens</i>	<i>hereupon</i>
<i>he's</i>	<i>hereby</i>	<i>her</i>
<i>however</i>	<i>how</i>	<i>himself</i>

Single-character words, prepositions, auxiliaries, pronouns and some very common adverbs are usually treated as stop words. However, there is no definite list of stop words, researchers defining their own list according to the domain of research.

### Metadata

Metadata [42] are also called ‘data about data’ or ‘information about information’. This is structured information that helps users in accessing an information resource, which can be any level of aggregation or any model, such as a collection, a single resource or a constituent part of some other resource (e.g. an image in a webpage or different edition of a book). Metadata are normally stored in a database system or HTML/XML files.

The Dublin Core is the ISO standard for metadata that define the cross-domain information resource, such as video, sound, image, text, and composite media like web pages. Typically Dublin Core metadata are used for online resources in the format of xml [43].

The Australian Government Locator Service (AGLS, see <http://www.agls.gov.au/>) is an application profile of the original Dublin Core Metadata Element Set (DCMES) of fifteen descriptors documented on the Dublin Core Metadata Initiative website. AGLS contains sub-properties for describing more categories of resources and allowing richer description of resources.

Resource Description Framework (RDF) is another standard metadata model for data interchange on the web, which provides interoperability between applications that exchange machine-understandable information on the Web. RDF emphasises facilities to enable automated processing of web resources [44-46].

### 2.2. Semantic Space Models

Natural language processing (NLP) [47-51] utilises computer systems to analyse, understand and generate language such as that which human beings naturally use every day. It is an important field of artificial intelligence (AI) in computer science. Common NLP problems include natural language understanding, Part-of-Speech Tagging (POST, e.g. GATE framework), speech segmentation and word-sense disambiguation.

To automatically and quantitatively handle NLP problems, semantic space models build mathematical models representing the context of the texts. There are basically two types of semantic space models: word frequency and Hyperspace Analogue to Language. A few well-known semantic space models are derived from these two basic models, such as LSA and COALS.

#### 2.2.1. Word Frequency (Term-Document Models)

Frequency of words is a foundational mathematical model of text which reflects the occurrence of words in a context. In this model, a number is correspondent to a unique word and represents the number of times that the word occurs in the text corpus. Thus the document can be represented with a vector. Analogically, a group of corpuses can be represented with a matrix where each row is correspondent to a corpus. This is a common model widely used in search engines (e.g. Google combines word occurrence with other information including position, font and capitalisation [52]) and other applications such as text processing, text classification, etc. [53-54].

There are also some variations of the word frequency model with improved feature processing mechanisms.

#### TF-IDF

TF-IDF statistics have been used to weight the values in semantic space models based on term-document statistics [55]. TF (term frequency) describes how often a term/keyword occurs in a document and IDF (inverse document frequency) is the measurement of the unevenness of the term distribution in the corpus, which is the specificity of a term to a document. In other words, IDF measures the number of documents in which a term/keyword occurs in the training dataset – the greater the

number of documents it occurs, the smaller the value of IDF. The higher the TF, the higher the importance (weight) for the document, whereas the more the term is distributed evenly, the less it is specific to a document [56]. There are a few common forms of TF and IDF:

$$TF = freq(t, D) \quad [a]$$

$$TF = \log freq(t, D) \quad [b]$$

$$TF = \log freq(t, D) + 1 \quad [c]$$

$$TF = freq(t, D) / \max(freq(D)) \quad [d]$$

$$IDF = \begin{cases} 0, & w_r = 0 \\ \log_{10} \frac{R}{w_r}, & w_r > 0 \end{cases} \quad [e]$$

$$IDF = \begin{cases} 0, & w_r = 0 \\ \frac{\log_{10} \frac{R}{w_r}}{R+1}, & w_r > 0 \end{cases} \quad [f]$$

where  $R$  is the total documents in the corpus,  $w_r$  is the total number of documents in which the word occurs. In [55] the authors used forms  $d$  and  $e$  to build the model to analyse the Reuters21578 dataset; in [57] the authors used forms  $a$  and  $f$  to model the TREC-6 and TREC-7 datasets.

### The LSA model

Latent Semantic Analysis (LSA) [58] is a word frequency-like semantic space model which is based not on an undifferentiated corpus of text but on a collection of discrete documents [27]. LSA starts with building a matrix in which the rows consist of the words that occur in at least two documents and the columns represent the documents. The cells (or features) indicate the number of times the corresponding word (row) occurs in the corresponding document (column); hence 0 means no occurrence in that document. The next step, the most important step of LSA, is applying SVD (see Section 2.1) on the matrix. It is believed that this SVD step reveals the inherent relationship between two words or two documents [58-61].

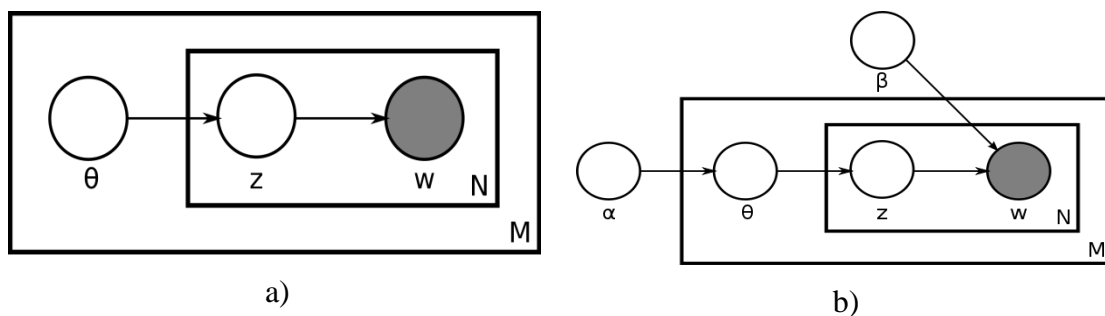
LSA has numerous existing applications for information retrieval, automatic essay grading and synonym testing [62]. In the bioinformatics domain, LSA has been used

to map MEDLINE abstracts to the Gene Ontology [63], and to extract semantic content from the UMLS [64]. LSA has also shown promising performance for diagnosing thought disorder in schizophrenia [65].

### Probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA)

Probabilistic Latent Semantic Analysis (pLSA) was introduced in 1999 by Jan Puzicha and Thomas Hofmann [66]. pLSA is a probabilistic variation on the classic LSA model. Instead of using term-document co-occurrences, pLSA models probability of each co-occurrence by assigning a latent class  $c$  conditionally to each document  $d$  according to conditional probability  $P(c|d)$ , and a word  $w$  is then generated from that class according to conditional probability  $P(w|c)$ . pLSA can model the co-occurrence of any pair of discrete variables in exactly the same way.

Latent Dirichlet allocation (LDA) [67] is an extended pLSA model with a Dirichlet prior [68] on the per-document topic distribution (Dirichlet is a family of continuous multivariate probability distributions parametrized by two vectors  $\alpha$  and  $\beta$  of positive reals). Similar to pLSA, each document may be viewed as a mixture of various topics, while in LDA the topic distribution is assumed to have a Dirichlet prior. In other words, the pLSA model is a special LDA model under a uniform Dirichlet prior distribution. Figure 10 illustrates pLSA and LDA.



**Figure 10** Plate notation diagrams: a) Illustration of pLSA; b) illustration of LDA; where positive-real vector  $\alpha$  is the parameter of the uniform Dirichlet prior on the per-document topic distributions,  $\beta$  is the parameter of the uniform Dirichlet prior on the per-topic word distribution,  $\theta$  is the topic distribution for a document,  $z$  is a topic for a word in a document,  $w$  is a specific word,  $N$  is the number of words in a document and  $M$  is the number of documents. (From [67])

pLSA, LDA and their extended models have been used for information retrieval [69] and filtering [70-71], natural language processing [72-73] and machine learning [67, 74] from text.

### 2.2.2. Term-Term models (sliding-window)

#### Hyperspace Analogue to Language

Instead of tracing the occurrence of the keywords to model a text corpus to solve the semantic space issues, Burgess et al. attempted in the 1990s to reveal the relationship between words in a corpus from their co-occurrence [18-20, 75], Burgess and Lund examined whether HAL could represent abstract concepts, such as love, hate and joy. They found that, in a comparison with human raters in predicting abstract variables for a set of words, “global co-occurrence information carried in the word vectors can be used to predict a tangible proportion of the human likert scale ratings.” They built a semantic space model called Hyperspace Analogue to Language (HAL) by constructing a high dimensional matrix in which each row/column of a vector represents a word, and the vector for that word contains the information of the co-occurrence of that word and the other word in the corpus. In the case of HAL, an  $N \times N$  matrix is instantiated with an  $N$ -length vector for each unique word occurring in a corpus. A ‘window’ several words in width is moved across the corpus. Wherever two words occur within the window the value at their intersection in the matrix is incremented, the immediate adjacent word(s) gains the highest value (HAL value), and the further from that word, and the smaller the HAL value, down to 0. Thus, a corpus is converted to a high-dimensional semantic space with minimal consideration to grammar. To illustrate the construction of the HAL matrix, we use the text in Figure 11 as an example. Table 2 is the HAL matrix for the sample text in Figure 11 for a window size of 3.



## Literature Review

*“Evaluating breast changes or masses usually starts with a mammogram or sonogram (ultrasound) performed by a radiologist.”*

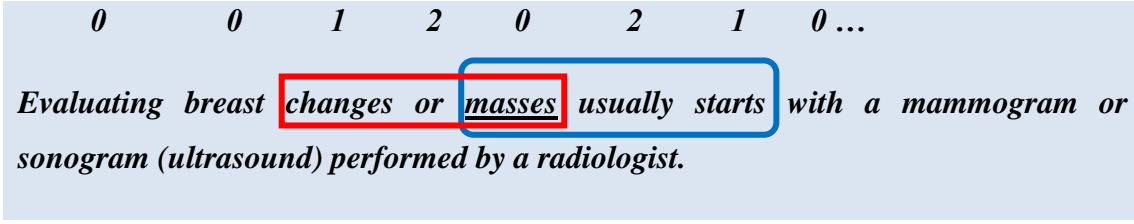


Figure 11 HAL values for word ‘masses’ with window size 3

Table 2 HAL matrix for the sample text for window size of 3

	evaluating	breast	changes	or	masses	usually	starts	with	a	mammogram	sonogram	(	ultrasound	)	performed	by	radiologist	.	evaluating	breast	changes	or	masses	usually	starts	with	a	mammogram	sonogram	(	ultrasound	)	performed	by	radiologist
evaluating	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
breast	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
changes	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	
or	0	1	2	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	2	1	0	0	0	0	
masses	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	
usually	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	
starts	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	
with	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	
a	0	0	0	0	0	0	1	2	0	0	0	0	0	0	1	2	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	2	1
mammog	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	
sonogram	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	
(	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	
ultrasoun	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	
)	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	
performe	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	
by	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0	
radiologis	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

At an algorithmic level, HAL requires going through the corpus word by word, and for each word assigning a value to other words in its neighbourhood (aka, the

‘window,’ the size of which is a significant parameter, typically 10 in practice). All words within the window are considered as co-occurring with each other with strengths inversely proportional to the distance between them. As the window is moved over the entire text corpus, scores accumulate into an  $N$  by  $2N$  matrix (the ‘HAL matrix’,  $H$ ), where  $N$  is the number of distinct words in the corpus. This can be accumulated across the text of multiple websites (or multiple articles for the Reuters data); In Figure 11 the window and its HAL score contribution is illustrated at the point where the window is applied to the word ‘masses’.

### Refined HAL

We can see that in the original concept of HAL punctuation, sentence and paragraph boundaries, the order of co-occurrence of words (i.e., before or after), and some extremely common words (e.g., ‘a’, ‘the’, ‘is’, ‘are’, etc. – the ‘stop words’) are equally treated. A problem arose in that the high frequency, or high variance, columns contribute disproportionately to the distance measure, relative to the amount of information they convey [27]. In the case of the example in Figure 11, the words ‘a’ and ‘or’ have the greatest sum of HAL value and hence will have a large effect on their inter-vector distance; but it is counterintuitive that such words are the most important aspects of the text.

In keeping with prior studies [59], punctuation, sentence and paragraph boundaries, the order of co-occurrence of words, and some extremely common words are not considered useful to the inference of the underlying semantic space, and are discarded. Table 3 shows the refined HAL matrix for window size of 3.

# Literature Review

Table 3 Refined HAL matrix for window size of 3

	evaluating	Breast	changes	masses	usually	starts	mammogram	sonogram	ultrasound	performed	radiologist
evaluating	0	2	1	0	0	0	0	0	0	0	0
breast	2	0	2	1	0	0	0	0	0	0	0
changes	1	2	0	2	1	0	0	0	0	0	0
masses	0	1	2	0	2	1	0	0	0	0	0
usually	0	0	1	2	0	2	1	0	0	0	0
starts	0	0	0	1	2	0	2	1	0	0	0
mammogr	0	0	0	0	1	2	0	2	1	0	0
sonogram	0	0	0	0	0	1	2	0	2	1	0
ultrasound	0	0	0	0	0	0	1	2	0	2	1
performed	0	0	0	0	0	0	0	1	2	0	2
radiologist	0	0	0	0	0	0	0	0	1	2	0

## The COALS and COALS-SVD model

The COALS model [27, 76] is another strategy that attempts to manage the problem of HAL that high-frequency words do not carry proportionate information. In COALS, instead of using the HAL values to form the matrix, the correlation values based on the HAL values are used. Details of the process of building a COALS model are:

- Building a HAL matrix using a smaller window size, typically 4, stop words and punctuations included;
- Picking the first  $m$  columns reflecting the most common words;
- Converting the HAL values to word pair correlations;
- Setting the negative values to 0 and taking the square root of the positive values.

COALS, like other HAL approaches, also ignores the ‘information of order’ (after or before are treated equally).

For COALS-SVD [27], a matrix is obtained by applying SVD on the COALS matrix.

### 2.3. Text Classification

#### 2.3.1. Classification Methods

##### **ID3**

ID3 is an algorithm for building Decision Trees developed by Ross Quinlan in the 1970s [77]. It commenced with finding an attribute or feature that can partition (using some sort of distance measurement, see Section 2.1) the training dataset into two or more subsets/classes/groups with the greatest entropy loss or information gain (see Section 2.1). That first best attribute is used for the root of the decision tree. Using this method, we can then recursively split the subsets further to generate more nodes and thus form the decision tree. This is applied repeatedly until no more partitioning can be done on any subsets, either because we have reached a leaf node where the classes are completely separated, or because no feature is available that can further differentiate the classes.

##### **Decision Forest**

A well-trained decision tree can give excellent performances on a certain dataset, but monotonicity and overfitting will normally limit its commonality and further improvement of its accuracy. However, using a set of several decision trees (called ‘decision forest’) instead of one can splendidly overcome such problems. A decision forest consists of a set of decision trees sharing the conjunct dataset. There are two sorts of patterns whereby the trees in a forest share a dataset: case splitting [78-79] and feature splitting [80]; case splitting randomly selects a number of cases from the dataset with or without repeats to build the trees, while features splitting splits the features in the dataset instead of cases.

The final output of a decision forest is normally ‘voted’ within the results of those trees inside it.

### K-Nearest Neighbour (KNN)

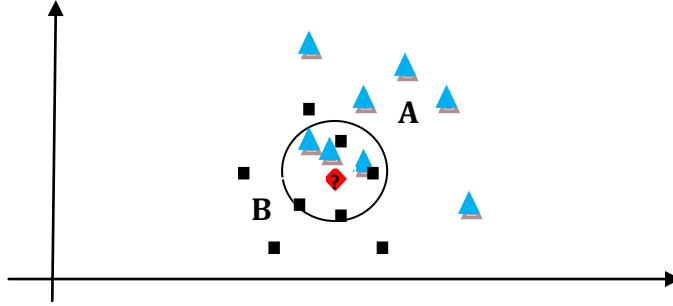


Figure 12 Illustration of KNN

The K-nearest Neighbour (KNN) algorithm is one of the most fundamental classification algorithms. As shown in Figure 12, to determine the ascription of a test case (the red diamond), with  $K = 7$  (normally an odd number is used), we select the 7 closest neighbours using one of the distance measure methods (see Section 2.1). In this case, 3 of the 7 closest cases belong to class A and the remaining 4 belong to class B, so the test case belongs to class B. If we pick  $K = 3$ , then the result will be the opposite of that for  $K = 7$ .

Since the introduction of KNN a number of refinements have been proposed, including [81-84]. One of the most recent, AKLH [85] is described further below.

### Adaptive K-Local Hyperplane

Adaptive K-Local Hyperplane (AKLH) method [85] is a recent extension of the manifold-related nearest neighbour classifiers which have been proposed to improve the K-nearest neighbour classifier. All these methods try to approximate the local data cloud by some kind of low-dimensional manifolds. In particular, it is both the improvement and the extension of the K-local hyperplane distance nearest neighbour (HKNN) method proposed by Vincent and Bengio [83]. The basic idea of the HKNN and AKLH methods is to approximate the potentially unseen instances in the manifold of each class by a local hyperplane. Specifically,  $K$  nearest neighbours of a query  $q$  are first selected as the class prototypes by a KNN method; a local hyperplane is then constructed to approximate the local manifold of each class based on these class prototypes. Hence, the class label of the query is assigned according to the distance between the query and the local hyperplane, HP, of each class.

The AKLH method resolves some basic shortcomings of the HKNN method, most notably that HKNN works well only for small values of  $K$ ; it suffers from bias in high dimensions, and implicitly assumes all features are equally relevant for classification, which may yield an unsatisfactory performance for datasets with complex structure. AKLH resolves these problems by considering the feature weight. The prototypes are selected by the adaptive nearest neighbour method (using the weighted Euclidean distance metric), while the feature weights are determined based on the combination of the ratio of the between-group to within-group sums of squares (RBWSS) criterion and the criterion used in the Relief-F algorithm [86]. Figure 13 illustrates the AKLH method.

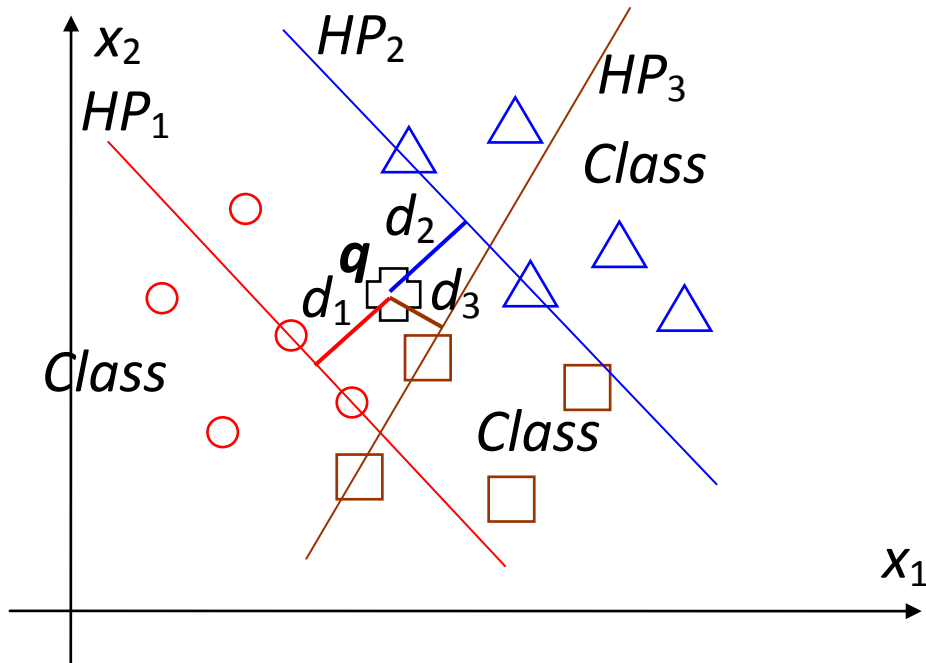


Figure 13 Graphical presentation of the AKLH method

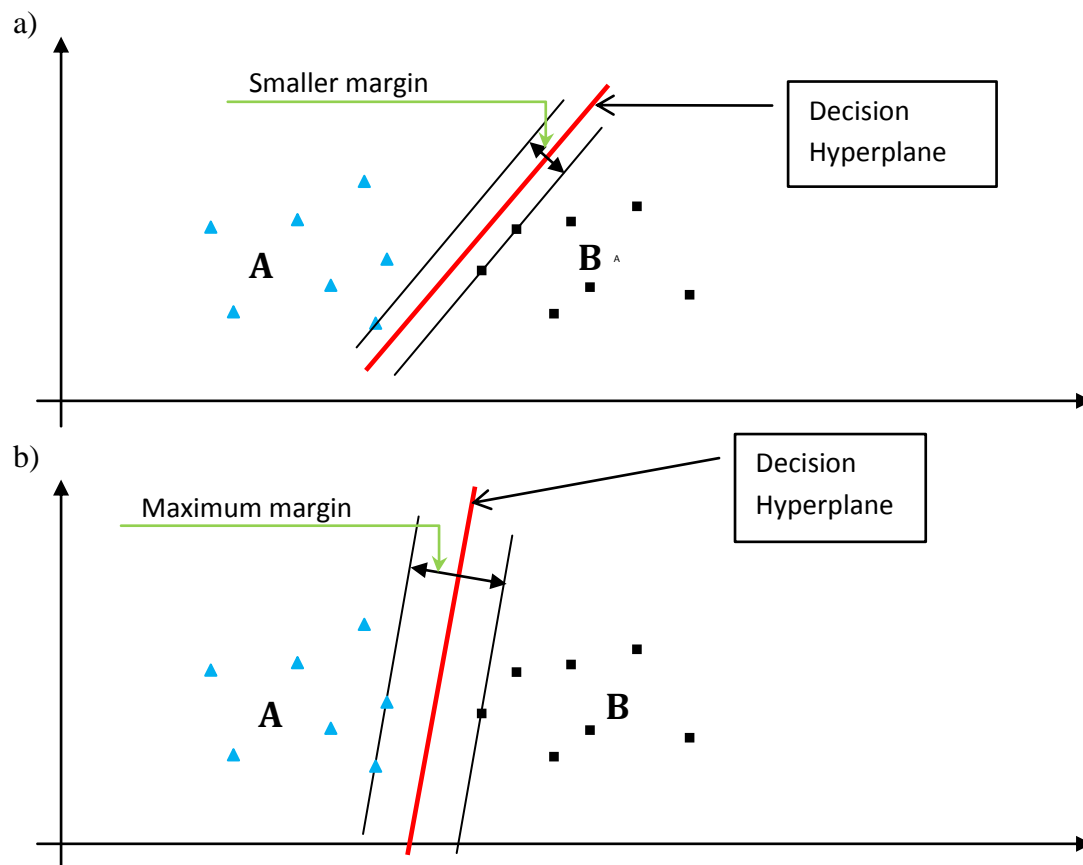
### Support Vector Machine

The Support Vector Machine (SVM) [87-90] is a supervised classification or regression method for machine learning and pattern classification which has been demonstrated to be a high-performance algorithm for classification on most popular benchmark datasets [55, 91-93]. The basic idea of SVM is mapping input vectors into

## Literature Review

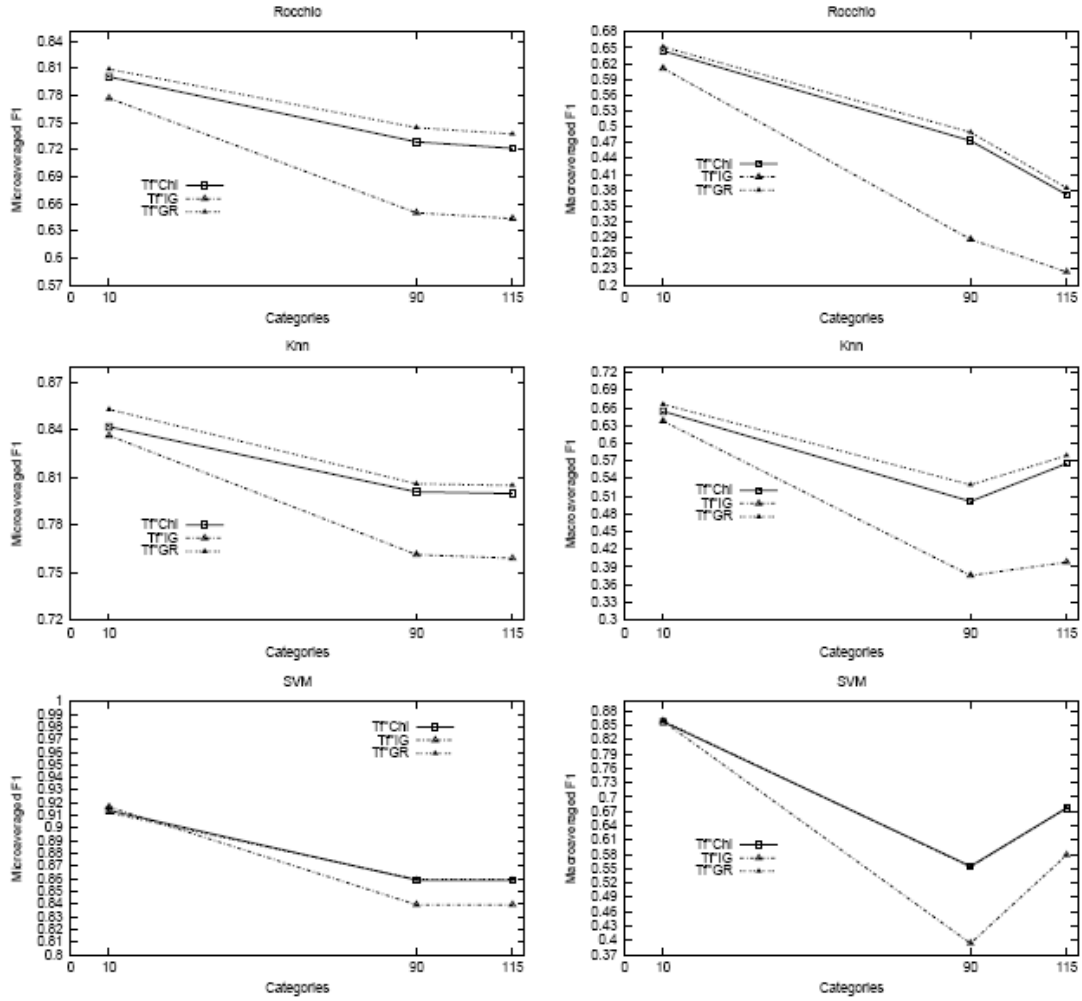
a high dimensional feature space, where a maximal margin hyperplane is constructed (see Figure 14). It was shown that when training data are separable, the error rate for SVMs can be characterized by  $h=R^2/M^2$ , where  $R$  is the radius of the smallest sphere containing the training data and  $M$  is the margin (the distance between the hyperplane and the closest training vector in feature space). This function estimates the VC dimension of hyperplanes separating data with a given margin  $M$ .

Note that Figure 14 shows the linear classification case; in practice, most cases are non-linear, and polynomial SVM normally has fewer test errors [94-97].



**Figure 14 Illustration of linear SVM. In a) and b) the distance between the hyperplane and the nearest data point on each side is called 'margin'. The margin in a) is smaller than that in b). The hyperplane with the maximum margin forms the SVM classifier.**

SVM is widely used for text classification [92, 98], and in bioinformatics [99-101]. SVM supports both regression and classification tasks [89, 91].



**Figure 15** Plots of micro-averaged F1 (leftmost) and macro-averaged F1 (rightmost) obtained with supervised weighting and a  $\xi = 0.0$  reduction factor [102]. Plots indicate results obtained with Rocchio (top), K-NN (middle) and SVMs (bottom). The X axis indicates the three subsets of Reuters-21578 [55].

Figure 15 (extracted from [55]) shows the relative dominance of SVM on Reuters21578 dataset (see next section). The micro-averaged F1-measures are plotted on the left side for three algorithms: Rocchio [103-104], KNN and SVM, with SVM consistently showing the best performance on micro-averaged F1 and in most cases showing the best macro-averaged F1 performance.

## 2.3.2. Previous Examples of Text Classification

Text Classification (TC) is also named ‘Text Categorisation’ [105], which has long been an important field of AI. The task of TC is to predict the category of an unknown text sample (i.e., a ‘document’) based on knowledge of the categorisation of a corpus of previously classified text samples. The application of TC is very extensive, including information retrieval, text routing, text filtering and text



understanding systems [105-107]. In the Internet age a range of new applications of TC are craving better solutions that have not as yet turned up, such as spam filtering, portal building and advanced searching (or categorised searching) [108-110]. I review some recent and representative projects here for comparison.

### **Reuters21578**

Reuters21578 is a well-known public open test collection for TC research [111], constituting a well-established benchmark for TC algorithms. Reuters21578 consists of 21,578 news stories appearing in the Reuters newswire in 1987, which are manually collected and classified according to five groups of categories: Exchanges, Orgs, People, Places, and Topics. The ‘Topics’ are economic subject categories, which is the only group that has actually been used in TC experimental research. There are 135 categories in this group, among which 20 have (in the ModApte split) no positive training documents; as a consequence, these categories have never been considered in any TC experiment, since the TC methodology requires deriving a classifier either by automatically training an inductive method on the training set only, and/or by human knowledge engineering based on the analysis of the training set only. Since the 115 remaining categories have at least one positive training example each, they can in principle all be used in experiments. However, several researchers have preferred to carry out their experiments on different subsets of categories. Globally, the three subsets that have been most popular are

- The set of the 10 categories with the highest number of positive training examples - R(10).
- The set of 90 categories with at least one positive training example and one test example - R(90). This appears to be the most frequently chosen subset;
- The set of 115 categories with at least one positive training example - R(115).

One problem with the previous version of the Reuters collection was that the ambiguity of formatting and annotation led different researchers to use different training/test divisions. To eliminate these ambiguities, Reuters21578 collection specifies exactly which articles are in each of the recommended training sets and test sets by precisely denoting the values those articles will have on the TOPICS, LEWISSPLIT, and CGISPLIT attributes of the REUTERS tags. Three splits are

provided: Modified Lewis ('ModLewis') split, Modified Apte ('ModApte') split, Modified Hayes ('ModHayes') split. The ModApte split is the most commonly used.

### IBM KitCat System

IBM researchers have built a text classifier based on a decision tree and word frequency [112], which aims to be a faster classifier capable of processing large corpuses by taking advantage of sparse datasets. The most special idea in this algorithm is the splitting criteria of the training dataset for the construction of their tree: selecting a feature  $f$  and a value  $v$  (typically, 1, 2 or 3) that can split the dataset into two subsets with the greatest information gain. This partition corresponds to a decision rule based on whether or not the word corresponding to feature  $f$  occurs more than  $v$  times in a certain document.

In this algorithm, they believe that a high frequency word is hardly a more informative predictor than a less frequent word, so to speed up the algorithm, they truncate the word count to be at most a value  $V$  (they choose 3).

Finally, they convert the tree into an equivalent set of symbolic rules, such as:

$$Share > 3 \ \& \ year \leq 1 \ \& \ acquire > 2 \rightarrow acq$$

That means “if the word *share* occurs more than three times in the document and the word *year* occurs at most one time in the document and the word *acquire* occurs more than twice in the document, then classify the document in the category *acq*.”

The results of experiments with KitCat are as following:

- For Reuters21578, with Mod-Apte splits, 9603 training items and 3299 testing items in 93 categories, the micro-averaged precision is 87%, recall is 80.5%, average is 83.8%.
- For 7000 web pages in corporate web sites in 42 categories, they obtained an accuracy of 86%.
- KitCat was also applied to 4913 emails sent to a US bank in 9 categories; with 3943 for training and 970 for testing, the micro-average precision is 92.8% and recall is 89.1%.

- KitCat is good at ‘dirty’ data. A 100% precision was obtained on four categories in a customer data set, with 84 test instances out of 8895 documents.

### **HAL and emotion and parts of speech**

In 1997, Burgess and Lund presented a model of high-dimensional context space, the Hyperspace Analogue to Language (HAL). They demonstrated the ability of this model of categorising the terms according to their semantic or grammatical environment, as well as representing the emotional connotation [19-20]. Their method entails first building the HAL matrix, then, based on that, creating a corresponding Euclidian distance matrix, and finally using Multidimensional Scaling (MDS) to convert it to a low-dimensional (especially 2D) matrix and plotting it. Visually they successfully separated the different noun classes and different emotional classes.

## **2.4. Consumer Health Information and Tailoring**

As mentioned in the Introduction, people are increasingly using the Internet to find health information, and there are various methods to help them locate sound information for their particular case.

In this section we review BCKO as an example of the portal approach to support consumers in finding good health information. We also review Recommender and Tailoring systems as two alternative technologies to match up consumers with appropriate content, and finally look at two systems (CHESS and Violet technology) that use a combination of these technologies. These three technologies are compared in Table 4.

## Literature Review

**Table 4 Comparison of three technologies for matching consumers to content.**

<b>Technology</b>	Portal	Recommender System	Computer Tailoring
<b>Description</b>	Human 'curator' selects links	System chooses links	System builds text

### **2.4.1. Breast Cancer Knowledge Online**

Breast Cancer Knowledge Online (BCKO) is a gateway to breast cancer information (see <http://www.bckonline.monash.edu.au/index.jsp>).

The resource selection process for BCKO was carried out by 'domain experts', that is, women with firsthand experience of breast cancer and extensive knowledge of the medical, supportive and psychosocial information needs of the breast cancer community. In addition, a study investigating the specific information needs of women with breast cancer and their families was carried out by the research team. The subsequent selection of materials was based on the data provided by women across all disease stages and age groups. The individuals responsible for the final selection of resources were: Sue Lockwood, Chair of the Breast Cancer Action Group (Vic. [Victoria, Australia]) and Rosetta Manaszewicz, Steering Committee member, the Breast Cancer Action Group (Vic.).

Stringent 'selection criteria' were developed with the objective of providing users with information that was credible, current, and represented the diversity of views, format presentation and type of information which women indicated they required in the user-needs study. Each resource was therefore selected according to criteria developed for that particular resource category. Whilst it was not mandatory that each resource meet all criteria within its respective category of 'medical', 'supportive' or 'personal,' all resources were deemed to meet a majority of the criteria designated for each category.

BCKO [113-114] attempts to address the issue of credibility and authoritativeness by supplying the user with a narrative 'quality report' for each resource contained within the portal. The overall objective is

## Literature Review

---

- to inform the user of certain characteristics which may or may not be present in the individual resource.
- to concentrate on those features which international 'best practice' agrees constitute evidence of information 'quality'.
- to allow the user to prioritise and make the ultimate decision as to which 'quality' features are important. This will often depend on the kind of information being sought.

### 2.4.2. Recommender Systems

A Recommender System [115-120] is an intelligent computer application that advises people about products, information or services in which they might be interested, aiming to reduce information overload and retain customers by selecting a subset of items from a universal set based on user preferences. There are three broad types of recommender systems: Collaborative Filtering Recommender Systems [120], Content-based Recommender Systems [121] and Knowledge-based Recommender Systems [122]. Collaborative filtering systems give recommendation based on customers' experience, habits or rating other than the current user; Content-based systems advise users based on their users' own profile, such as user preferences and their interaction history with the recommender system; while the knowledge-based system does this via an internal database of knowledge (i.e. set of rules), arriving at a recommendation by asking questions in a step-by-step fashion.

The advantage of the collaborative filtering method is that more data is collected, and with this the greater probability that the user will get a good recommendation (i.e. best match); while the drawback is that its performance is dependent on the availability of relevant data (e.g. if your interest is not close to that of others, then the system will not be much help).

The advantage of the content-based method is that it works without needing to access a huge database, and hence is reasonably fast. The problems are that its performance depends on an efficient classifier and having sufficient data about the user.

The advantage of the Knowledge-based method is that it works more accurately than the other two without collecting any data from the customer. The drawback is that it is

slower than the others, and since it needs the user's complete participation, is not suitable for some circumstances that require full automation.

Some examples of recommender systems include:

- Yenta: a multi-agent, referral-based matchmaking system [123]: Yenta is a content-based recommender system. Currently the Yenta-Lite system works by collecting user's emails, newsgroup articles and user files that are received, read or written by the user. Each of them is treated as a *document* – called a *grain*. A user is considered to have an interest if a few *grains* are similar to others, in which case the system will group those people with common interests: it is called Matchmaker. Yenta-Lite has been simulated for 1000 Yentas, and shown good performance and convergence.
- A smart e-learning recommender system [124]: this is a collaborative filtering recommender system. It works by maintaining a repository of a number of papers including magazine papers, conference papers and workshop papers, and organising and tagging these by features such as length, author, etc. Users are clustered into groups by their learning interests. Recommendations are then generated from the cluster of users and the repository of papers. This system was still under construction at the time of publication.
- A multi-agent TV recommender [125]: this is another content-based recommender system. It tracks a user's **implicit** viewing history, **explicit** preferences and **feedback** information. With implicit information it calculates the likelihood of the program and recommendation scores, combining these with the recommendation generated via the explicit information. The final recommendation will be fine-tuned according to the feedback. The explicit recommender was found in experimentation to be better than the implicit recommender.

### 2.4.3. Computer tailoring

Computer Tailoring [126-129], widely used in healthcare to generate information that is specifically adapted to one person or patient, is a kind of intelligent interactive

## Literature Review

---

computer technology. As concluded by De Vries et al. [126], computer tailoring requires at least four components: 1) a profile of the patient, or a diagnosis at the individual level of characteristics relevant to a person's health behaviour or illness; 2) a message library containing all the health education messages that might be needed; 3) an algorithm, or a set of decision rules that evaluates the diagnosis and generates the appropriate messages; and, 4) a channel or medium to deliver the message to the intended user, such as a letter. The advantages of using a tailored intervention can be summarised as follows:

- Extraneous information has been removed;
- The remaining information is highly individualised and more personally relevant to the recipient.
- Patients place emphasis on information relevant to themselves only;
- Information that is attended to is more likely to have an effect than that which is not;
- When attended to, information that addresses the unique needs of a person will be useful in helping them become and stay motivated, acquire new skills, and enact and sustain desired lifestyle changes.

Examples of computer tailoring systems include:

- A tailored multimedia nutrition education pilot program for low-income women receiving food assistance [130]. This is an innovative and promising tool to motivate people to make healthy dietary changes. This program has been tested among 378 low-income women, where they were randomly split into intervention and control groups. A baseline survey and a one to three months post-intervention survey were taken to collect the users' profiles, including dietary fat intake, stage of change, knowledge of low-fat foods, self-efficacy and eating behaviour. The program, based on an in-built nutrition knowledgebase as well as a user feedback knowledgebase, gives those participants suggestions. As a result, both groups showed improvement on knowledge, stage of change, certain eating behaviours and fat intake, without significant difference between the two groups.

- Web-based computer-tailored smoking cessation program [131]. This is a trial web based system in England and the Republic of Ireland. A total of 3971 subjects purchased a particular brand of nicotine patch and logged on to use a free web-based behavioural support. Tailored and non-tailored behavioural smoking cessation materials are available online for the participants to read. The result is continuous abstinence rates at 6 weeks for tailored vs. non-tailored condition is 29.0% vs. 13.9%, 12 weeks is 22.8% vs. 18.1%, and the satisfaction with the program is much higher in the tailored than in the non-tailored condition.
- A website-delivered computer-tailored intervention for increasing physical activity in the general population [132]. The participants are 434 parents and staff from schools in Belgium in the spring of 2005, who were divided into three groups: intervention groups receiving intervention with or without repeated feedback and a control group. Physical activity levels were self-reported at baseline and at 6 months (n = 285). Significant increases were found for active transportation (+20, +24, +11 min/week respectively) and leisure-time physical activity (+26, +19, -4 min/week respectively); and a significant decrease for minutes sitting on weekdays (-22, -34, +4 min/day respectively).

#### **2.4.4. Comprehensive Health Enhancement Support System (CHESS)**

CHESS is an interactive computer system containing information, social support and problem-solving tools developed by the University of Wisconsin-Madison [133]. CHESS modules have been developed for breast cancer, AIDS/HIV Infection, substance abuse, sexual assault and academic crisis. A practical online model of CHESS is maintained by Hartford Hospital that provides help to individuals coping with a diagnosis of breast cancer or prostate cancer (see <http://www.hartfordhospital.com/cancer/PatientResources/CHESS/default.aspx>).

CHESS is designed to provide timely, easily accessible resources (information, social support, decision-making and problem-solving tools) when needed most. CHESS consists of three groups of components:



- Information components include Questions and Answers, Instant Library, Ask an Expert and Getting Help/Support.
- Social support components include Discussion Groups and Personal Stories .
- Problem solving components include Decisions & Conflicts and Action Plan.

After the pilot study, CHESS was reported to be well accepted, and easy to use and understand.

### **2.4.5. Violet Technology (VT)**

Violet Technology was developed by Ma et al. [134] in 2005, and consists of four parts and a web portal employing these four parts: a comprehensive Diabetes Information Profile (DIP); information tailoring and prioritisation algorithms (supporting an Information Service); quiz tailoring and prioritisation algorithms (supporting a Quizzing Service); and agenda personalisation algorithms (supporting formulation of patient question sets as an Agenda Service).

The DIP preserves the user's diabetes information. The DIP contains diabetes-related situations (e.g. lifestyle), browsing history, information preferences, quizzing history and history of agenda generation.

The Information Service operates in two steps, Filtering and Prioritisation. Filtering removes irrelevant information and gives the result to Prioritisation, which assigns a priority (i.e. weight) to an information item according to three groups of rules: significant data-oriented; patient's knowledge level oriented; and patient's information preference oriented.

The Quizzing Service also uses filtering and prioritisation. The quiz filtering rules are matched against the patient's DIP. The prioritisation mechanisms of the quizzing service use three mapping functions: significant data mapping; patient's educational exposure mapping; and mapping the patient's response to quiz questions.

The Agenda Service preserves four sources from the patient-specific agenda question pool: (1) the patient's greatest difficulties in diabetes management; (2) the patient's diabetes-related issues; (3) the information items that the patient added during information browsing; and (4) the questions that the patient created for themselves.

## Literature Review

---

A small range of testing showed that the VT system is easy to learn and use, and that the tailoring and quiz services work well.

## Chapter 3. Methods

Since Burgess and Lund [19] had been more interested in linguistics than classification per se, we found that we had a wide space of parameters and methods to consider in adapting HAL as a practical classifier for our problem. Herein we present what we believe are some of the more fruitful and illustrative approaches. In this section we describe our data sources, the specific issues of deriving and managing the data as a HAL model, each of the classification algorithms we apply, and finally the protocol by which we assess their performance as reported in the Results. In each case, but particularly with respect to the classification algorithms, we provide an overview of the pathway and rationale by which the specific methods reported herein were chosen.

### 3.1. Data sources

In this project, three data sources have been used: ‘Transition’ dataset, BCKO dataset and Reuters21578. I started from Transition, and then shifted to BCKO; after achieving excellent accuracies on BCKO, I selected the Reuters21578 public dataset for comparison.

#### 3.1.1. Transition

My PhD research started with experimentation on the Transition dataset [2]. This is a process of convoluted passage during which people redefine their sense of self and redevelop their self agency in response to life events such as chronic illness [1, 135-136]. The Transition dataset consists of emails written by people with chronic illness in an Internet virtual community organised by Dr. Debbie Kralik [137-138], Head of Research for the South Australian Royal District Nursing Service (RDNS). The Transition dataset covers a log of the emails from 20 women with chronic illness creating entries from 2003 to 2005.

Our corpus for analysis is based on the long-term conversations between 20 women who self-describe as struggling to cope with chronic illness. These women participated in an electronic discussion group implemented as a limited-access major domo email list service, facilitated by an expert in transition (Dr. Kralik) who

moderated the conversation and could contact the participants outside the list if needs arose. Previous research [59] had analysed data from this list to examine the quantitative changes in language use on dimensions of ‘kin’ and ‘negative emotion’ (which were expected to roughly correspond with the key transition concepts of ‘ordinariness’ and ‘extraordinariness’). The HAL vectors for ‘sense of self’ (the sum of HAL vectors for words ‘me’ ‘my’ ‘I’ and ‘myself’) for two participants were used for my investigations. Seventeen months of data from the corpus were used.

### 3.1.2. BCKO

The major distinction in BCKO is the resource *type* attribute, which identifies site content as any of ‘medical’ (evidence-based), ‘supportive’ (regarding support resources), and ‘personal’ (individual views). While it may be possible to contrive an ad hoc set of heuristics for distinguishing these classes of sites, we have focused on examining the language use as the basis for automated classification. Classically, a word frequency vector provides a set of features for classification, one feature for every distinct word used in any of the webpages under consideration. Depending on the details of the data pre-processing, such a method may yield some 5000 features for classification.

A key finding of the initial BCKO user needs analysis was the need to identify quality resources that dealt not only with medical and scientific issues relating to breast cancer, but also with its psychosocial impacts. The metadata schema developed to describe the resources therefore incorporates an encoding scheme for categorising the type of resource as medical, supportive and/or personal perspective (an indication of the tone of the article). The BCKO portal database provides metadata to support personalised search for approximately 1000 consumer health websites. To provide an initial test for HAL-based prediction of the BCKO type, I examined the problem of matching the classification of types ‘medical’ or ‘supportive’ to that given in the portal’s database (discarding the ‘personal’ type for the present study because that particular type code was utilised infrequently). To train the classifier, a corpus was extracted from the text of the webpages indexed by the portal.

We believe, in Semantic Space, the determinant of the character of a webpage is its textual content, whereas the content near the edge and corner is often irrelevant to the

main content, thus leading to an incorrect classification. Therefore in this project the text is conditioned automatically to remove items outside the main text of the webpage, including sidebars, ads, images and web links.

BCKO indexes 135 sites that the coders have typed as supportive and 701 of type medical. The two types are not mutually exclusive: the 127 sites which are classified as both medical and supportive are omitted in the assessment of algorithms in section 3.3.4 and this issue of non-exclusive class membership is subsequently addressed in section 3.3.7. The sites coded exclusively as supportive are the rarer group, with 51 such sites available for training.

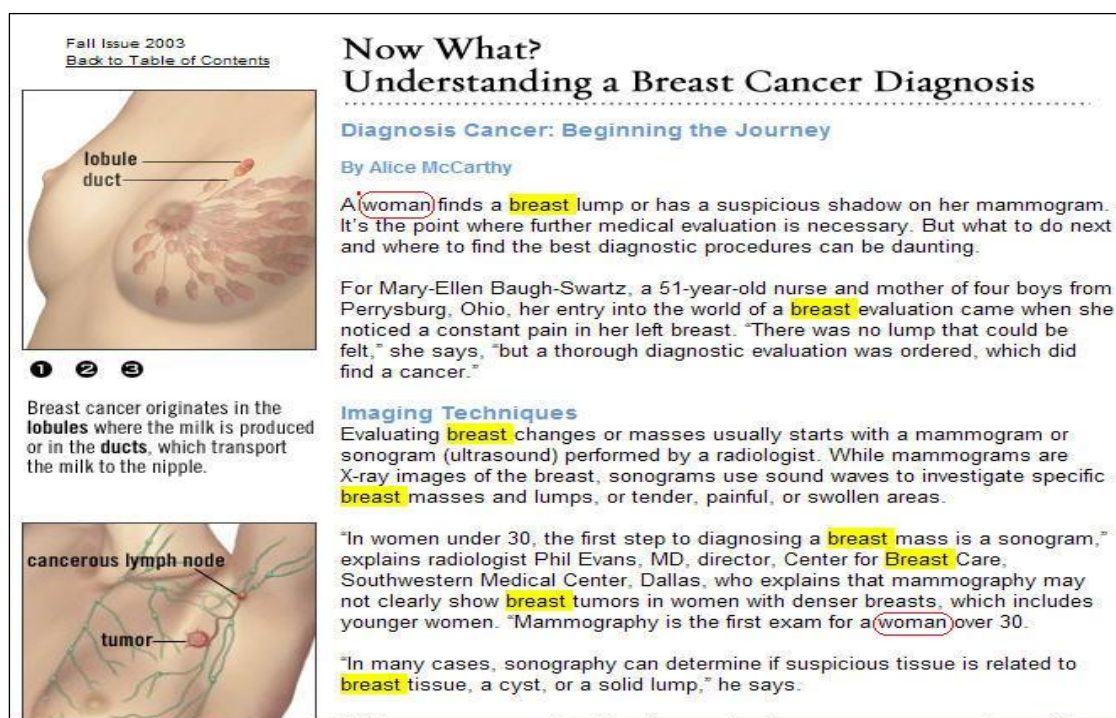


Figure 16 Excerpt from webpage of type 'medical'

Figure 16 is an excerpt from a webpage of type medical; it is basically informative, and largely in the third person. Those highlighted and circled words are key words (possessing high values of HAL and apt to form high-level nodes in decision trees). Figure 17 is an excerpt from a webpage of type supportive. Both the medical and supportive documents use the word 'breast' with roughly equal frequency, but aspects of the context of use differ; in the supportive one, for instance, 'wife' and 'husband' appear within a few words of 'breast'.



## Methods

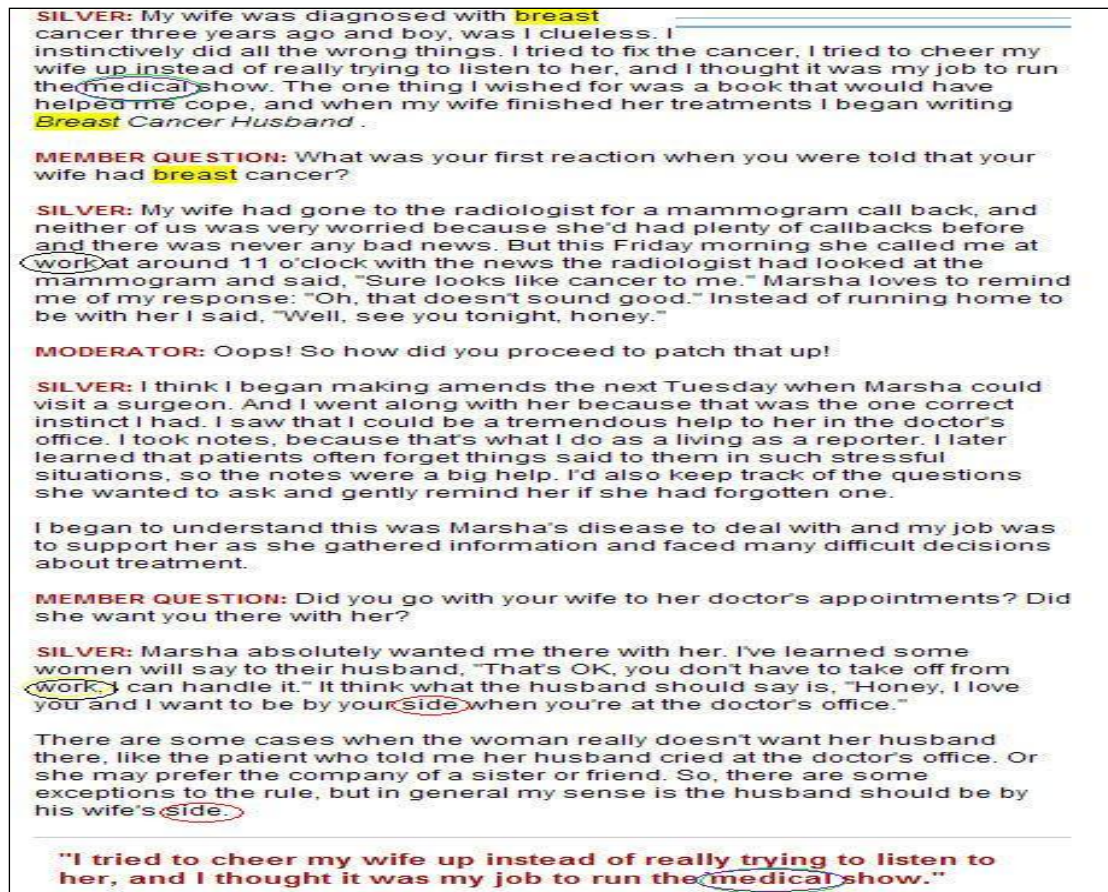


Figure 17 Excerpt from webpage of type 'supportive'

Our experiments also cover the attributes of disease stages and authors' credentials in the BCKO portal. For the disease stages attribute I was able to retrieve 213 valid webpages (ones accessible at the URL indicated by the portal at the time of the study) coded exclusively to 'Early Breast Cancer', 17 coded exclusively as 'Recurrent Breast Cancer' and 138 exclusively 'Advanced Breast Cancer'. Due to the limited 'Recurrent' data, we focus on 'Early' versus 'Late' stage as a classification problem in the remainder of this thesis. For the author credentials attribute there are 9 values, of which a website is assigned only one: Cancer Organisation, Clinician, Commercial Body/Group, Consumer Group, Education Institution, Government Organisation, Lay Author, Medical Organisation and Researcher. Clinician and Lay Author have 319 and 52 valid webpages, respectively, and are utilised for the present study.

Since in the BCKO database, the sense of attribute 'type' is relatively clear, to make it easy to understand, hereafter I describe the algorithms mainly using the *type* attribute,

which consists of three kinds of members (see <http://www.bckonline.monash.edu.au/index.jsp>):

1. Medical: scientific, evidence-based materials, medical and research articles
2. Personal: personal views of consumers, patients and health professionals
3. Supportive: supportive materials which provide details on accessing useful facilities, organisations and support/advocacy groups.

Among these three types of webpages, there are relatively few instances of personal in practice, ‘personal’ is close to supportive and therefore in this research we omit the webpages of type personal.

### **3.1.3. Reuters21578**

To provide a well-known basis for comparison, I employ the Reuters21578 data set, a sequential set of news articles, mostly concerning business and economy, coded into a number of categories. Reuters21578 has become a long-standing benchmark for text classification algorithms and is freely available for experimentation purposes from <http://www.daviddlewis.com/resources/testcollections/~reuters21578/>. For the study of comparison between the classifiers in Section 3.2.3, I examine the two categories with the highest number of training cases, which are ‘earn’ and acq. There are 3735 cases in class ‘earn’ and 2125 cases in class acq.

I also used 10 categories with the highest number of members (R10, see Section 2.3.2) for the experiment of Multi-classes Classification and achieved over 98.3% accuracy using the ModApte split. These 10 categories are ‘earn’ 3735 cases, ‘acq’ 2125 cases, ‘crude’ 355 cases, ‘trade’ 333 cases, ‘ship’ 156 cases, ‘interest’ 211 case, ‘sugar’ 135 cases, ‘gold’ 99 cases, ‘coffee’ 114 cases and ‘gnp’ 73 cases. However, classification of such a large number of categories is not particularly relevant to the core classification problem (where the number of categories of interest in BCKO is closer to 2 or 3). Thus, results on larger numbers of classes will not be discussed further in this thesis.

### 3.2. Algorithms

#### 3.2.1. Data pre-processing

In this project, I concentrate only on the ‘pure’ core body text itself. Much information is considered unhelpful in the HAL model. The web pages are ‘cleaned’ in the following process:

- Remove all the tags and tables and forms.
- Remove the carriage return and paragraph information.
- Remove images and hyperlinks.
- Remove punctuation and non-alpha words
- Remove the names of dialogists.
- Remove stop words.

The stop words list used in this project is from <http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop> and was discussed in [41]. This list, containing 571 words, is included in Appendix A.

#### 3.2.2. Feature Selection

Since the HAL matrix,  $H$ , is both very large (~7000 by 7000) and very sparse, use of the entire matrix in the classifier development process is both computationally cumbersome and also conceptually out-of-keeping with the concept of Semantic Spaces (which assumes the number of relevant semantic dimensions to be less than the total vocabulary size). I first sort the vectors (columns) by the sum of their items in descending order and select the first  $n$  columns ( $n < N$ , say  $n = 450$ ), then sort the rows by the sum of the values in the selected columns for each row in descending order and select the first  $m$  number rows ( $m < N$ , say  $m = 100$ ). Now I have a sorted  $m \times n$  matrix much smaller than the original (called matrix  $H'$ ). Table 5 shows the dominant terms in each of the medical and supportive types of websites (i.e., those words with the largest vector sums in the HAL matrices based on their text corpora). I then move every succeeding column from the second one to the end of the first column, finally achieving a single vector of length  $m \times n$ . Each matrix for each webpage is going to be converted to a vector and associated with its type (i.e., medical or supportive). The vectors for all  $r$  webpages (say 100) are assembled into a matrix in



## Methods

which each row represents a webpage (called matrix  $H^*$ ); the corresponding type of each webpage will be put into another single-column matrix.

We assume that the most important HAL-based features are among those for which I have abundant data. Thus, I sort each  $m \times n$  vector in descending order of the sum across the  $r$  webpages. Since in our case  $m \times n$  equals 45,000 and is still a large feature set, I often use just some of the  $p$  features,  $p < m \times n$ , with the largest sums.

**Table 5 Largest HAL vector sums for text corpora based on the Medical and the Supportive web sites**

Medical		Supportive	
Word	HAL value	Word	HAL value
cancer	145047.0	cancer	79304.0
breast	127739.0	children	57217.0
women	73062.0	treatment	34776.0
treatment	56426.0	time	34641.0
patients	43493.0	people	31541.0
chemotherapy	34381.0	child	30999.0
risk	31878.0	death	29979.0
therapy	26450.0	feel	29225.0
cells	26324.0	life	28633.0
disease	25479.0	breast	28563.0
studies	23130.0	family	26311.0
effects	22758.0	make	20878.0
estrogen	21283.0	care	19959.0
brca	19175.0	things	19649.0
study	18443.0	dick	19434.0
tumor	18248.0	foley	19340.0

Table 6 shows the HAL matrices for the medical and supportive samples (showing just the subset of the matrices with the 10 words with the highest HAL scores and the 20 words with the largest co-occurrence). Differences are apparent – for example, consider the larger use of the word ‘children’ against the 10 dominant words in

## Methods

supportive versus medical pages. With such visually-apparent differences in the HAL matrix it is unsurprising that we should be able to develop classifiers that can distinguish these differences automatically. Decision tree/forest classifiers are developed on a larger matrix of the top 100 highest scoring words from each of the supportive and medical training sets.

**Table 6 Highest-value components of HAL matrices for 80 Medical and 80 Supportive web sites.**

medical	cancer	breast	women	treatment	patients	children	time	chemotherapy	risk
cancer	5200	15371	3570	2047	1387	40	249	925	2961
breast	15371	4448	3407	1516	1184	18	288	671	2496
women	3570	3407	2440	839	216	30	182	382	1220
treatment	2047	1516	839	1244	506	25	196	765	98
patients	1387	1184	216	506	954	0	116	853	201
children	40	18	30	25	0	0	0	1	10
risk	2961	2496	1220	98	201	10	44	116	500
effects	452	260	262	459	89	5	87	556	65
chemotherapy	925	671	382	765	853	1	112	642	116
therapy	1007	857	370	422	399	0	56	351	164
side	324	111	186	434	62	6	80	464	50
time	249	288	182	196	116	0	132	112	44
years	674	665	956	226	313	10	41	106	191
feel	66	68	82	157	12	9	29	84	0
people	173	42	33	85	27	7	9	147	21
life	200	212	223	153	89	13	11	20	37
family	356	345	121	5	34	8	0	2	110
radiation	295	148	87	303	184	13	33	190	73
child	24	27	31	81	0	1	7	8	5
cells	1776	756	167	303	35	0	31	311	57
supportive									
cancer	2962	2984	551	1196	850	1115	529	185	203
breast	2984	686	449	369	138	113	129	116	136
women	551	449	334	139	31	53	107	138	20
treatment	1196	369	139	828	247	175	253	114	38
patients	850	138	31	247	418	20	100	70	26
children	1115	113	53	175	20	2278	511	39	4
risk	203	136	20	38	26	4	24	8	84
effects	226	92	74	444	93	41	68	76	0
chemotherapy	185	116	138	114	70	39	49	116	8
therapy	150	107	70	173	116	25	28	134	24
side	167	133	61	408	63	20	56	60	21
time	529	129	107	253	100	511	452	49	24
years	368	116	45	118	83	219	78	19	38
feel	643	167	233	252	273	522	274	67	14
people	669	47	40	131	123	438	296	21	40
life	734	136	68	197	166	361	233	38	43
family	753	103	54	117	33	577	289	0	30
radiation	211	263	96	532	53	6	68	174	62
child	288	15	1	42	12	786	249	3	0
cells	110	21	0	28	19	22	15	12	0

### 3.2.3. Classifiers

We started to address the classification problem by adapting an induced decision tree algorithm for classification [139]. While this gave promising results, we then decided to improve the solution by creating multiple decision trees, i.e., a decision forest [140], which yielded improved classification accuracy in all cases; hence we report decision forest rather than single decision tree results herein. Study of the patterns of failure in the individual decision trees and correlations in the HAL matrix revealed that a key weakness in applying decision tree type solutions to our problem is that our webpages are individually often quite brief. This results in a word that is very informative in the corpus as a whole frequently being totally absent from a particular page. This insight led us to enhancements in the way we interpreted induced decision trees, utilising what we call a validation path, and ultimately led to a simplified algorithm that yields better (and faster) classification for our problem, which we call Summed Similarity Measurement (SSM).

Our objective for this thesis is to establish if the HAL-based classifier is suitable for this Health Informatics problem, and not to pursue the Machine Learning question of what method is best per se. However, to assess that we have not made unnecessary (or at least counterproductive) innovations, we: (a) compare our decision forest and SSM to SVM, chosen specifically because of its demonstration as a solid performer on Reuters21578 [55] as well as its widespread availability; and (b) compare our choice to use features from HAL's Semantic Space model as compared to simpler word frequencies. With respect to the latter comparison, we look at word frequencies with both SVM and decision forests. Moreover, since the notion of creating multiple classifiers for a single data set is not limited to decision trees, we also look at the creation of a 'forest' of SVM classifiers.

It should be noted that classification of article tone (supportive versus medical) was used for exploration of algorithm parameters (notably, HAL window size, the number of HAL columns to retain for classification, and the number of decision trees per forest) whereas BCKO breast cancer stage and authorship, as well as the Reuters data, were used to confirm performance without undo 'data dredging' in this regard.

### Decision Tree

To create a decision tree based on HAL vectors of words in the corpus, I examine how well each candidate word  $j$  splits the training set into estimated membership (e.g. medical vs. supportive). The decision word for the root node of our decision tree is taken based on the maximum entropy reduction (i.e., information gain, a la ID3) from the parent to the child. This is repeated recursively, choosing a new best word for each node of the tree, until all training cases are correctly classified or the remaining training cases can no longer be split.

In the case of medical versus supportive, the medical group is denoted as  $\mathbf{H}'_{t[med]}$  and the supportive group is denoted as  $\mathbf{H}'_{t[sup]}$ ; the union of these two sets of vectors is the final base of the training set:

$$\mathbf{H}'_t = \mathbf{H}'_{t[med]} \cup \mathbf{H}'_{t[sup]} \quad [3-1]$$

Since cosine is well normalised and amenable to high dimensional vectors, it has been selected as a measure of association. In keeping with the dimensional theory of a HAL matrix, a high cosine on the vector for a particular word between two HAL matrices indicates that the two corpuses use the word in a similar context. Thus, the similarity of the HAL matrix for the  $i^{\text{th}}$  test website on the  $j^{\text{th}}$  word ( $j \in$  the words in  $\mathbf{H}'_t$ ) to the medical corpus is:

$$Sim(\mathbf{H}'_i(j), \mathbf{H}'_{t[med]}(j)) = \cos(\mathbf{H}'_i(j), \mathbf{H}'_{t[med]}(j)) = \frac{\mathbf{H}'_i(j) \cdot \mathbf{H}'_{t[med]}(j)}{|\mathbf{H}'_i(j)| \times |\mathbf{H}'_{t[med]}(j)|} \quad [3-2]$$

The similarity to the supportive corpus' use of word  $j$  is defined in the same manner with respect to  $\mathbf{H}'_{t[sup]}$ .

Taking word  $j$  as a candidate basis for classifying cases as medical or supportive, I simply estimate the type of the test case as being that with the highest similarity measure. Ties are taken consistently to arbitrarily go with the one branch as the estimated type (this occurs when the decision word is missing in a specific test case's corpus (see previous work [139])).

Figure 18 shows one of the induced decision trees, which uses the word 'Breast' in the first decision node (note case is ignored in processing and simply represents the case of the first instance of the word encountered in the corpus). Starting from 70

## Methods

---

medical and 70 supportive, the entropy reduction from the root to the first child node is:

$$Entropy(70,70)$$

$$- \frac{49+1}{70+70} Entropy(49,1) - \frac{21+69}{70+70} Entropy(21,69)$$

$$\text{where } Entropy(m,n) = \frac{m}{m+n} \log_2 \frac{m}{m+n} + \frac{n}{m+n} \log_2 \frac{n}{m+n}$$

Figure 19 illustrates some examples of the decision tree of Figure 18 in use to classify test cases (those in the 8th of the data held back from training and used to estimate classification accuracy). Figure 19(a) shows a webpage of type medical (case ‘169 – medical’) being tested first on its HAL vector for the word ‘Breast’. In this case  $\text{Sim}(H_{169}(\text{Breast}), H_t[\text{med}](\text{Breast})) = 0.76$  and  $\text{Sim}(H_{169}(\text{Breast}), H_t[\text{sup}](\text{Breast})) = 0.73$ , so the decision proceeds to the left-hand node. It is then assessed in terms of the word ‘Treatment’ and again found more similar to the medical corpus on this word. That node is a leaf (with all 42 training cases on the left of ‘Treatment’ being medical), so the test webpage is classed into the medical group, which is correct. Similarly, Figure 19(b) shows a successful classification of a supportive web page.

Figure 19(c) shows a classification of a special medical site. It is noteworthy that the text corpus of the website lacks any instance of the word ‘Treatment’ – resulting in a zero Sim score and requiring arbitrary resolution of the tie. In this case, an incorrect result will be given in the classical decision tree, but in my program, since the similarities in the validation path are [0.673, 0, 0.194, 0.074] for medical and [0.605, 0, 0.196, 0.077] for supportive, so  $\text{Sim}[\text{med}] = 0.941$ ,  $\text{Sim}[\text{sup}] = 0.878$ , the final result is medical, which is correct.

## Methods

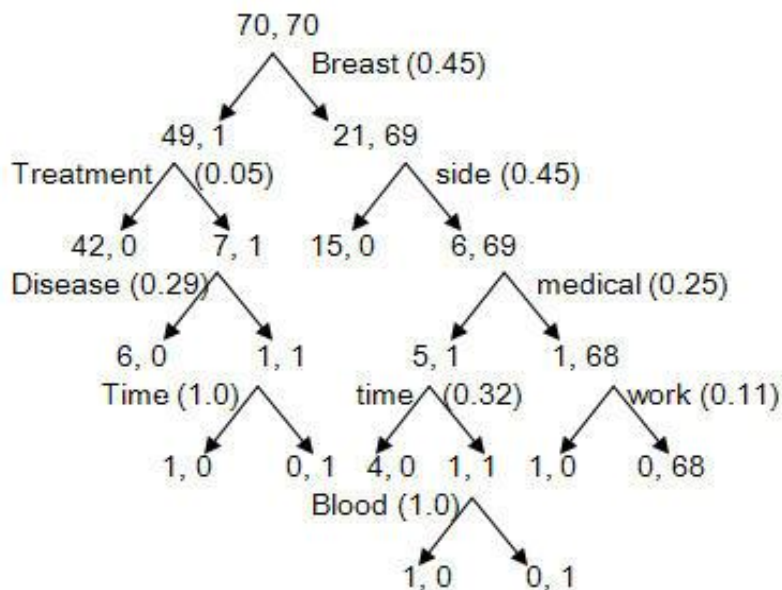


Figure 18 An induced decision tree for 70 training websites of each supportive and medical; words (e.g., 'Breast') are the decision words for that node in the tree, with the entropy reduction in parentheses; number pairs at nodes indicate number of medical and supportive documents, respectively.

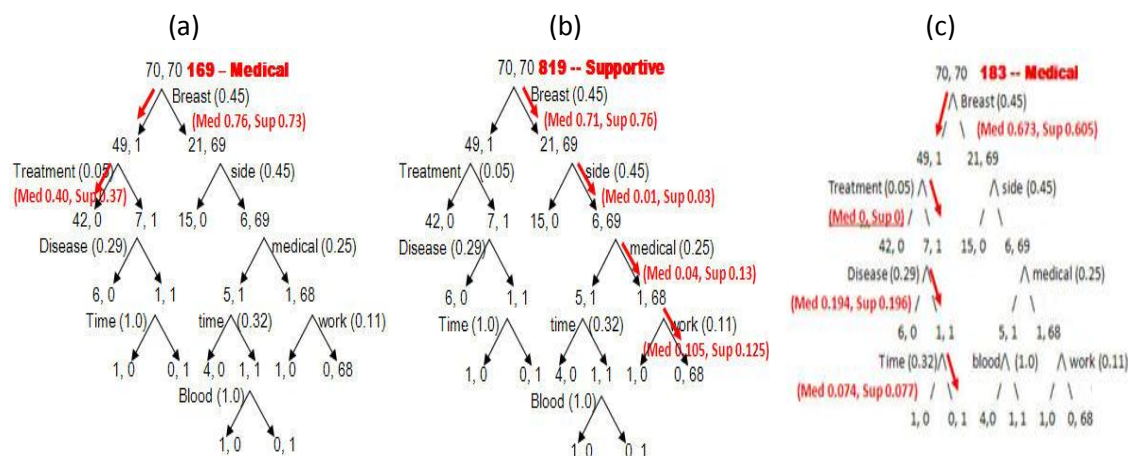


Figure 19 Decision tree showing number of 'medical' and 'supportive' cases, decision word (and its entropy gain in parentheses) annotated with validation path arrows and similarity measures. (a) Decision flow for a medical webpage in the decision tree resulting in a correct classification (the page is: <http://www.cancerbacup.org.uk/info/goserelin.htm>); (b) Correct classification of a supportive webpage on the same decision tree (see [http://my.webmd.com/content/chat\\_transcripts/1/103833.htm](http://my.webmd.com/content/chat_transcripts/1/103833.htm)); (c) classification of a medical webpage that is missing the keyword 'Treatment' (see <http://theoncologist.alphamedpress.org/cgi/content/full/5/5/393?maxtoshow=&HITS=10&hits=10&RESULTFORMAT=&titleabstract=b>)



A Validation Path (VP) is the path taken in a tree for a given test case. Figure 19(c) shows a VP for a medical test case with case ID #183. In the validation path, the keyword ‘treatment’ does not occur in case 183, and thus both of the similarities of 183 to medical and supportive are 0. Empirically, we find that the remaining entropy reduction after a tie node is relatively small, as for instance in the case of Figure 19(c).

### Round-Robin Feature Allocation

A decision tree may yield a reasonable accuracy, but in many cases the absence of a keyword (i.e., one used as a decision node in the tree) in the test webpage will lead to very unstable performance. To eliminate this problem, instead of using one single tree, I use a decision forest, taking the combination as the final result.

Generally there are two kinds of methods to split the training dataset for the decision forest: case splitting (e.g. Bagging) [78-79] and feature splitting [80]. In this project we have only a small size of dataset (see Section 3.1.2), while the number of features is huge, and HAL is a feature-focussed (see Section 1.2.3) Semantic Space model, thus we pick the feature splitting method for the classifier.

The training dataset is sorted in descending order by the sum of its elements values in  $\mathbf{H}'_t$  and the items in the training dataset are assigned to the sub-dataset for each tree in a round-robin fashion [140] (i.e. assigning one feature to every tree in turn, then starting over until every feature is allocated to a single tree). The result is what I call a Round-Robin Decision Forest (RRDF) on HAL. The reason to use Round-Robin is to spread out the sorted features and get multiple nearly equally effective trees.

The final resulting category of the classic decision forest method is determined by the majority of the results of the trees within the forest – i.e. by voting. In this project, to further minimise the impact of the lack of particular words in a given test case, and to exploit the fact that the HAL matrix is highly correlated [2], rather than taking the outcome of each decision tree directly, I create a similarity measure for class  $c$ , determined by the sum of the similarity of each node in the validation path, VP:

$$S_{i,c} = \text{Sim}(\mathbf{H}'_i, \mathbf{H}'_{t[c]}) = \sum_{j \in \text{VP}} \text{Sim}(\mathbf{H}'_i(j), \mathbf{H}'_{t[c]}(j)) \quad [3-3]$$

where word  $j$  is a node in the validation path of webpage  $i$ ,  $\mathbf{H}'_{t[c]}$  is the pre-processed training HAL matrix (see Section 3.2.2) for class  $c$ , and  $\mathbf{H}'_i$  is the pre-processed HAL matrix for webpage  $i$ . We then take the class  $c$  with the highest similarity as the decision of the tree. I call this method ‘RRDF summed similarity’.

We explore a few other variations of the RRDF concept. We apply SVM (see 0 below) on word frequency for subsets of the available words and vote for the final result (I label this ‘RRSVMoWfreq’). As another method, I follow IBM researchers who used word frequency (0, 1, 2, or  $\geq 3$ ) as the basis for nodes of a decision tree [112]. When extending this method to a round-robin decision forest, I call this RRDFoWfreq. Also, I apply RRDF to SVM on the matrix  $\mathbf{H}^*$  (see Section 3.2.2).

### Summed Similarity Measure on HAL (SSMoHal)

SSM on HAL is a variation similar to our use of the validation path for RRDF on HAL. However, instead of summing the similarities of words in the validation path only (as per equation 3-3), I sum the similarity of every common word as the final similarity measure; thus:

$$S_{i,c} = \text{Sim}(\mathbf{H}_i, \mathbf{H}_{t[c]}) = \sum_{j \in \text{path}} \text{Sim}(\mathbf{H}_i(j), \mathbf{H}_{t[c]}(j)) \quad [3-4]$$

where word  $j$  exists both in test case  $\mathbf{H}_i$  and the sum of the training cases of class  $c$  ( $\mathbf{H}_{t[c]}$ ), the matrices are all full matrices (see Section 3.2.2; i.e., not reduced to  $\mathbf{H}'_i$ ).

### AKLH

I use a shortened (typically 800 features for each case)  $\mathbf{H}^*$  matrix (see Section 3.2.2) as the input for the AKLH algorithm (described in AKLH in Section 2.3.1). To further improve the performance of AKLH, I also tried sorting this matrix (vector-of-features) based on their summed value of each feature on the same column, and it improved the performance of the algorithm substantially, but it takes the advantage of knowing the test case in training since the testing cases are taking part in the sorting. As I did not have the authority to access and edit the program for the AKLH algorithm to avoid this problem, I had to relinquish this method.



### Support Vector Machine (SVM)

In this experiment, I follow the common practice of using normalised term frequency – inverse document frequency (TF\*IDF, see Section 2.2.1) as the features for SVM.

We supply SVM with the frequencies of the 1200 most frequent words (which we call SVMoWfreq). We utilise the widely used SVMlight interface in the experiments (see “how to use” in <http://svmlight.joachims.org/>). The performance of SVMlight on Reuters21578 data has been shown to vary by only about 1% over the three common kernel functions: linear, polynomial and radial basis function [141]. Thus, we use the linear-kernel function to avoid the need for parameter tuning and to get greatest speed of performance. We also apply SVM with the 1200 highest value elements of the matrix  $H^*$  (which we call SVMoHAL).

### 3.3. Procedures

#### 3.3.1. Exploration

##### Transition Dataset

The study had two major phases. First (prior to when I had commenced work on this thesis), a Semantic Spaces researcher (Professor Warren) facilitated two transition experts to cluster the 200 largest-magnitude words in the sense-of-self vector for each of two participants in the discussion group by sequentially placing the words on a large surface. The experts were encouraged to create labels for the clusters whenever they saw a natural association emerge. These cluster maps were then used to identify word clusters to form axes for review of participant data over time. The expectation is that major transitions (a la the arrows in Figure 20) should show as measurable changes in the relationship of the participant data to axes defined by key concept word clusters. For this purpose the strength of relationship,  $r$ , of a participant,  $P$ , in a given month to a given concept cluster,  $C$ , is defined as the projection of the participant’s text for that month onto the text of the cluster:

$$r_{month}(P, C) = \text{cosine}(W_{P, month}, W_C) \quad [3-5]$$

## Methods

where  $W_{P,month}$  is a vector that is the sum of the HAL vectors for all the words used by a participant in a given month based on the corpus of that participant's emails for that month, and  $W_C$  is the sum of the HAL vectors of the concept cluster words across the entire corpus; both  $W_{P,month}$  and  $W_C$  are of length  $N$ , where  $N$  is the total vocabulary size in the corpus. If a participant did not use a given word in a given month, the corresponding element of  $W_{P,month}$  has a value of zero.

While we achieved some interesting results and learned about the nature of HAL vectors in this domain [2], the transition data set was deemed too difficult for testing our theories on the feasibility of Semantic Space models for consumer health data classification because the 'class' of different elements of the corpus (i.e. their stage of Transition) was itself somewhat subjective and would change in complex ways from month to month for any given individual. For this reason we moved on to use of the BCKO dataset where class information (in terms of metadata values) had already been clearly defined.

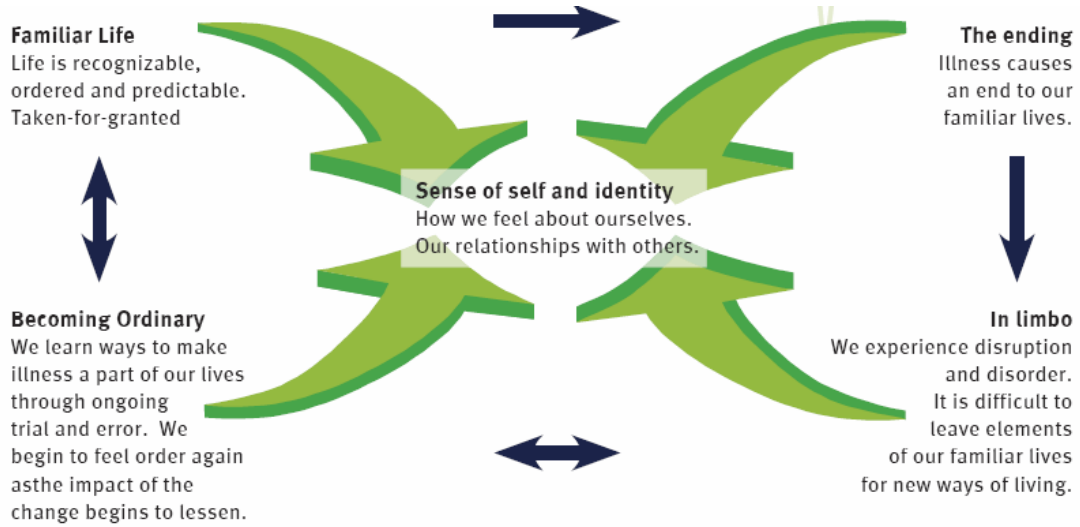


Figure 20 Stages of Transition in chronic illness (from [1])

### Adaptive K-Local Hyperplane (AKLH) on BCKO

I used 80 webpages each randomly selected from those pages exclusively coded as medical and supportive in the BCKO database. By using the matrix  $H^*$  (see Section

3.2.2), I derived a 160 by 35,000 matrix. Since this large matrix for the 160 webpages is rather computationally cumbersome, we sort each webpage in descending order by the sum of features in the same column, and pick the first  $p = 800$  features as the training set.

### **3.3.2. Comparing standard voting to summed similarity along the Validation Path for RRDF**

Instead of using the standard decision forest method, I have improved the method by summing the *similarity* between the test case and the training classes of the key words on the Validation Path. The results are shown in Section 5.1.3.

### **3.3.3. Parameter Optimization**

Utilising the HAL matrix features with decision trees, decision forests and SSM involves a few very important parameters that can deeply affect the final result of the experiment. In order to keep the result as sturdy as possible, I did a series of isolated experiments to tune those parameters in the order of how I perceived their importance combined with the order that the parameters are used in the program. The size of the window forming the HAL matrix is a fundamental parameter affecting the nature of the HAL matrix; next are the columns and rows remaining for training and testing, followed by the number of trees forming the decision forest. I have also tried using duplicated words for each tree in the RRDF and analysed the influence of the tie-resolution approach when a word is absent. Every parameter is tested individually, while the other parameters use the default values. The default window size is 10, the default column size is 350, the default row size is 250, the default tree number is 9, the default size of duplication is 0 and the tie is to the default branch of a tree.

### **Window size of HAL**

As introduced in section 2.2.2, the size of the window to build the HAL matrix is the first factor determining the character of the HAL model. The wider the window, the larger the ‘context’ that is taken to indicate that two terms are associated with one another when they appear nearby in the corpus. In Burgess and Lund’s work and

Bruza et al's work, a window size of 10 was suggested. I have tested the RRDFss with the window size from 2 to 19.

### **Columns and Rows**

Since the dimensionality of the HAL matrix is normally very high, it is quite computationally cumbersome to handle the original matrix without any selection of features, and this could lead to over-fitting problems. An appropriate trimming on the columns or rows of the HAL matrix might dramatically improve the performance of the algorithms. In this thesis, I have sorted the HAL matrix by the summed value of each column so that the meaningful words for the columns are selected. A similar procedure is applied to the rows, except the summing is restricted within the selected columns.

### **Number of trees forming the decision forest**

The number of trees that compose the decision forest is obviously an important factor that affects the accuracy of the decision forest. Given the distinctive nature of the decision tree node features in our work (i.e., HAL vectors) we felt that precedent from prior decision forest work may be unreliable, and thus experimented with a range forest sizes. I have tested all the odd tree numbers from 1 to 59.

### **Overlap**

Since in this project the features are distributed using a round-robin method, the features for each tree decrease while the number of trees increases; to ensure there are enough training features for each tree, adjacent trees share a few features, that shared part being called 'overlap'. I have tested 0 to 8 overlaps.

### **Tie Resolution**

In a decision tree, the decision has to select one branch when the feature word on that node does not occur in the training or test case, resulting in a tie. How to proceed in the use of a tie affects the final result in the case of a standard decision forest. In this project, I use the method described in section 3.3.2; the result is shown in chapter 5.

### 3.3.4. Classification Accuracy (Resampling)

To provide an accurate estimate of the performance of each classification algorithm, for each class, we randomly shuffle the dataset and pick the first ten cases as test cases and the rest as training cases. We perform 100 shuffles for the experiment and take the arithmetic mean of the results as the final accuracy of each algorithm. To test performance with limited training data, we train each algorithm and compute its accuracy on the test data for the range from a single training case up to the full training set. In each shuffle, the order of the dataset (i.e., membership of test data and order of cases for introduction as training data) is identical for every algorithm assessed. Our resampling protocol assesses performance with an equal number of cases from each class and thus is limited by the rarer class in the data set. We employ 50 randomly selected cases from each class for the supportive versus medical and lay versus clinical experiments (as this is the nearest round number to the maximum available in the rarer class in the BCKO data set); we take advantage of the larger available data supply and use 80 randomly selected cases for early versus advanced stage of breast cancer, and 400 randomly-selected cases for earn versus acq from Reuters21578.

As I mentioned in Section 2.1, for a balanced dataset, the number of cases in each class are equal, and thus the micro-averaged F value is equal to the accuracy. Since I use a balanced dataset for the resampling experiment, I simply use accuracy as the performance measurement.

### 3.3.5. ‘Natural Order’ Experiments

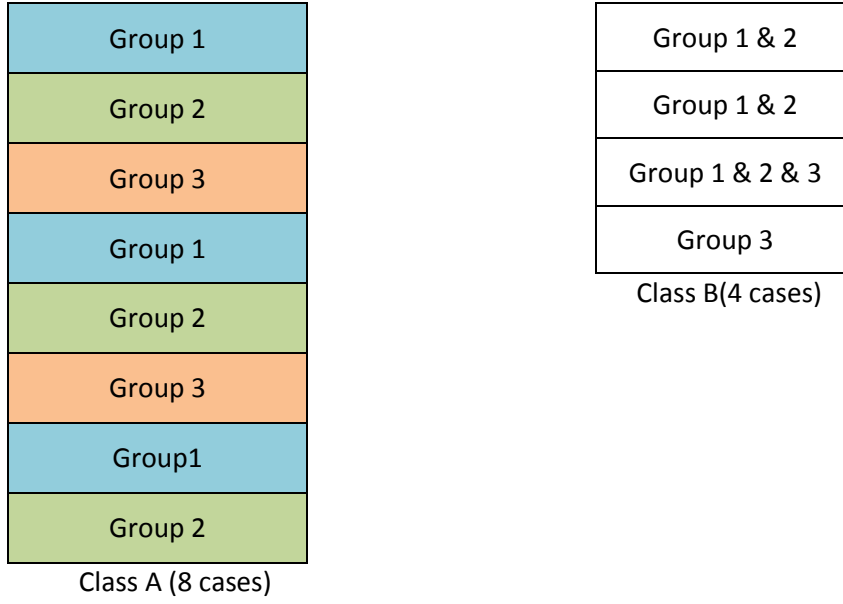
Realistic scenarios for use of our classifier do not necessarily involve a balanced number of cases from each of two alternative classes. To simulate real use we look at our data sets in the ‘natural orders’ in which the cases occurred. For Reuters21578 this is the date-time order of the news articles. For BCKO, the webpages include a case id indicating the order they were entered into the metadata database as the BCKO project progressed. In the natural order protocols, we use 150 consecutive cases (omitting only those where the metadata coders assigned the case to both of our comparison classes; i.e., both supportive and medical or both early and late). We report accuracy of the best-performing algorithms from the resampling experiments as a running mean with respect to each case as classified with all chronologically prior

cases acting as the training data. That is, the chronologically 2<sup>nd</sup> case is classified based on just the 1<sup>st</sup> case in the corpus (and hence the classifier always estimates that the membership is the same) on through to the 150<sup>th</sup> case which is classified based on the first 149 cases (with a mix of class memberships balanced such as it was in the corpus at the time the 150<sup>th</sup> case was added).

### 3.3.6. Improving Performance on Less Frequent Classes

We note that in the natural order experiments most of the misclassifications are with the less frequent classes. Skewed class frequency appears to be a common feature with the consumer health webpages, reflecting the difficulty of finding the ‘lesser voices’ (e.g., of supportive versus medical tone, or lay versus clinician authors) using conventional search engines. As such, it is desirable to have a classifier which aids identification of these more difficult to find resources. We have explored splitting training data in the bigger class into a few smaller datasets around the size of the smaller class and reusing the smaller class (i.e. over-sampling) to train multiple classifiers, then taking the majority decision as the final classification result. This approach was used previously by Chen et al. [30] to classify NCTR (National Center for Toxicological Research) estrogen activity datasets using ensemble classifiers; it was also applied by Ling and Li [142] to classify three unbalanced datasets from a Canadian bank, a life insurance company and a company ‘bouns program’ using C4.5 decision trees, and they dramatically dropped the error rate from 40% to 3%.

## Methods



**Figure 21 Illustration of the over-sampling for the unbalanced datasets.**

The method I used here is (assuming class  $A$  is the larger class, and class  $B$  is the smaller class, i.e. cardinality of  $A$ ,  $|A|$ , is greater than  $|B|$ ):

1. If  $|A| < 4$  and  $|B| < 4$ , do nothing, just use them as is, where  $|A|$  is number of cases in class 1 and  $|B|$  is number of cases in class 2;
2. If  $|A| < 1.5 * |B|$ , do nothing, just use them as is;
3. If  $|A| > 5 * |B|$ , split class 1 into 5 slots (or more for bigger differences) using shuffling method as Figure 21;
4. Otherwise ( $|A| \leq 5 * |B|$ ), split class 1 into 3 slots using shuffle as in step 2;
5. For class 2, I use consecutive cases corresponding to each slot in class 1;
6. If the total number of cases in class 2 is smaller than or equal to the number of cases in the corresponding slot in class 1, use the whole class 2;
7. Otherwise, pick a start point in class 2 using equation  $\text{slot} * (|B| - |slot|) / (SLOT - 1)$  and pick the same number of cases as corresponding slot in class 1 from class 2, where  $\text{slot}$  is serial number of the corresponding slot (0 for Group 1),  $|slot|$  is the number of cases in that slot (3 in Group 1 and 2 in Group 3),  $SLOT$  is the total number of slot (3 in Figure 21) ;

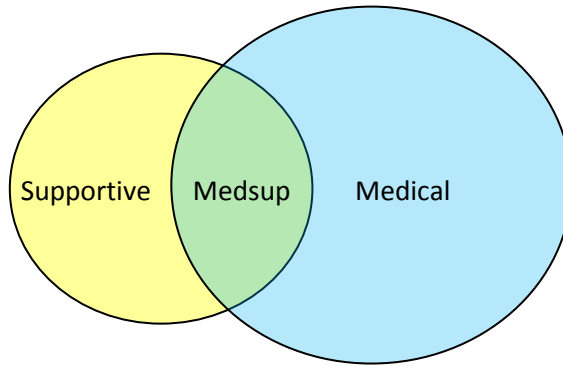
8. Get the final result by voting within those slots (between group 1, 2 and 3 in Figure 21).

In the results section, I show the revised classification accuracy for the natural order experiments with this refined handling of skewed class frequencies.

### 3.3.7. Non-Exclusive Classes Classification

Our data also present the further issue of non-exclusive class membership, where multiple values of a metadata attribute may apply to the same case. We attempt to address this issue with options that include modelling joint membership (e.g., supportive and medical tone) as a separate class or using ranges of the vote counts in a decision forest to indicate likely joint membership when votes are relatively evenly split.

We expect that this problem will be difficult in practice for those consumer health websites where the metadata is defined such that there are a large number of possible overlaps yet few available instances of such overlaps.



**Figure 22 Illustration of non-exclusive classes where the overlapping portion of the two classes is flagged as a third class (e.g. Medsup)**

Figure 22 illustrates the non-exclusive class membership that is found in the BCKO data. This is also done for stages of breast cancer (and, in that case, is particularly easy to understand – some web resources are relevant for multiple stages of breast cancer). For most of the experiments in this thesis I have ignored (omitted) the cases assigned to both the medical and supportive sets by the BCKO metadata coder.



## Methods

---

However, since this non-exclusive classification approach was chosen by the BCKO team, it is appropriate to explore some strategy to deal with it.

I treat the overlapping part of the two classes as a third class. For training, I use only cases that are exclusively supportive or medical. In testing, however, I include the non-exclusive (medsup) as a third class. The non-exclusive class should get an intermediate score as compared to the exclusively classed cases.

In the results I show how non-exclusive data is distributed based on the difference of scores from the supportive and medical classifiers and early and advanced classifiers for both RRDF and SSM.

### **3.3.8. Computational Complexity and Performance**

I will compare SVM and SSM on their computational merits, particularly time complexity and speed, in the next chapter.

## Chapter 4. Implementation and Java API

This project is implemented using the Java programming language and compiled and run under jdk1.6.

### 4.1. Data Structure

As previously mentioned, the HAL model is a very large and very sparse matrix model, so I have chosen *HashMap* as the base data structure to store the HAL matrices, such that the performance of the program in time and space complexity can be substantially improved. As an extension, I have created three data structures based on *HashMap*: *HMmatrix*, *HMclass* and *Node*.

HMmatrix is for storing a matrix using *HashMap*, where a *HashMap* maps a *String* (name of a word in the row) to another *HashMap* in which a string (a word in the column) maps a *Double* (HAL value). HMclass is for storing multiple HMmatrices, in which each id (string) maps a matrix. The Node structure stores HMclasses for each class respectively. Below are listed the structures and methods of the three classes for the three data structures. Figure 23 illustrates the three data structures.

HMmatrix:

```
class HMmatrix extends HashMap {  
    HashMap<String, HashMap<String, Double>> data;  
    HMmatrix();  
    HMmatrix(HMmatrix olddata);  
    HashMap<String, Double> get(String key);  
    void put(String key, HashMap<String, Double> input);  
    void add(HMmatrix hm);  
    HashMap<String, Double> add(HashMap h1, HashMap h2);  
    boolean containsKey(String s);  
    public Set keySet();  
    public int size();  
    public void clear();  
}
```

## Implementation and Java API

---

HMclass:

```
class HMclass extends HashMap {  
    HashMap<String, HMmatrix> data;  
    HMclass();  
    HMclass(HMclass olddata);  
    HMmatrix get(String key);  
    void put(String key, HMmatrix input);  
    boolean containsKey(String s);  
    public Set keySet();  
    public int size();  
    public void clear();  
}
```

Node:

```
class Node {  
    HMclass data[];  
    Vector<String> head;  
    int NoC;  
    Node(int noc);  
    Node(Vector<String> d);  
    Node(Node d, int noc);  
    HMclass get(int i);  
    public Set keySet(int i);  
    Vector<String> getHead();  
    void add(int clust, HMclass d);  
    int size(int clust);  
    void clear();  
    void add(int clust, String key, HMmatrix d);  
}
```

HMmatrix					HMclass		Node
W <sub>1</sub>	W <sub>11</sub>  HAL	W <sub>12</sub>  HAL	W HAL	W HAL	ID1	HMmatrix	HMclass
W <sub>2</sub>	W HAL	W HAL	W HAL		ID2	HMmatrix	HMclass
W <sub>3</sub>	W HAL	W HAL			ID3	HMmatrix	...
W <sub>4</sub>	W HAL	W HAL	W HAL	W HAL	ID4	HMmatrix	
W <sub>5</sub>	W HAL	W HAL	W HAL		ID5	HMmatrix	
...					...		
a)					b)		c)

**Figure 23 Illustration of the data structures:** a)  $W_{ij}|HAL$  indicates `HashMap<String, Double>` for the non-zero HAL value of the  $i$ th word with its  $j$ th word - an HMmatrix stores a sparse HAL matrix; b) ID is the id of a webpage, HMmatrix is its HAL matrix; c) a Node stores all the training dataset, each HMclass maps a data group.

## 4.2. Structure of Program

Implementation of the project starts from extracting the texts from file(s) (Extract class), pre-processing and cleaning up those texts (see Section 3.2.1) and building the HAL matrices (PreProc class). The main part of the program is the post-processing (ProProc class). The steps are as follows:

1. Extract text of corpus from files or hyperlinks;
2. 'Clean up' the text;
3. Build HAL matrix for each webpage;
4. Summing HAL matrix of each webpage for each data group separately;
5. Summing all the HAL matrices in the training dataset;
6. Sorting columns (rows in Figure 23 a) of the HAL matrix in step 5 in descending order by the sum of HAL values for each column. Sorting rows in descending order by the sum of the first N values for each row.

7. For SSM, use the result in step 4 to classify the test case(s); for RRDF, building the decision forest using the first  $m$  words with highest sum of HAL value and then classifying the test case(s).

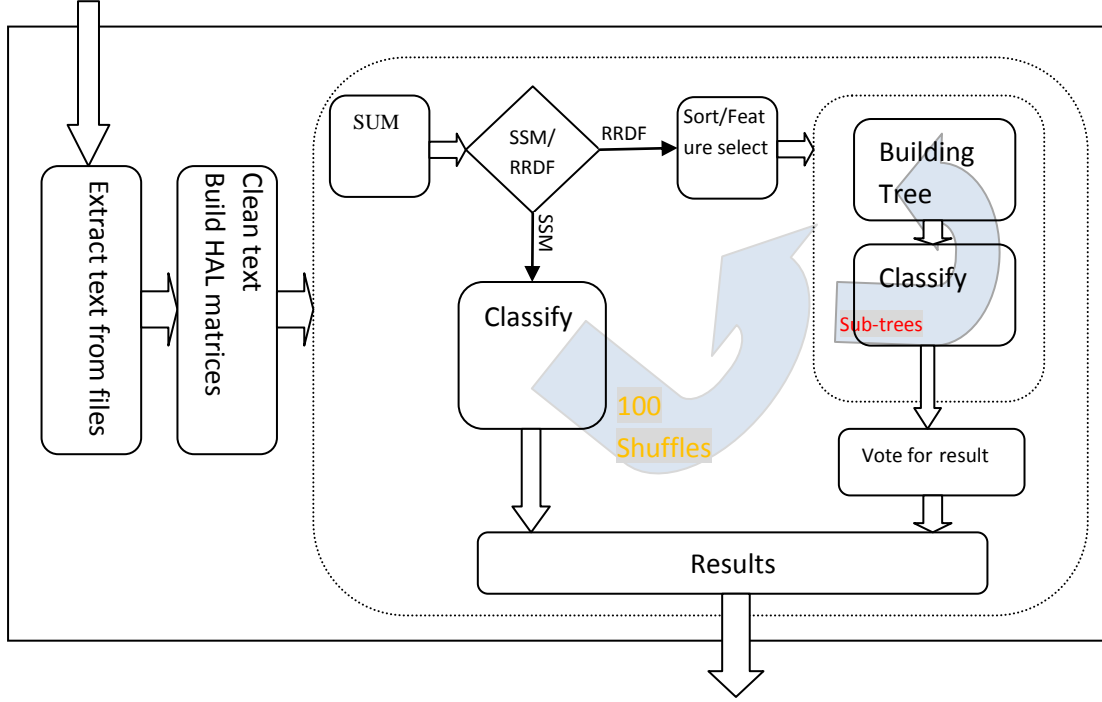


Figure 24 The structure of the software system

Figure 24 shows the design structure of the program for this thesis.

### 4.3. Java API

I have created an Application Programming Interface (API), *Classifier*, as a Java package. This API provides seven methods:

1. `public Classifier(String ts, Vector<String> class_set)`

This is the constructor of the class: *ts* is the name of the training set; *class\_set* contains names of classes involving in the classification.

2. `public Classifier(String ts)`

This is another constructor of the class for the existing training set: *ts* is the name of the training set;

3. `public boolean addTrainingData(String file, String class_id)`

Method *addTrainingData* adds training files into the training set. The training *file* could be a file, a hyperlink to a webpage (starts with “http://”) or a text of corpus (starts with “str://”); *class\_id* indicates which class that the *file* to be added belongs to. Returns true if a file is successfully added, otherwise false.

4. `public void learn(String method, int w, int trees)`

Method *learn* takes four arguments: *method* indicates the classification algorithm, either ‘*ssm*’ or ‘*rrdf*’; *w* is the size of window to build HAL; *trees* refers to the number of sub-trees for the decision forest, 0 for SSM.

5. `public String classify(String file, String method)`

Method *classify* takes *file* – hyperlink of a web page (starts with “http://”) or name of a file or a text of corpus (starts with “str://”), and *method* – the name of classification algorithm (*ssm* or *rrdf*) and returns the string of the result of classifying the *file*.

6. `public Vector<Double> getScores(String file, String method)`

*getScores* is another version of method *classify*; it takes the same arguments as *classify*, but it returns a group of percent values that reflect the proportion of similarity of the *file* to all the classes in the training set.

7. `public Vector<Double> getHalVec(String word, String class_id, int n)`

*getHalVec* returns the first *n* HAL values of the Vector for a *word* in class *class\_id* in training set *ts*. *class\_id* can be ‘*all*’ for the sum of all matrices of the training case in this training set.

8. `public Vector<Double> getHalVec(String word, String class_id)`

*getHalVec* returns the HAL values of the Vector for a *word* in class *class\_id* in training set *ts*. *class\_id* can be ‘*all*’ for the sum of all matrices of the training case in this training set.

### 4.3.1. A simple example of the API in use:

Below is the Java code for a simple example application using the API.

```
import java.util.*;
import org.teac.Classifier;

class myclassify {
    public static void main(String args[]) {
        try {
            Scanner sc = new Scanner(System.in);

            Vector<String> classNames = new Vector<String>();
            classNames.add("God"); classNames.add("Planet");
            Classifier classifier = new Classifier("PG",classNames);
            Vector<String> p = new Vector<String>();

            p.add("http://marsprogram.jpl.nasa.gov");
            p.add("http://seds.org/archive/nineplanets/nineplanets/mars.html");
            // p.add("http://solarsystem.nasa.gov/planets/profile.cfm?Object=Venus");
            // p.add("http://nineplanets.org/venus.html");

            for (String s : p) {
                classifier.addTrainingData(s, "Planet");
                System.out.println("Added " + s);
            }

            Vector<String> g = new Vector<String>();
            g.add("http://en.wikipedia.org/wiki/Mars_(mythology)");
            g.add("http://www.meridiangraphics.net/mars.htm");

            for (String s : g) {
                classifier.addTrainingData(s, "God");
                System.out.println("Added " + s);
            }

            classifier.learn("rrdf", 9, 39);
            System.out.println("Learning done (press Enter to continue)");
            sc.nextLine();

            System.out.println(classifier.classify("http://www.solarviews.com/eng/mars.htm",
            "ssm"));
            System.out.println(classifier.classify("http://www.solarviews.com/eng/venus.htm",
            "ssm"));
            System.out.println(classifier.classify("http://www.crystalinks.com/marsrome.html",
            "ssm"));
            System.out.println(classifier.classify("http://ancienthistory.about.com/od/aphroditevenus/
            a/Venus.htm", "ssm"));
        }
    }
}
```

## Implementation and Java API

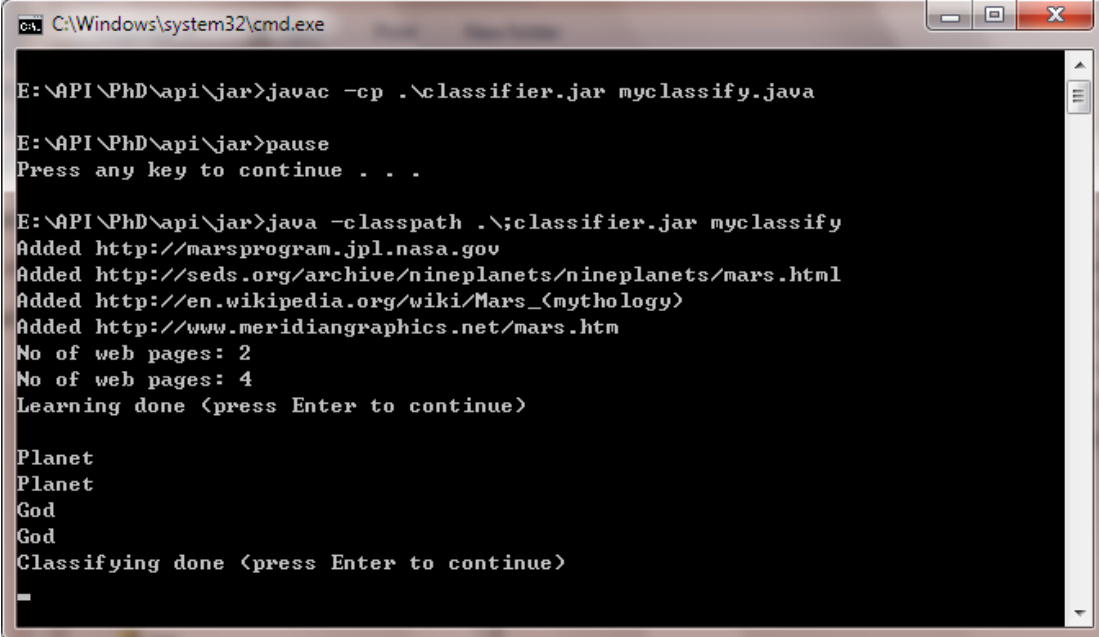
```
        System.out.println("Classifying done (press Enter to continue)");
        sc.nextLine();

    }
    catch (Exception e) {
        e.printStackTrace();
    }

}

}
```

The output of the test program:



```
C:\Windows\system32\cmd.exe

E:\API\PhD\api\jar>javac -cp .\classifier.jar myclassify.java

E:\API\PhD\api\jar>pause
Press any key to continue . . .

E:\API\PhD\api\jar>java -classpath .\;classifier.jar myclassify
Added http://marsprogram.jpl.nasa.gov
Added http://seds.org/archive/nineplanets/nineplanets/mars.html
Added http://en.wikipedia.org/wiki/Mars_(mythology)
Added http://www.meridiangraphics.net/mars.htm
No of web pages: 2
No of web pages: 4
Learning done (press Enter to continue)

Planet
Planet
God
God
Classifying done (press Enter to continue)
```

Figure 25 Output of the example program using the API

As can be seen from the code and the output in Figure 25, the system is trained with two web pages about Mars the planet (as class ‘Planet’) and two webpages about Mars from mythology (as class ‘God’). The program then classifies further webpages – one about planet Mars, one about planet Venus, one about mythological Mars and one about mythological Venus – and in each case correctly classifies the topic as planet or god. It is able to correctly place the sense of Venus although the training data was entirely concerned with Mars.

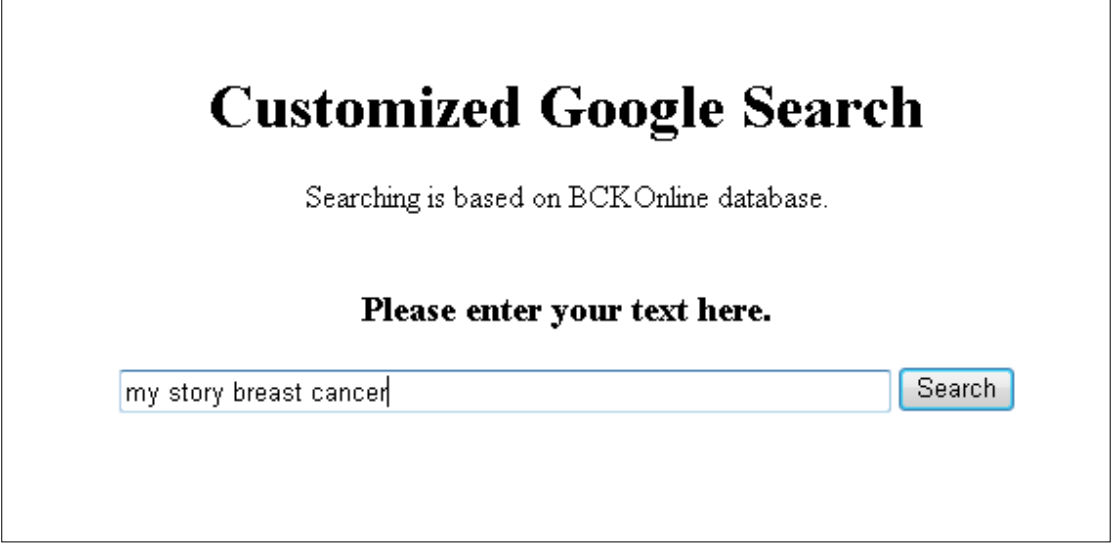
### 4.3.2. A larger example – Customised Google interface:

Using the API, I have built a customised Google searcher, to illustrate the sort of application that might be of use to a health consumer. The searcher takes the results



from Google through the AJAX interface, then identifies the nature of the webpages according to a pre-trained classifier and tags it at the end of Google results. Currently this customised search engine is working based on the BCKO dataset. It could be enlarged in the future or even have an interface added for the end users to train their classifier by themselves.

The query entry interface of the search engine is shown in Figure 26. Figure 27 shows the search result for ‘my story breast cancer’. In these examples the classifier is trained with supportive and medical cases from BCKO and offers these class names as tags for the end user based on SSM.



The image shows a web interface titled "Customized Google Search". Below the title, it says "Searching is based on BCKOnline database." There is a prompt "Please enter your text here." followed by a text input field containing "my story breast cancer" and a "Search" button.

**Figure 26** Interface for the customised search engine.

# Implementation and Java API

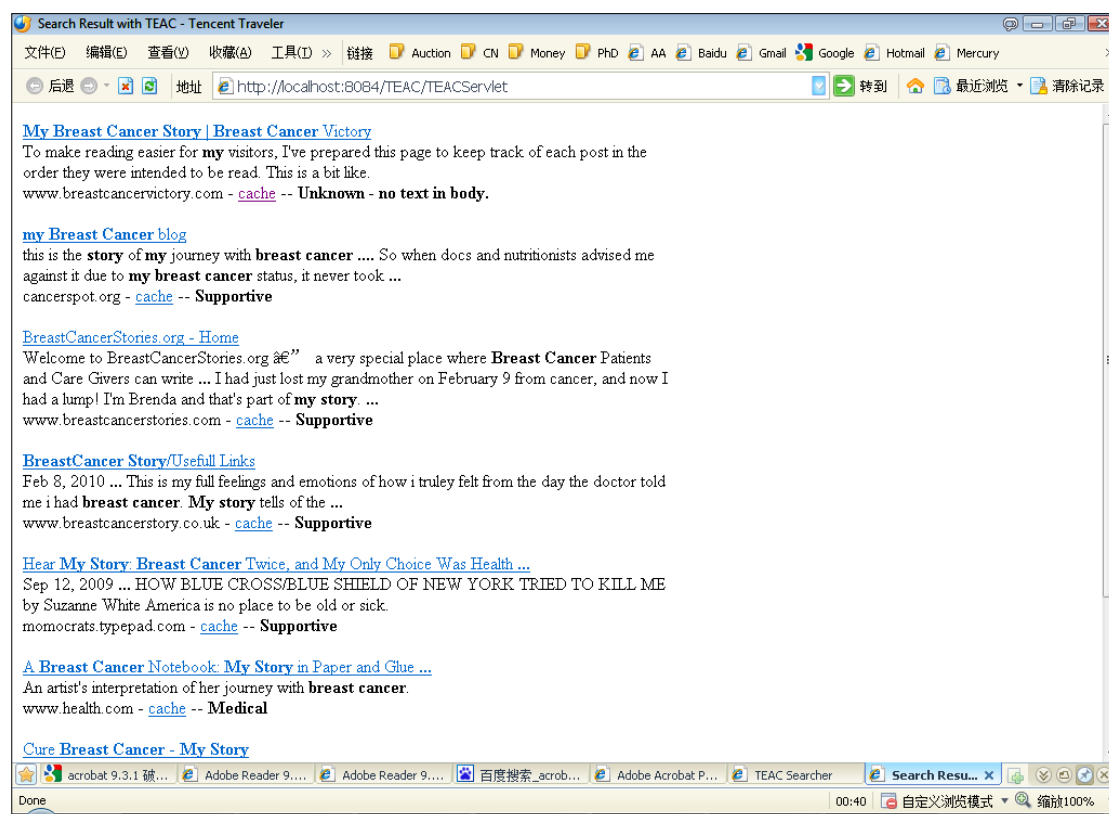


Figure 27 Tagged search result for "my story breast cancer"

The bold black word after 'cache' is the tag that the program added to the Google result. Figure 28 shows another example for searching 'tamoxifen'. In this figure we can see Search result for 'tamoxifen' is normally 'medical' web pages.

# Implementation and Java API

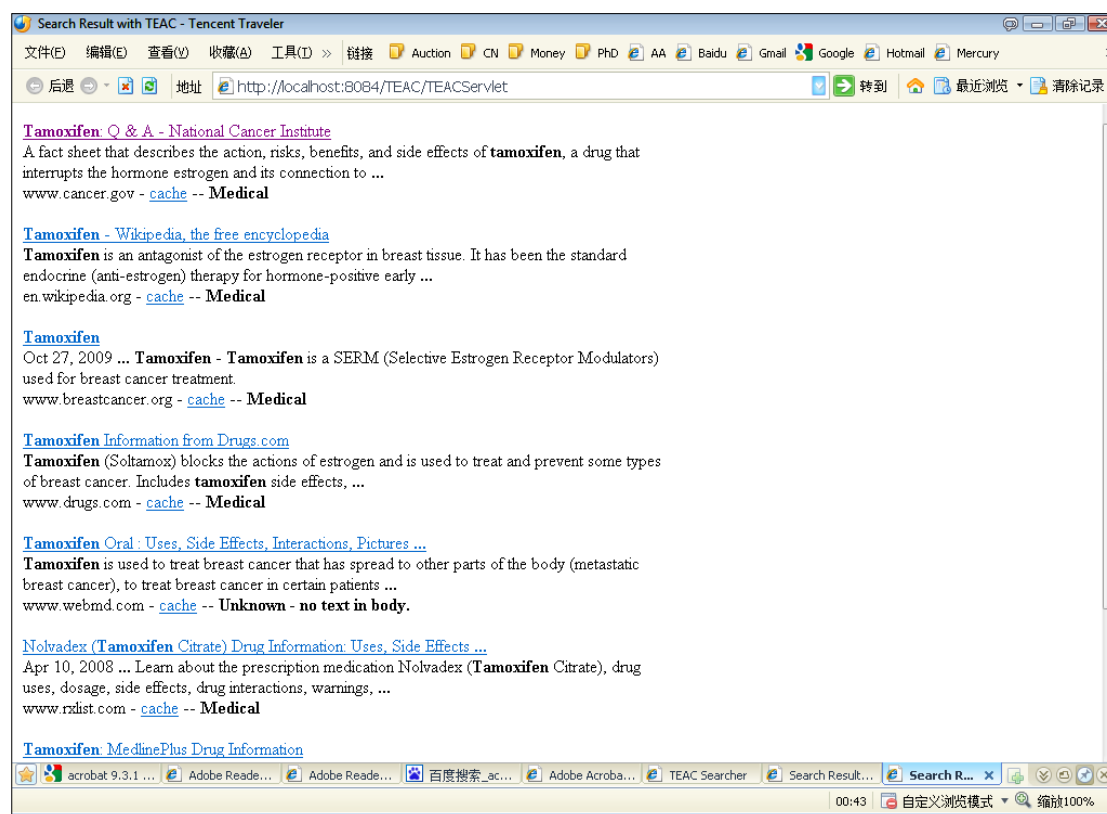


Figure 28 Tagged search result for 'tamoxifen'

## Chapter 5. Results

### 5.1. Exploration and Parameter Optimisation

#### 5.1.1. Transition Dataset

The cluster maps for two participants ('Susan' and 'Cyndi', not their actual names) were formulated by expert clustering of their sense-of-self HAL vectors (Figure 29 and Figure 30, respectively). Promising segments of Susan's map (marked on Figure 29) have been used to define axes for projection for further analysis of Susan's transitions in language usage over time as she participated in the forum. Table 7 shows the 20 largest HAL values and their associated words for these three new axes as well as the two axes used in [59]. Figure 31 shows the projection, as per equation 3-5, of Susan's submissions to the discussion forum by month onto these five axes. Susan's stages of transition were assessed by a transition expert who moderated the forum based on qualitative analysis of her text entries to the forum. The projections in Figure 31 map poorly to those identified states, but reveal that HAL vectors are highly correlated.

The transition data presented challenges for application of HAL for classification. Notably, the transition data lacked well defined class membership for training, also the story-talking and personalised writing style made it more difficult. Thus, I chose to pursue a problem amenable to classifier development applying HAL, i.e., consumer webpages as organised by the BCKO portal.

While I chose to move away from using it further in this thesis, the Transition data set provided valuable experience with HAL. For instance, it demonstrated that HAL vectors that were appealing to domain experts could be derived automatically from a consumer-authored corpus (similar to some of our supportive data in the BCKO data set). I also found that HAL vectors for different domain terms in a corpus tend to be highly correlated.

# Results

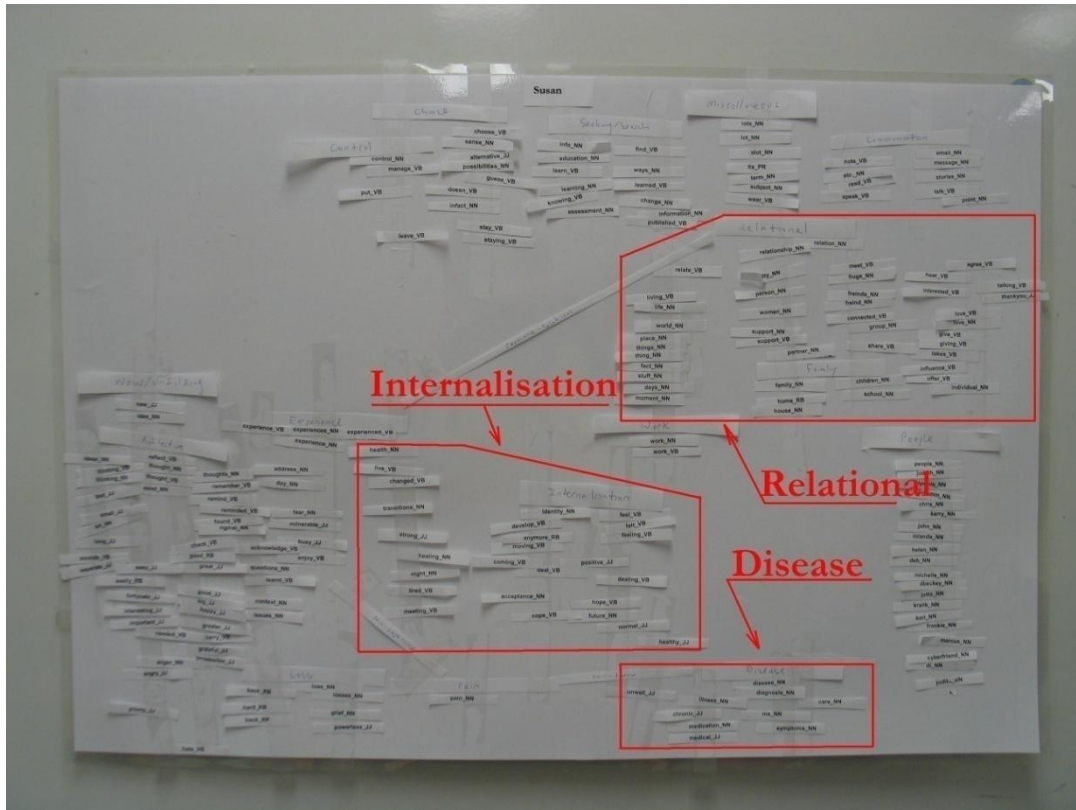


Figure 29 Manually produced cluster map of largest sense-of-self terms for Susan (overlaid with subsequently selected projection axes) [2]

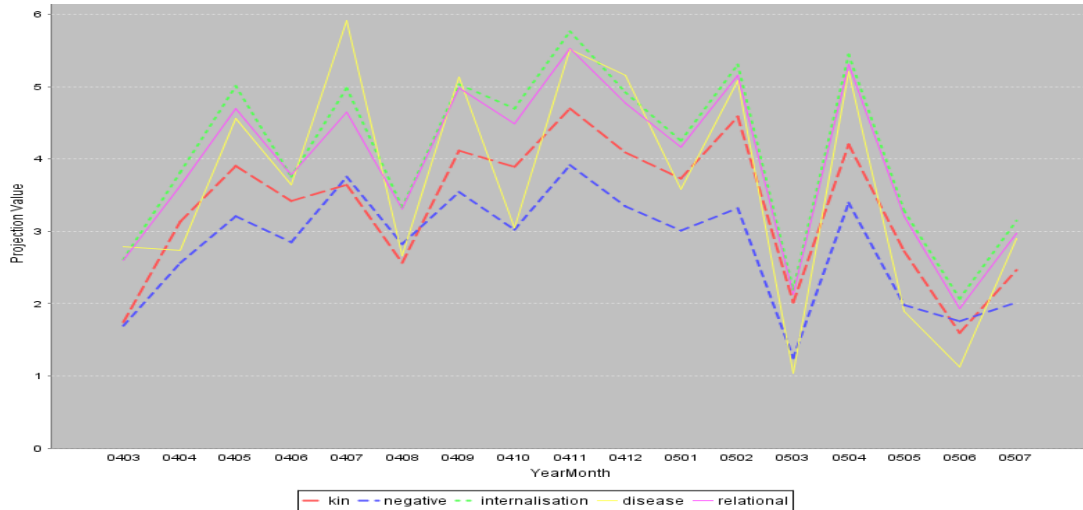


Figure 30 Segment of manually-produced cluster map for Cyndi [2]

# Results

**Table 7 Largest values for HAL vectors of projection axes [2]**

<u>Kin</u>	<u>Negative Emotion</u>	<u>Internalisation</u>	<u>Disease</u>	<u>Relational</u>
time 53.6	pain 711.9	time 247.81	illness 808.3	good 192.86
friends 40.4	fatigue 306.5	feel 238.88	chronic 624.6	time 190.81
good 35.8	back 255.75	good 227.12	people 341.4	people 187.44
children 34.4	chronic 230.0	people 202.31	living 323.3	michelle 166.61
mother 32.1	time 188.0	hope 198.27	life 262.7	illness 148.47
work 31.3	feel 183.75	things 193.12	pain 261.4	things 144.61
years 30.9	day 183.0	illness 192.38	experience 189.9	life 137.72
life 30.4	hope 179.5	life 181.65	feel 186.7	feel 134.56
told 28.1	life 167.88	pain 176.65	time 166.5	hayden 129.5
family 27.9	bad 165.88	day 149.77	health 163.5	life's 126.97
back 27.8	experience 147.9	feeling 147.15	good 163.4	adam 118.5
things 27.0	things 144.88	back 146.69	person 155.0	hope 117.5
day 26.3	people 144.0	lot 130.65	things 141.9	precious 113.06
home 26.3	feeling 138.75	di 120.15	long 125.9	day 111.36
people 25.7	lot 137.88	michelle 119.0	important 118.9	mum 110.33
feel 24.2	worse 132.88	bit 117.69	back 111.8	back 108.86
hope 23.1	good 127.75	today 114.54	find 109.0	hanna 108.39
wife 22.6	find 121.38	chronic 113.54	condition 105.9	chronic 107.33
daughter 21.8	sleep 111.63	find 107.12	changed 103.5	work 105.02
working 21.5	days 105.5	make 105.85	lot 101.8	pain 100.5



**Figure 31 Projection of Susan's discussion entries by month [2]**

## 5.1.2. AKLH method

AKLH was applied to the  $H^*$  matrix for medical versus supportive data. Figure 32 shows the accuracies for different window sizes of the HAL matrix for AKLH. Figure 33 shows the accuracies using a different sample size for each type – medical and supportive, with window size constant at 10. Figure 34 compares the accuracies using AKLH, decision forest (RRDF) with voting on HAL ( $H'$ ) and on word frequency, Decision tree (also on  $H'$ ) and K-nearest neighbour (on  $H^*$ ). To make a comparison to

## Results

a common classical algorithm, I added the result of KNN in Figure 34. These comparisons were reported in [143] – in this experiment, I used 8-fold cross validation for the classic RRDF; we can see in this figure the top three methods have shown similar performance. Note that confidence intervals were computed using the classic Gaussian estimate, which for accuracy can give results that need to be interpreted with some common sense (e.g. the upper bound being  $>100\%$  in some cases). The result in Figure 32 shows that a window size of 10 (which was recommended) leads to the best accuracy for AKLH, while larger or smaller window sizes give somewhat lower accuracy (however, noting the 95% confidence intervals, none of the differences are statistically significant).

The results, while promising overall, raised issues with access to source code on a leading edge algorithm, which was still part of a PhD thesis (that of Tao Yang under the supervision of A/Prof Kecman) at the time of this work. So I switched from AKLH to the well-established high-performance SVM algorithm, which had a popular free implementation (SVMLight, <http://svmlight.joachims.org/>).

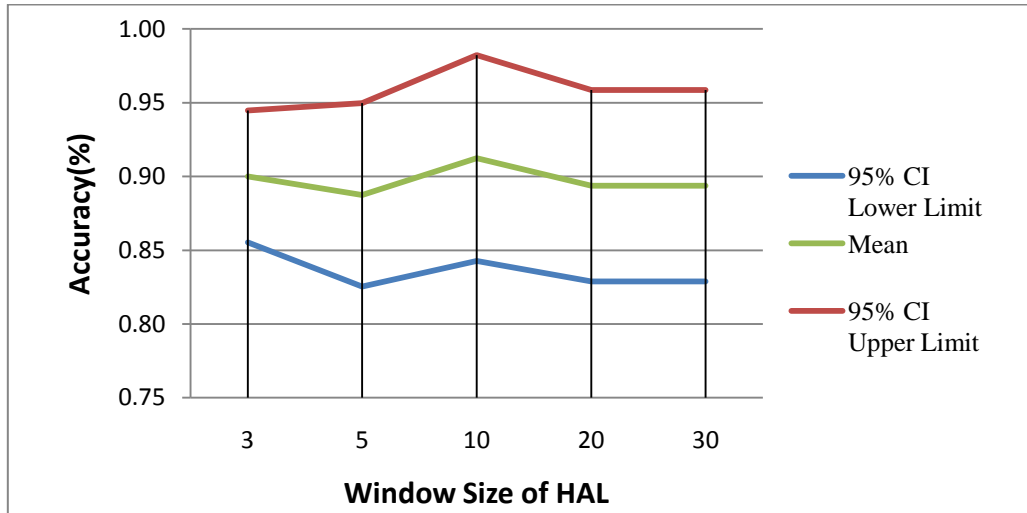
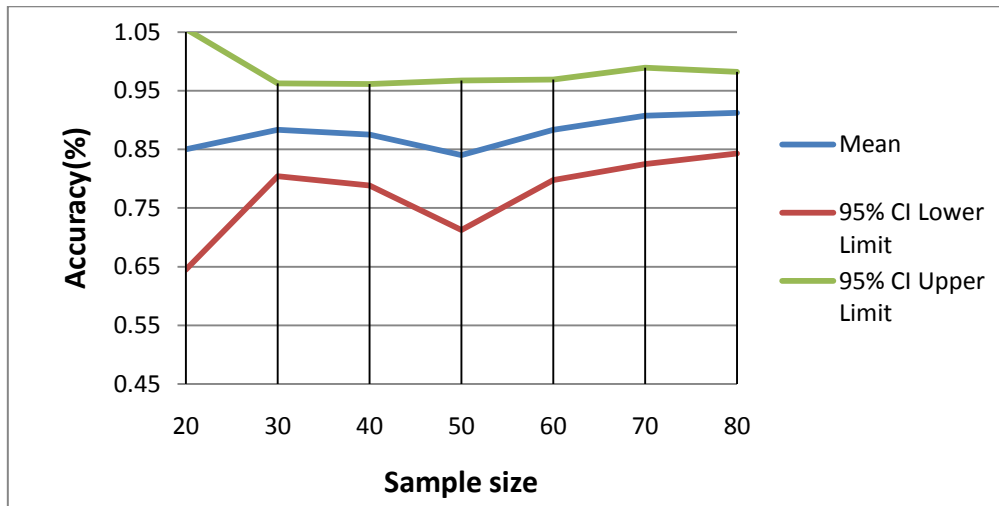
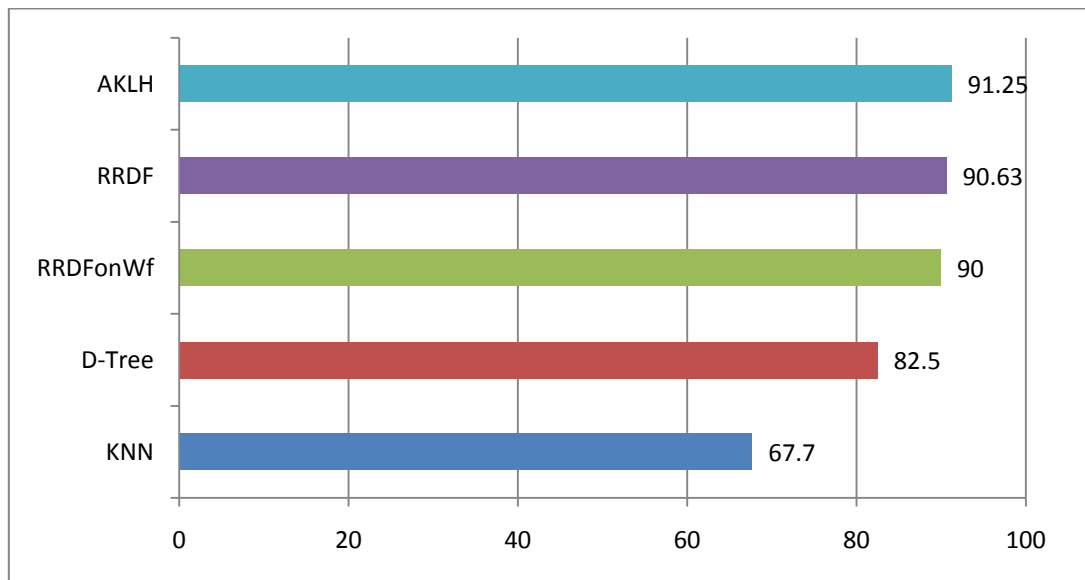


Figure 32 Accuracies for different HAL window sizes for AKLH [143].

## Results



**Figure 33 Classification accuracy by number of cases available of each type (medical and supportive) for AKLH [143]**



**Figure 34 Comparison of accuracies using different algorithms [143].**

### 5.1.3. Comparing voting to summed similarity along the VP for RRDF

Figure 35 through Figure 38 show the results of comparison between voting and summed similarity along the VP for RRDF on the attributes earn vs. acq in Reuters21578 and medical vs. supportive, early vs. advanced and lay vs. clinician in BCKO. It can be seen in those figures that RRDF with summed similarity along the Validation Path is significantly superior to standard decision forest with voting for virtually every training set size on every data set.



## Results

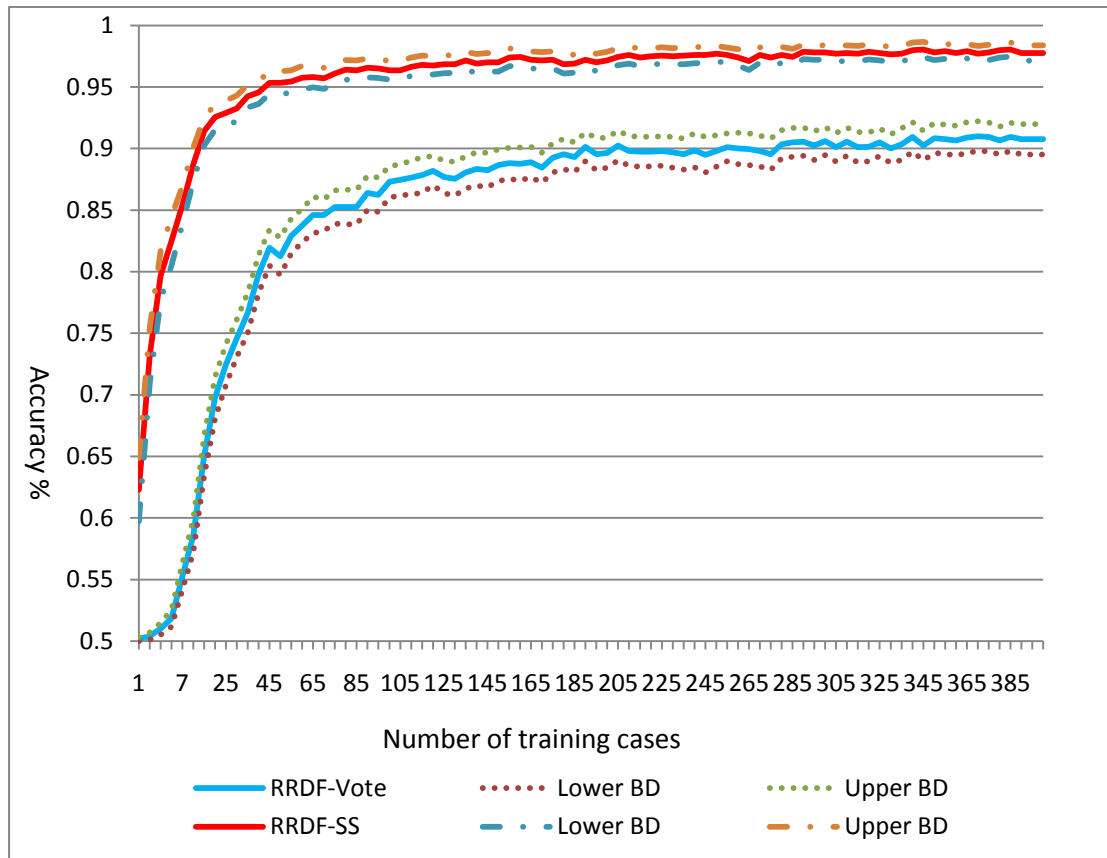


Figure 35 Comparison between standard voting and summed similarity along the VP for RRDF on earn vs. acq in Reuter21578 (accuracy and its 95% confidence interval)

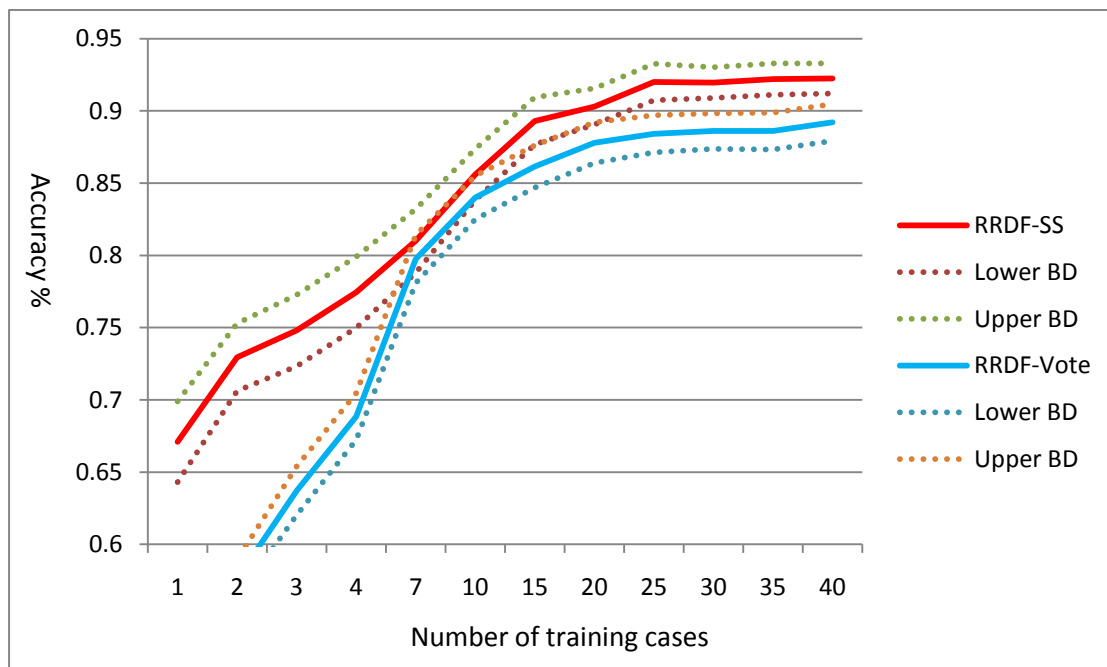


Figure 36 Comparison between standard voting and summed similarity along the VP for RRDF on medical vs. supportive (accuracy and its 95% confidence interval)

## Results

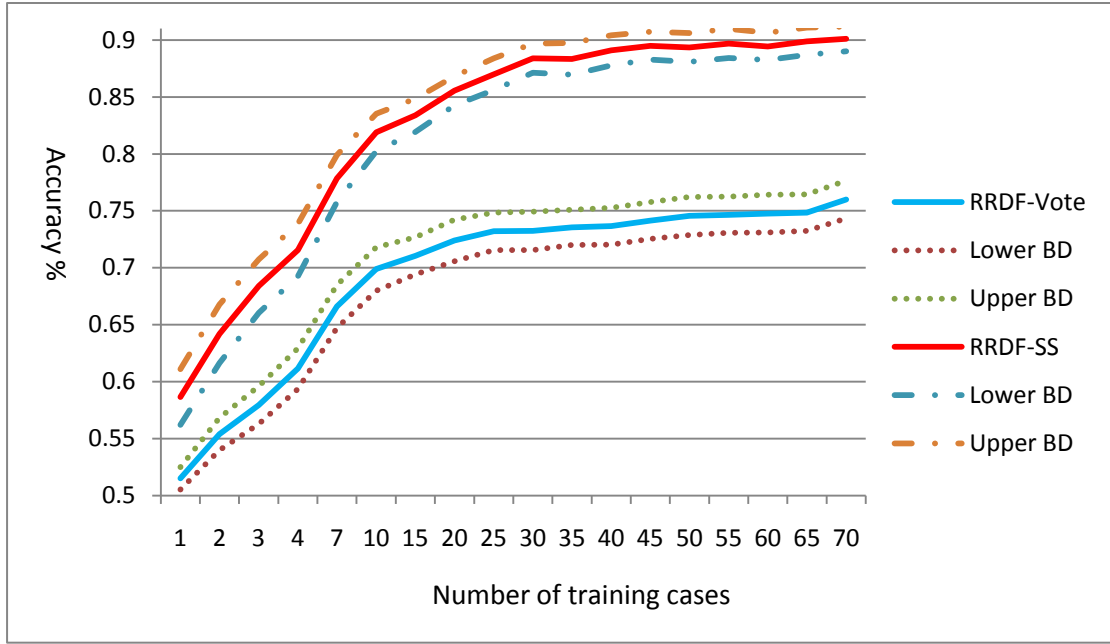


Figure 37 Comparison between standard voting and summed similarity along the VP for RRDF on early vs. advanced (accuracy and its 95% confidence interval)

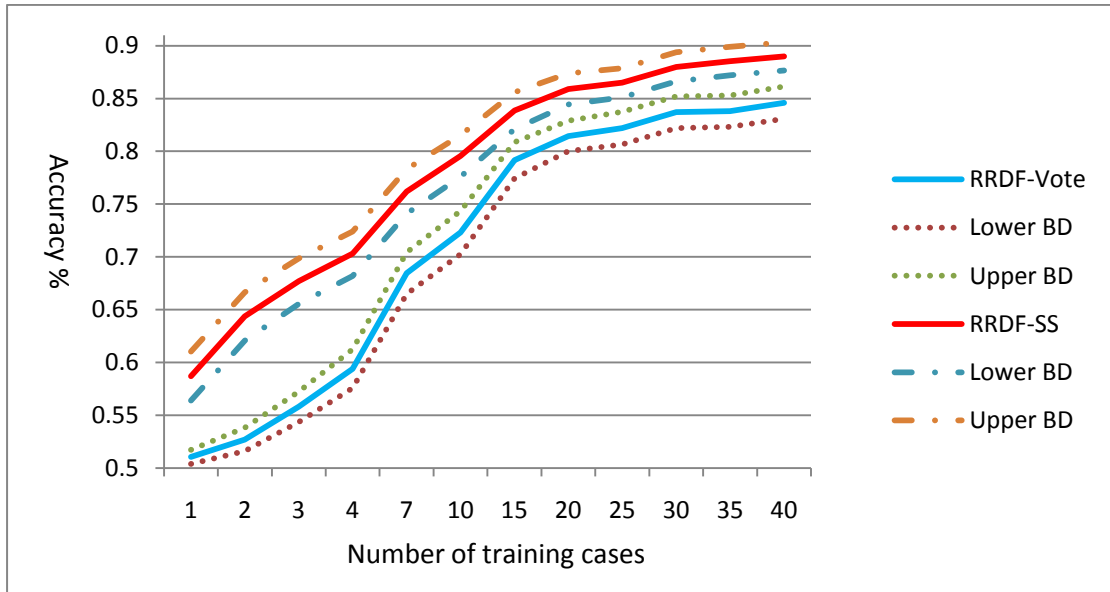


Figure 38 Comparison between standard voting and summed similarity along the VP for RRDF on lay vs. clinician (accuracy and its 95% confidence interval)

### 5.1.4. Tuning HAL parameter

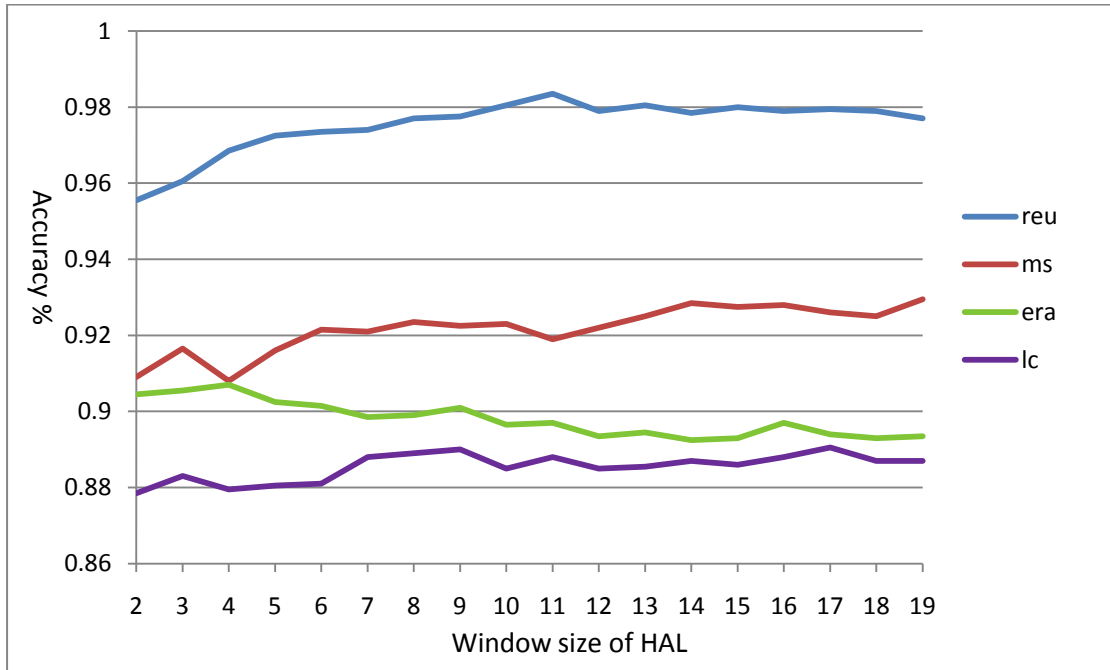
#### Size of window for building HAL

Figure 39 shows how the size of window for the HAL matrix affects the accuracy for RRDF on the four pairs of attributes: earn vs. acq in Reuters21578, medical vs. supportive, early vs. advanced and lay vs. clinician in BCKO. We can see that the

## Results

accuracy is not particularly sensitive to the size of window. In light of this insensitivity over a substantial range, I choose a window size of 9 in all subsequent analyses. A choice in this range is supported by:

- Our results in the AKLH study (see Figure 32), albeit in the absence of statistically significant differences; and
- Windows of this size have been used in past applications of HAL [59, 144].
- This size intuitively fits with a ‘neighbourhood’ somewhat longer than a single sentence and thus fits the notion of term-term matrix entries representing word context in a corpus.



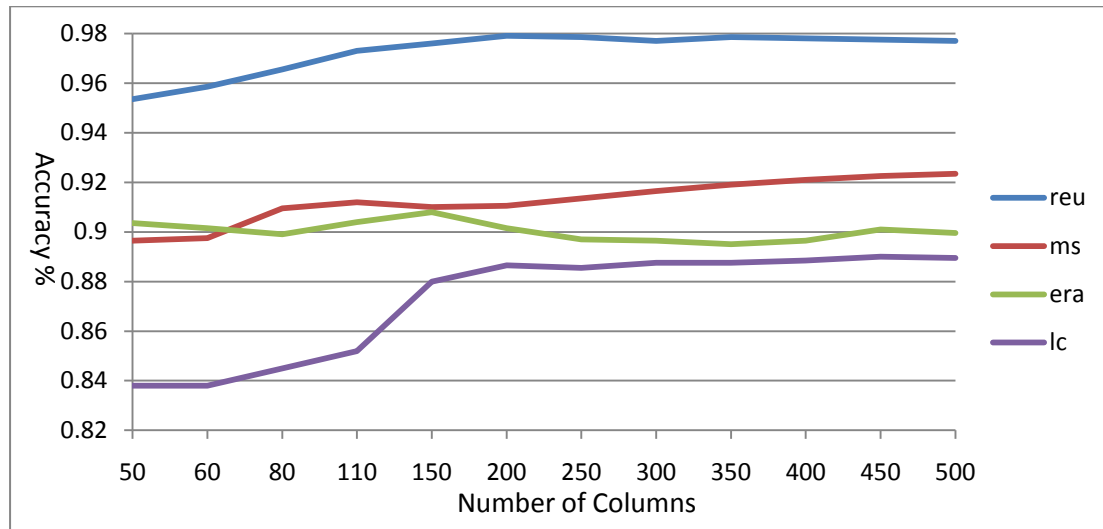
**Figure 39** Accuracies affected by the window size of HAL (‘reu’ – Reuters21578 ; ‘ms’ – medical v. supportive; ‘era’ – early v. advanced; ‘lc’ – lay v. clinical).

### Columns and Rows

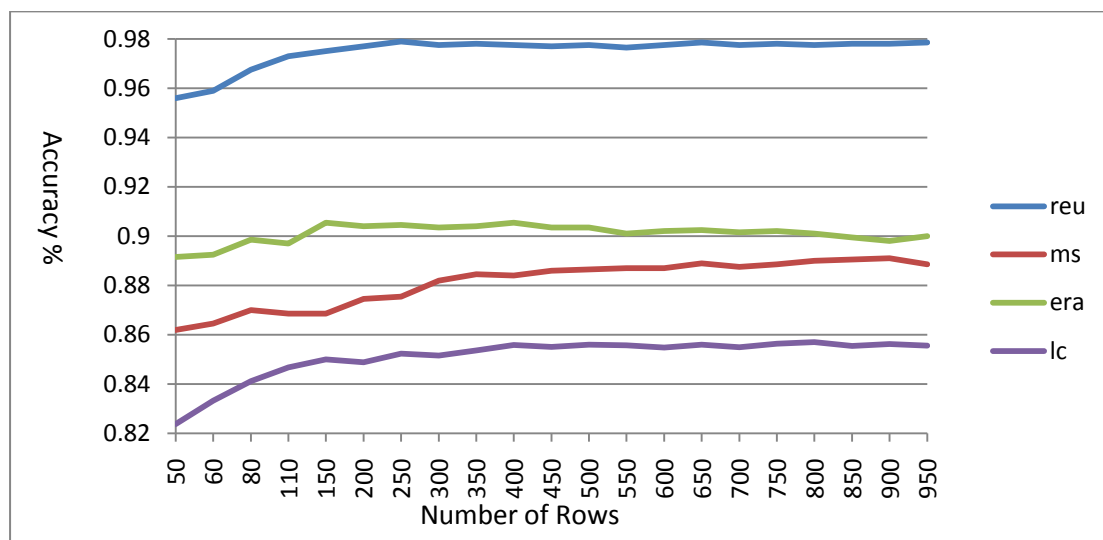
Figure 40 shows the relationship between the accuracy and the selected column size of the reduced HAL matrix on the four data sets. In the figure, we can see the accuracy gradually escalates for three out of four data sets, and finally becomes relatively constant, thus I use 450 columns as a trade-off of accuracy and speed. Figure 41 shows how the selected row size of the reduced HAL matrix affects the accuracies for the four data sets. In the figure, the accuracies increases with the

## Results

growing of the selected row size. However, using the full size of the rows can significantly increase the accuracies, therefore in this project I just use the full size for the rows.



**Figure 40** Accuracies affected by number of columns selected ('reu' – Reuters21578 ; 'ms' – medical v. supportive; 'era' – early v. advanced; 'lc' – lay v. clinical).



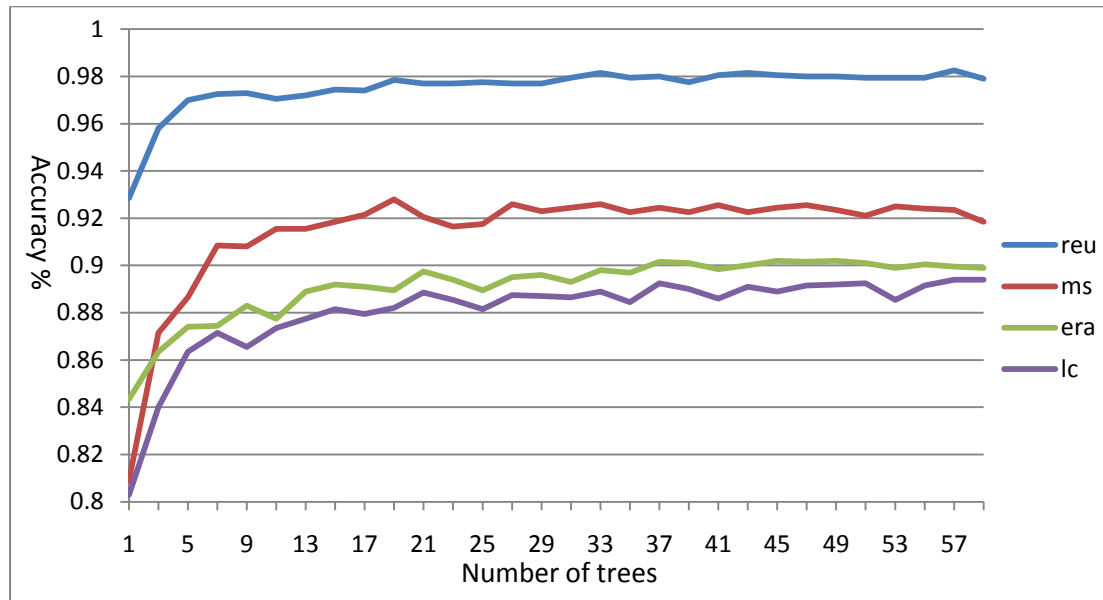
**Figure 41** Accuracies affected by number of rows selected ('reu' – Reuters21578 ; 'ms' – medical v. supportive; 'era' – early v. advanced; 'lc' – lay v. clinical).

### Number of trees forming a decision forest

Figure 42 shows the relationship between the accuracy and the number of trees in the forest. From the figure we can see that decision forest works much better than decision tree (single tree in the forest) and the classification accuracy rises with the

## Results

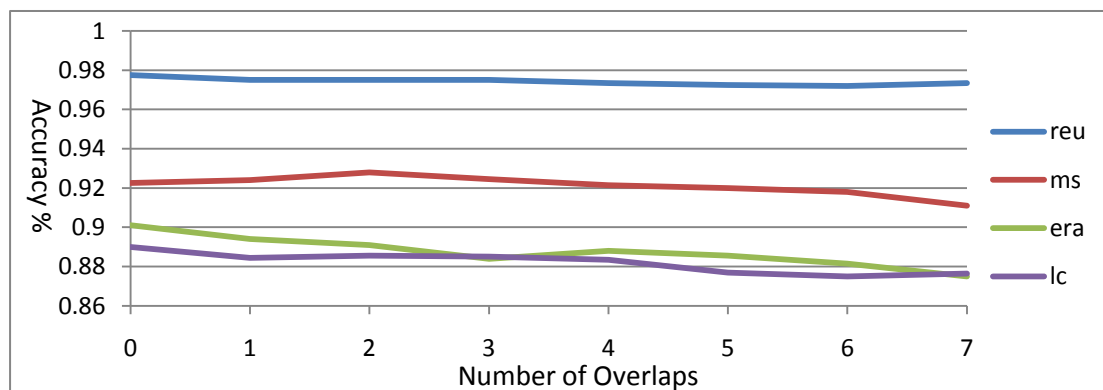
number of trees in the decision forest until about 20 trees, and is thereafter stable up to 60 trees (the largest number we tried). I chose an odd (tie-breaking) number in the middle of this range and hence use 39 trees per forest throughout this project.



**Figure 42** Accuracies affected by the number of trees in Decision Forest ('reu' – Reuters21578 ; 'ms' – medical v. supportive; 'era' – early v. advanced; 'lc' – lay v. clinical).

### Overlaps

Figure 43 shows how the size of overlap affects the accuracy. In this figure, one can see that this parameter makes the accuracy worse, so I use an overlap of 0 in the remainder of this project.

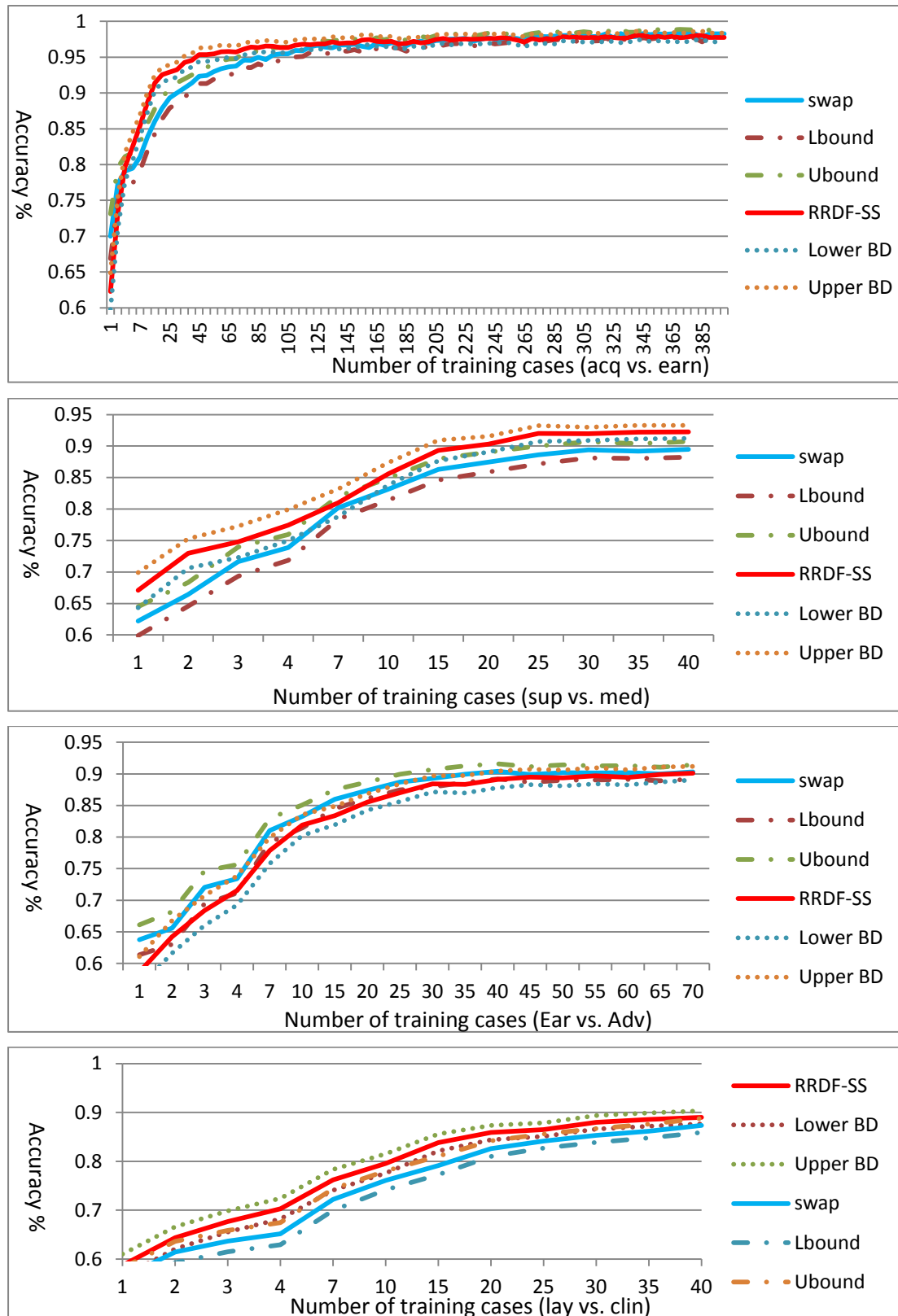


**Figure 43** Accuracies affected by Overlaps ('reu' – Reuters21578 ; 'ms' – medical v. supportive; 'era' – early v. advanced; 'lc' – lay v. clinical).

### Ties

Figure 44 shows how the accuracy varies if we change the tie resolution approach; i.e., when the keyword is missing in the test case, we pick the right-hand path instead of the left (the latter being our default). From the figure we can see the accuracy is slightly increased for two data sets and decreased in the other two. In any event, the parameter does not appear to be particularly influential.

## Results



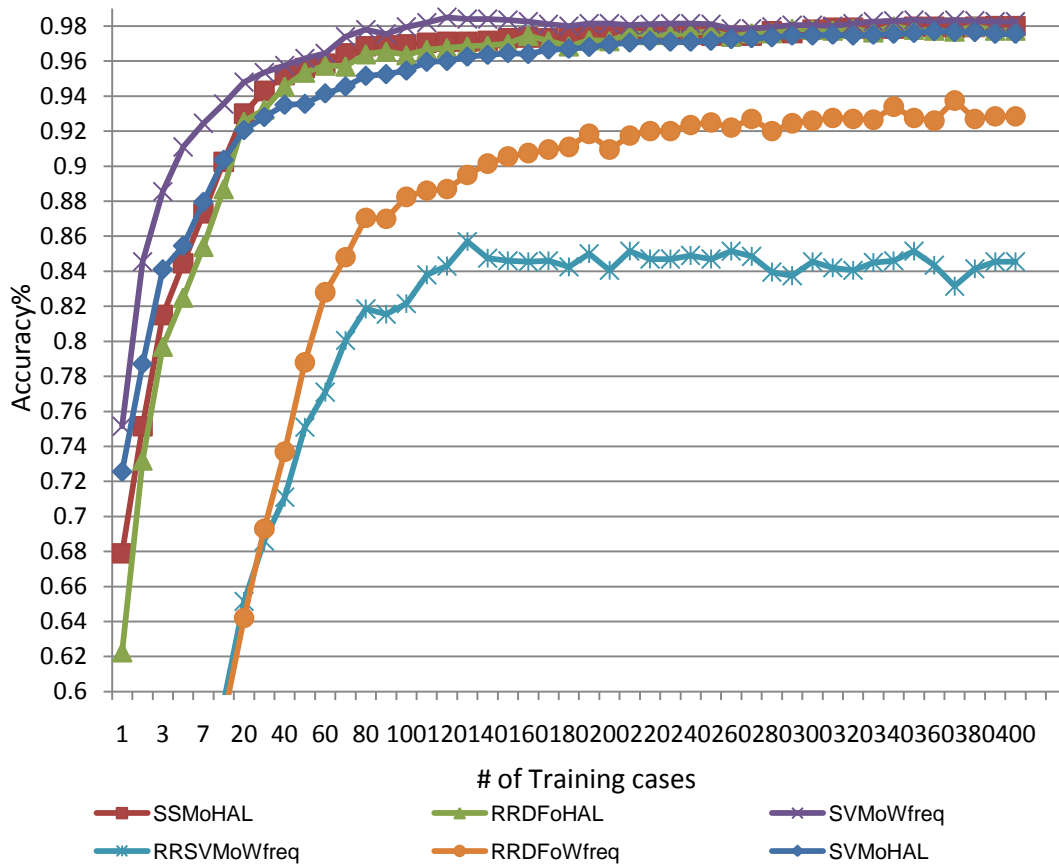
**Figure 44** Accuracies after changing the ties to right branch for all four datasets.

## 5.2. Performance Assessment

From Section 5.1.3, we can see RRDF-SS has a constant better performance over RRDF-Vote; thus, hereafter we always use RRDF-SS as our RRDF method in the remainder of this thesis.

### 5.2.1. Resampling

Figure 45 through Figure 48 show how the five algorithms perform in terms of accuracy on the balanced datasets as assessed with resampling. Table 8 shows the mean performance of each algorithm for each data set with the maximum training data, including a 99% confidence interval (CI) based on the variance over the 100 resamples. SSMoHAL and RRDFoHAL are consistently among the top performers, and SSMoHAL significantly outperforms SVM on the author credential attributes.



**Figure 45 Classification of the two most popular article categories (earn and acq) in Reuters21578**



# Results

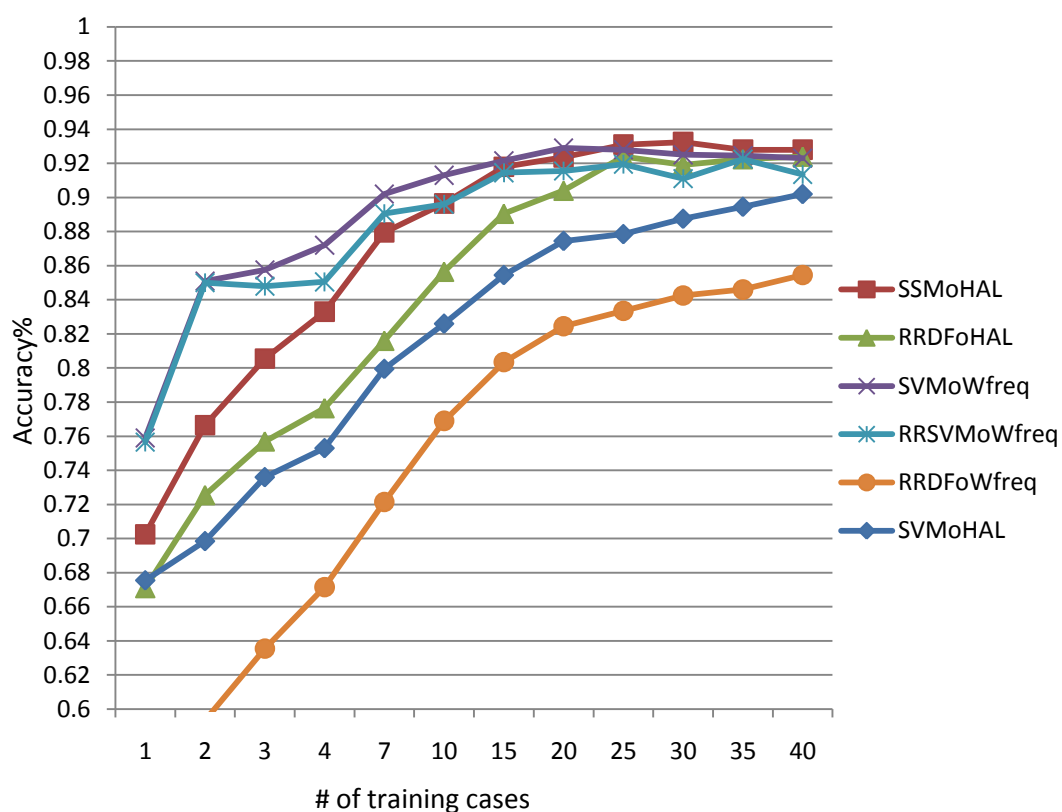


Figure 46 Classification of medical vs supportive BCKO articles

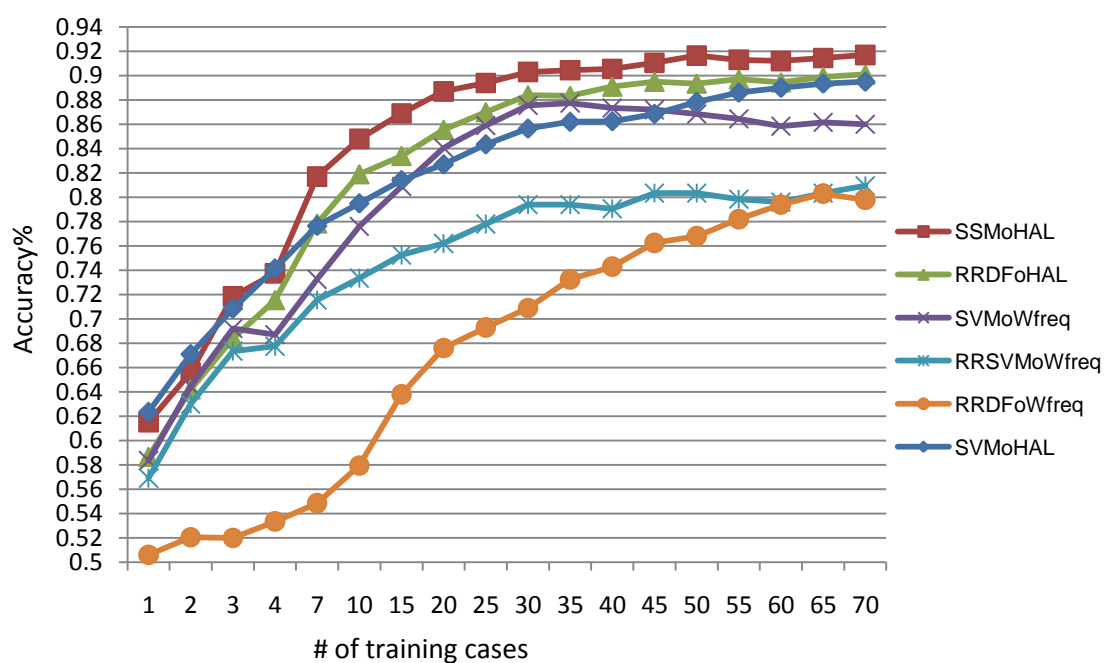
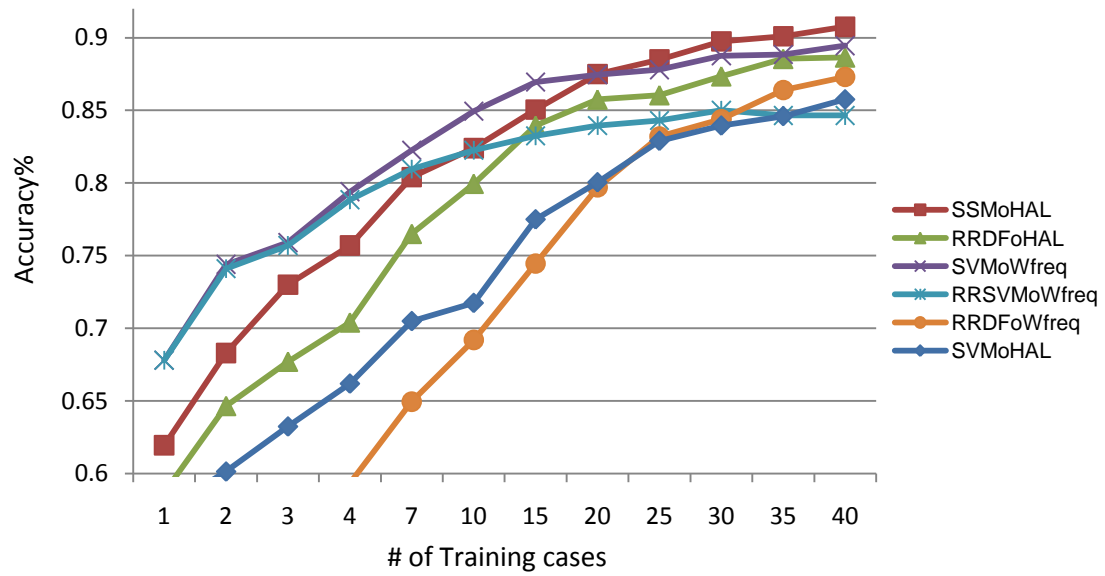


Figure 47 Classification of BCKO articles by disease stage (early vs advanced)

## Results



**Figure 48 Classification of BCKO articles by author qualification (clinician vs lay)**

**Table 8 BCKO and Reuters21578 classification accuracy (based on 100 resamples)**

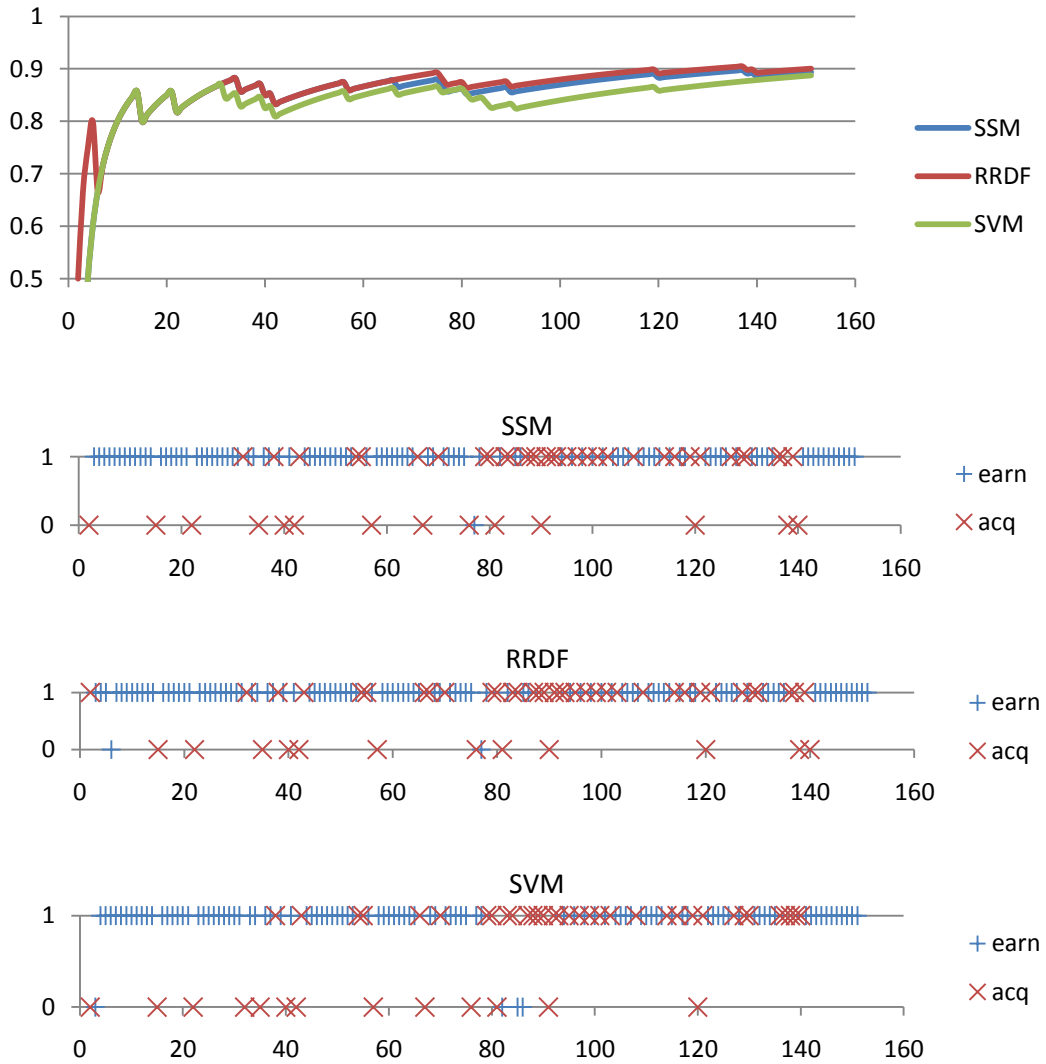
Data Set	Reuters21578		Medical vs Supportive		Early vs Advanced		Lay vs Clinical	
Training set size	390 / class		40 / class		70 / class		40 / class	
	Mean	99% CI	Mean	99% CI	Mean	99% CI	Mean	99% CI
SSMoHal	* 98.00	97.13-98.87	* 92.80	91.66-93.94	* 91.70	90.83-92.57	* 90.75	89.76-91.74
RRDFoHal	* 97.75	96.92-98.58	* 92.40	91.14-93.66	* 90.10	89.00-91.20	* 88.65	87.51-89.79
SVMoWfreq	* 98.25	97.18-99.32	* 92.30	91.28-93.32	86.00	84.68-87.32	* 89.45	87.65-91.25
rrSVMoWfreq	84.55	83.20-85.90	* 91.35	90.08-92.62	80.95	79.80-82.10	84.65	83.42-85.88
RRDFoWfreq	92.85	91.92-93.78	85.45	84.59-86.31	79.80	78.69-80.91	87.30	85.37-89.23
SVMoHal	* 97.55	96.29-98.81	* 90.20	88.27-92.13	* 89.50	86.80-92.20	85.75	83.59-87.91

\* Top performance group.

## Results

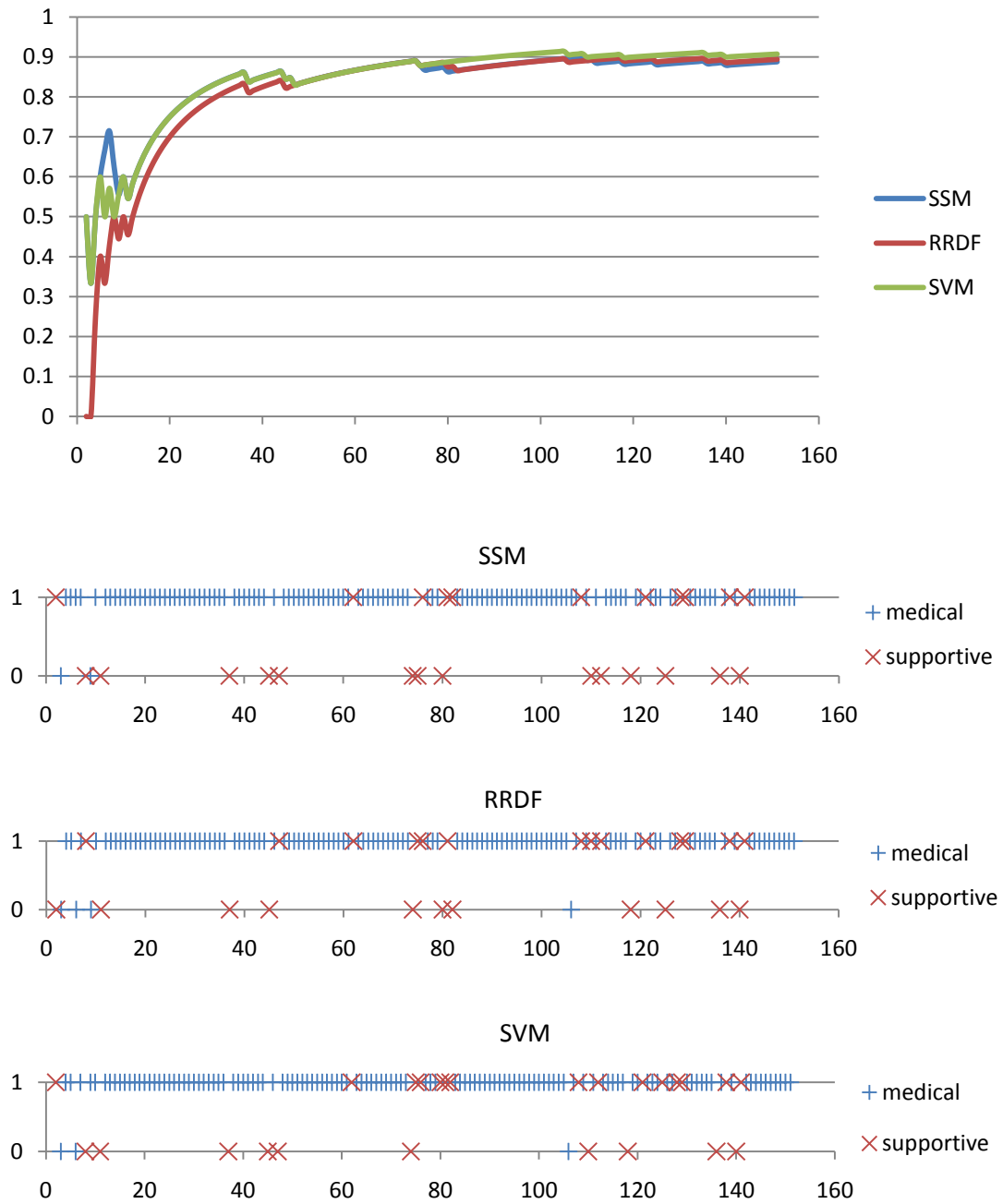
### 5.2.2. Natural Order

In Figure 49 through Figure 52, we plot the accuracy of these three top-performing algorithms (labeled as SSM, RRDF and SVM, respectively) for the natural order protocol. In Figure 49 to Figure 52, part (a) plots the accumulated accuracies as training data is added, and parts (b) to (d) show for each case whether it is correctly or incorrectly classified and labels the class of that case.



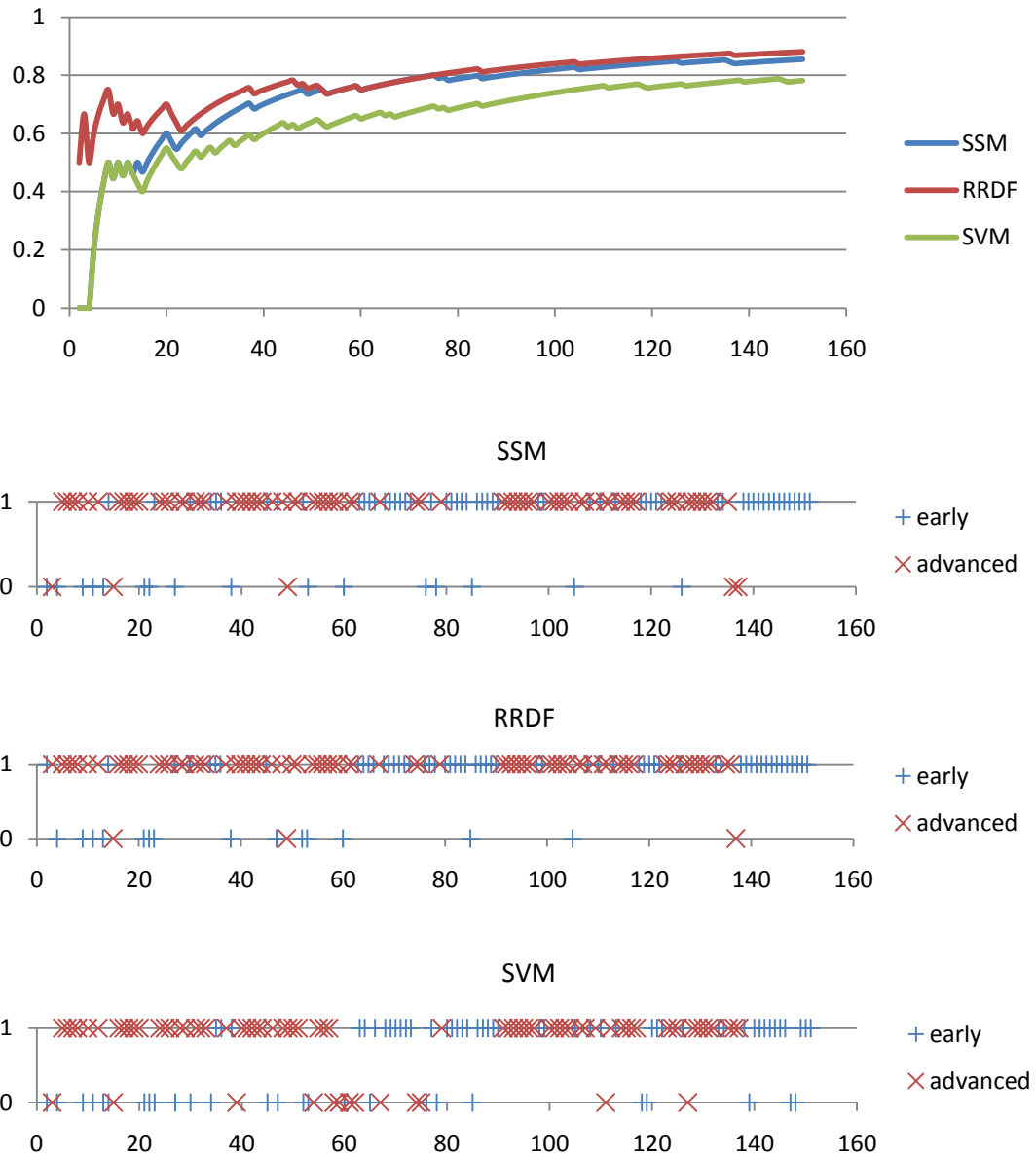
**Figure 49** Accumulated accuracy of earn vs acq in Reuters21578 in natural order. b, c and d) Each test case is classified using SSM, RRDF and SVM: 1 is correct, 0 is incorrect.

## Results



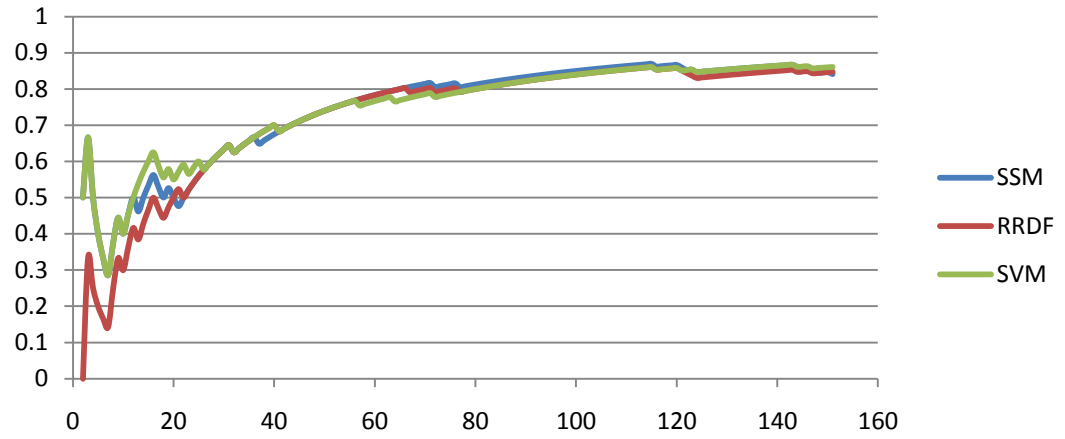
**Figure 50 a) Accumulated accuracy of medical vs supportive in BCKO in natural order. b, c and d) Each test case is classified using SSM, RRDF and SVM: 1 is correct, 0 is incorrect.**

## Results

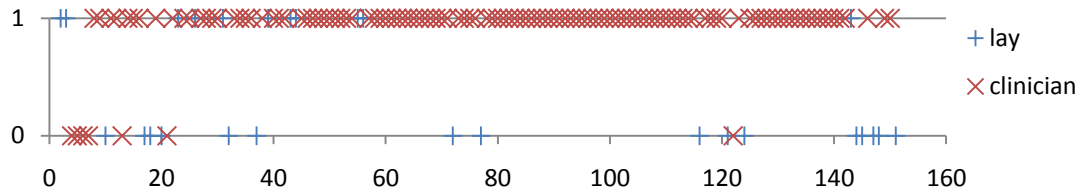


**Figure 51 a) Accumulated accuracy of early vs advanced in BCKO in natural order. b, c and d) Each test case is classified using SSM, RRDF and SVM: 1 is correct, 0 is incorrect.**

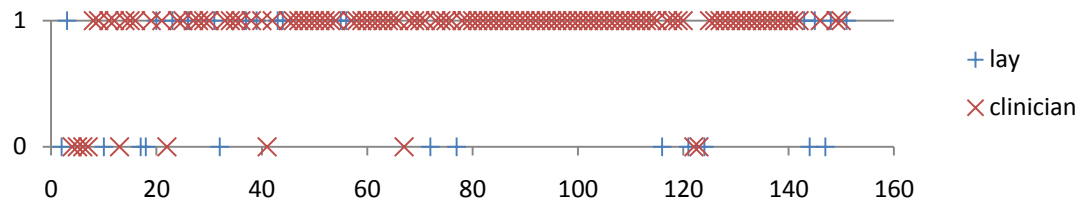
## Results



SSM



RRDF



SVM

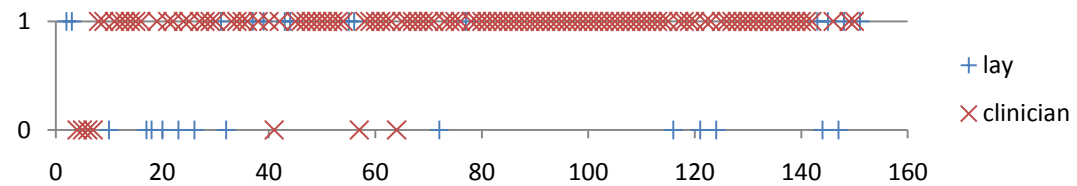
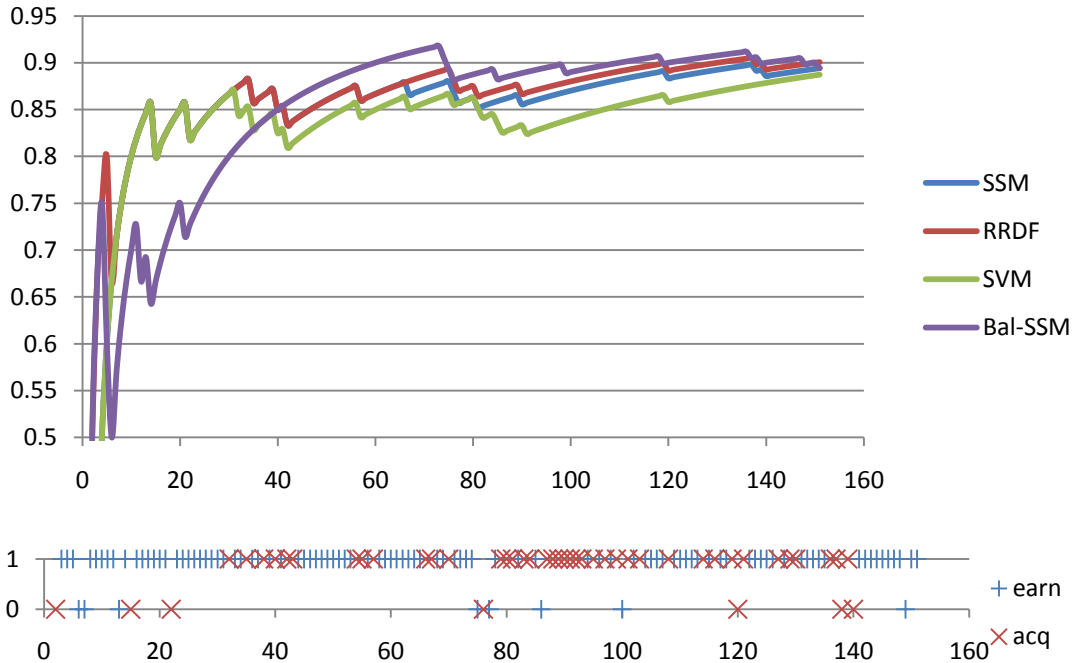


Figure 52 a) Accumulated accuracy of lay vs clinician in BCKO in natural order. b, c and d) Each test case is classified using SSM, RRDF and SVM: 1 is correct, 0 is incorrect.

## Results

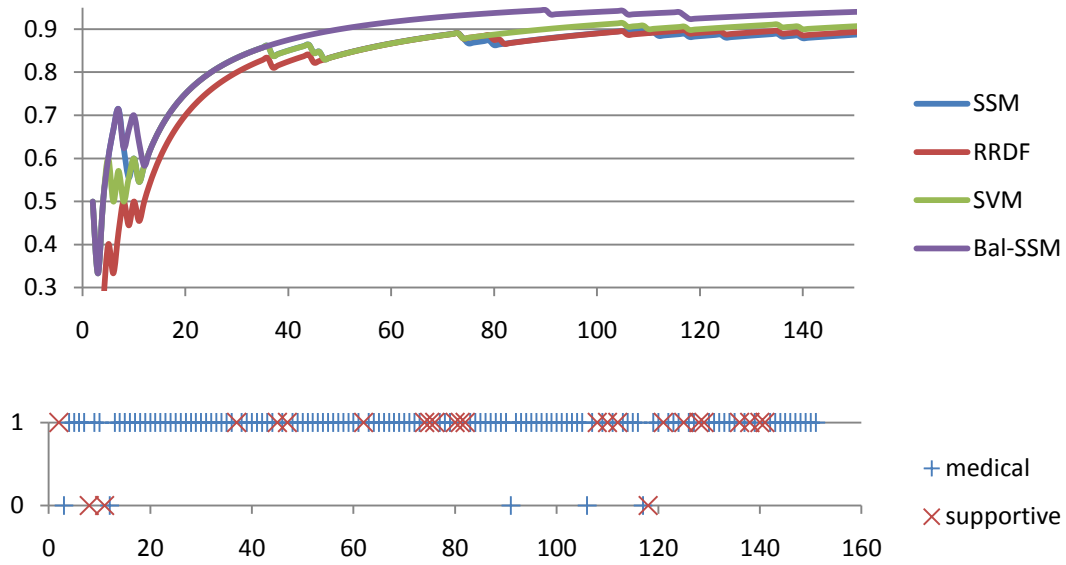
### 5.2.3. Unbalanced Classes

Splitting training data for unbalanced class frequencies and oversampling the less frequent classes yields modified results for the natural order experiments as shown in Figure 53 to Figure 56. It can be clearly seen that after using Balanced SSM (Bal-SSM), the misclassified cases are evenly counterchanged between cases from the opposite classes, while in the original SSM case, the cases from the less-frequent class are much more easily misclassified; also the accumulated accuracies for Reuters21578 and the attributes medical vs. supportive are significantly improved.

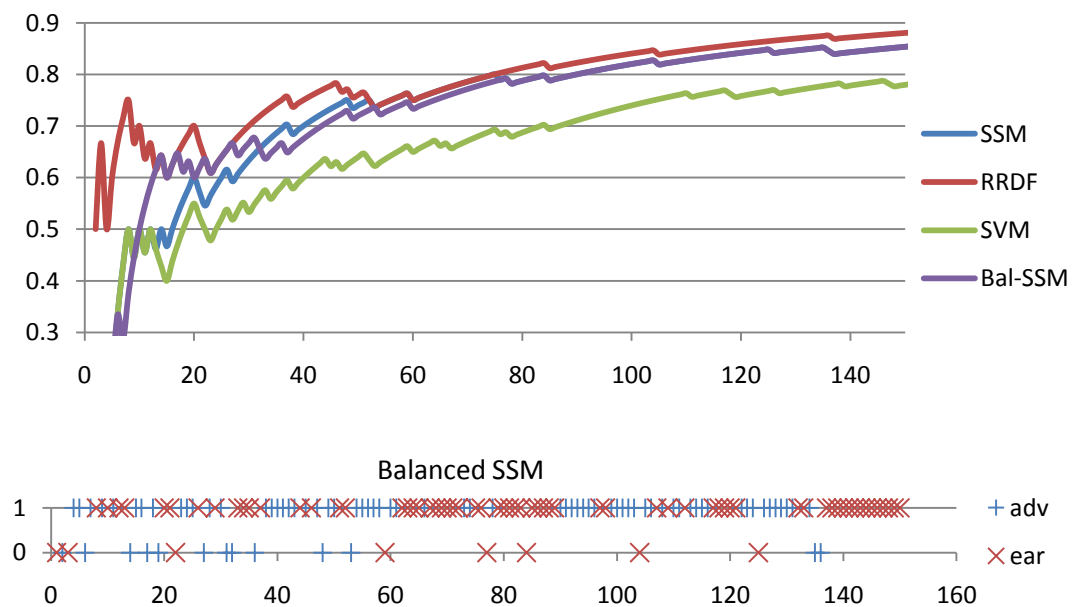


**Figure 53 a) Refined SSM on earn vs. acq in Reuter21578; b) Each test case is classified using refined SSM: 1 is correct, 0 is incorrect.**

## Results



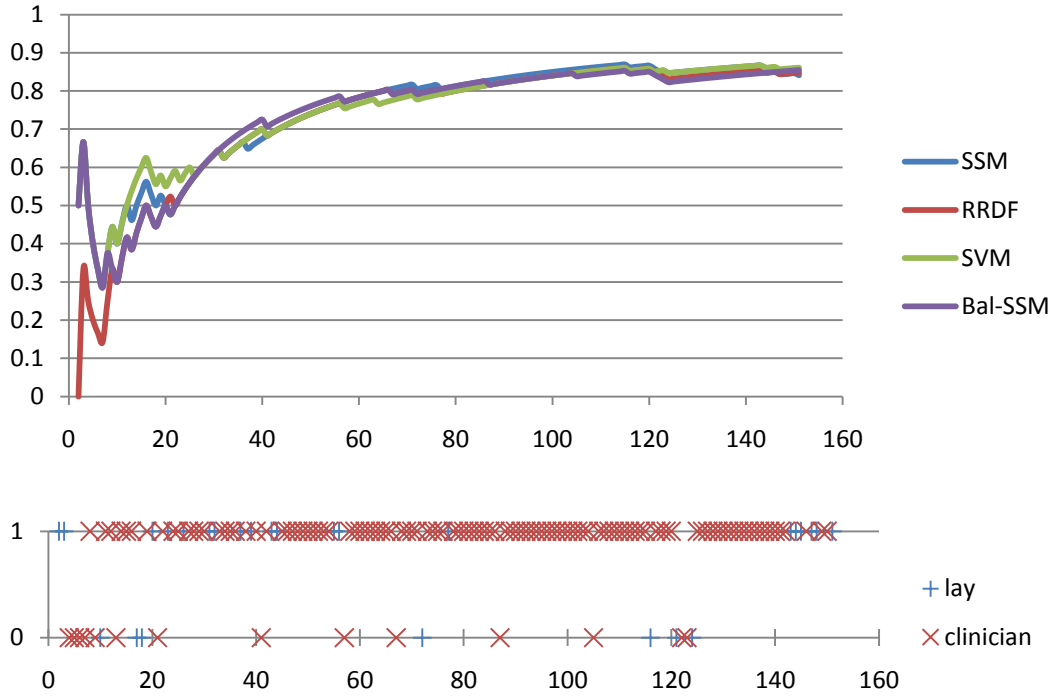
**Figure 54 a) Refined SSM on medical vs. supportive; b) Each test case is classified using refined SSM: 1 is correct, 0 is incorrect.**



**Figure 55 a) Refined SSM for early vs. advanced; b) Each test case is classified using refined SSM: 1 is correct, 0 is incorrect.**



## Results

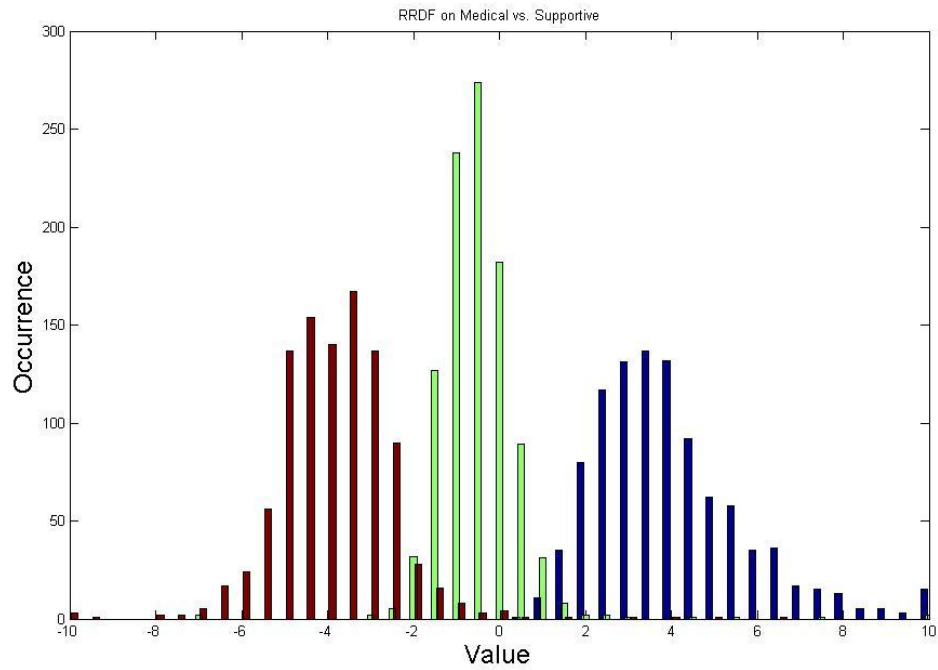


**Figure 56 a) Refined SSM for lay vs. clinician; b) Each test case is classified using refined SSM: 1 is correct, 0 is incorrect.**

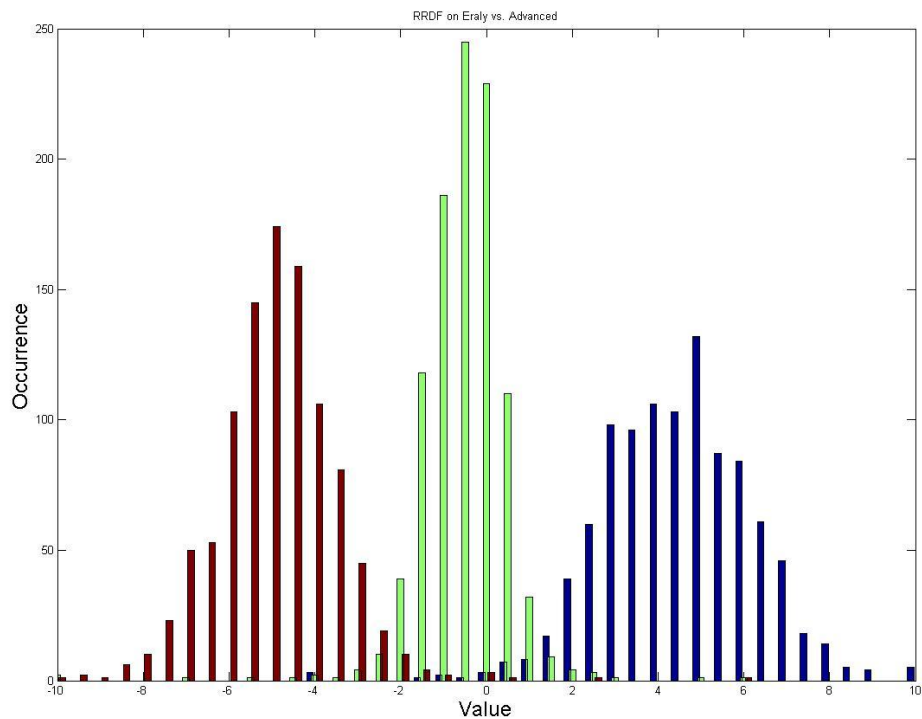
### 5.2.4. Non-Exclusive Classes Classifying

Figure 57 through Figure 60 show the histogram of the result for classification of 1000 test cases, some of which are randomly repeated, each for two Exclusive classes and one Non-Exclusive class using both RRDF and SSM on medical vs. supportive and early vs. advanced. Apparently in Figure 57 and Figure 58 the three groups of testing cases are in perfect normal distribution, thus easy to be separated, while in Figure 59 and Figure 60, although the majority of the cases in the two exclusive classes are far away from the division boundary, a large portion of the cases in the non-exclusive class almost mix with the other two exclusive cases, therefore difficult for separation. For this perspective, RRDF exceeds SSM.

## Results

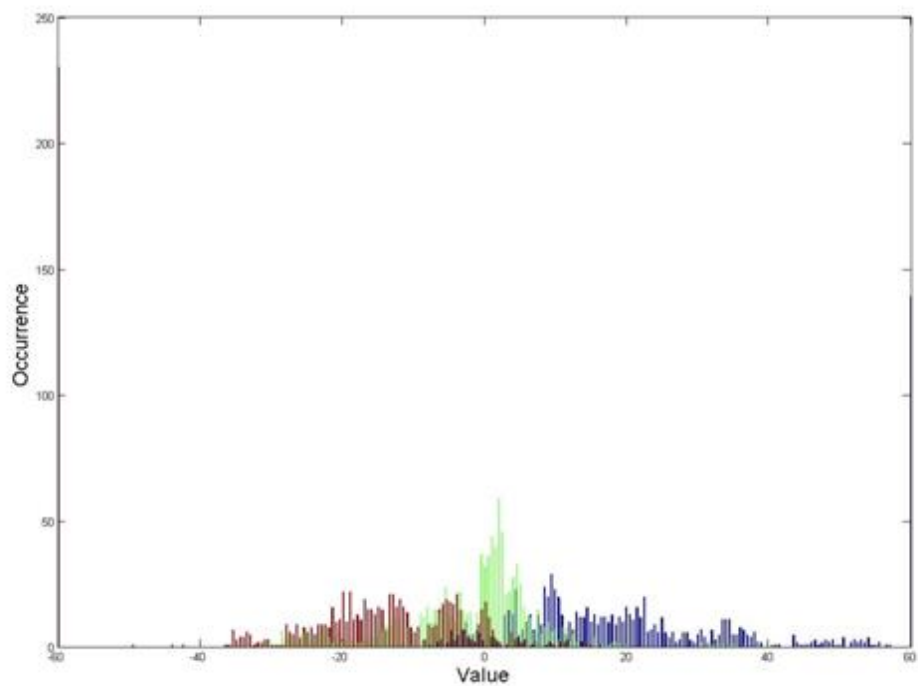


**Figure 57** Plotting of 1000 cases by the difference between the supportive and medical HAL values for RRDF

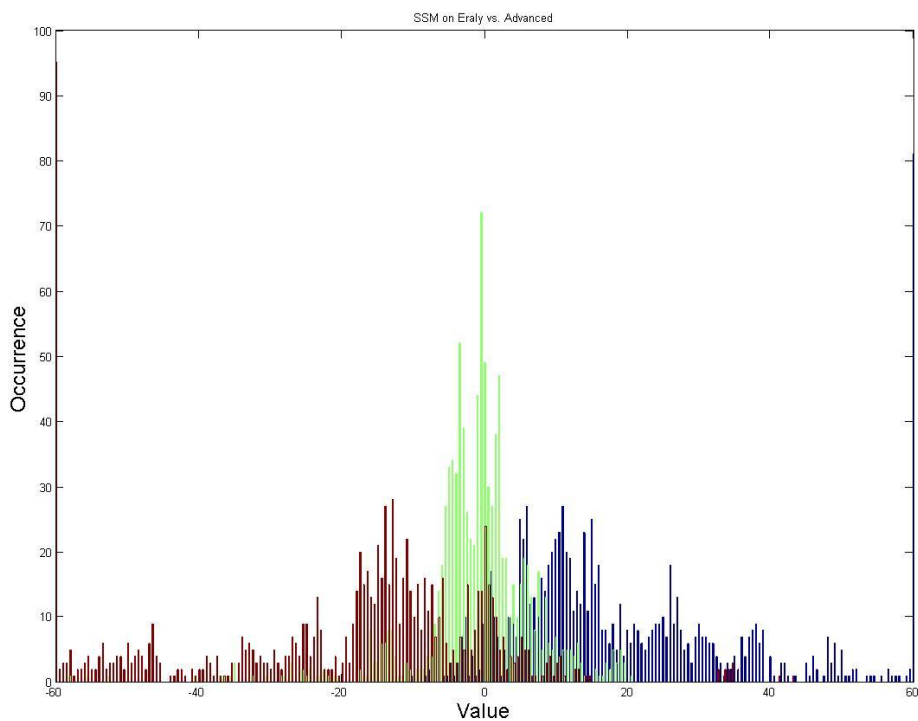


**Figure 58** Plotting of 1000 cases by the difference between the early and advanced HAL values for RRDF

## Results



**Figure 59** Plotting of 1000 cases by the difference between the supportive and medical HAL values for SSM (note there are ~230 occurrences of medical at the left boundary and ~140 occurrence of supportive at the right boundary of the graph)



**Figure 60** Plotting of 1000 cases by the difference between the early and advanced HAL values for SSM (note there are ~95 occurrences of early at the left boundary and >80 occurrence of advanced at the right boundary of the graph)

### 5.3. Computational complexity analysis of SVM and SSM

The amount of time consumed in the training phase is a major consideration for the practicality of a classifier. Conceptually, the size of the HAL matrix is  $N^2$ , where  $N$  is the vocabulary size of the corpus. However, in practice the matrix is sparse, for sparse matrices hash data structure can provide  $O(1)$  performance [145-146], thus  $O(N^2)$  operations can be avoided.

For SSM, the training process has two steps: first, creating the HAL matrix of each web, and second adding the HAL matrices of all the webpages in the training dataset together, grouping by class. Each step has time complexity based on the number of additions to be hash-mapped into the HAL matrices and is  $O(wR\bar{n})$ , where  $w$  is the window size of HAL,  $R$  is the total number of webpages in the training dataset, and  $\bar{n}$  is the mean number of non stop-words on each web page. The time complexity of SVM is reported as  $O(R^2)$ , although it can be optimised to be superior to this in practice [147-148]. Also, I assume the time complexity of SVM has some dependence on the number of features (which is large in our application). As such, the big-O complexity analysis leaves some room for either algorithm to be superior depending on the nature of the data.

For our experiments, using a machine with Intel Core 2 Duo E8400 3GHz processor, 4GB RAM and Microsoft Windows Vista Enterprise 64-bit for 50 cases in each class of two classes in which 40 are training cases and 10 are test cases, SSM takes about 16 minutes for 100 resamples; SVM takes approximately 30 minutes for data preparation (notably, TF\*IDF computation) and 35 minutes for training (in SVMlight). Thus, our experience is that SSM is substantially faster for our data.

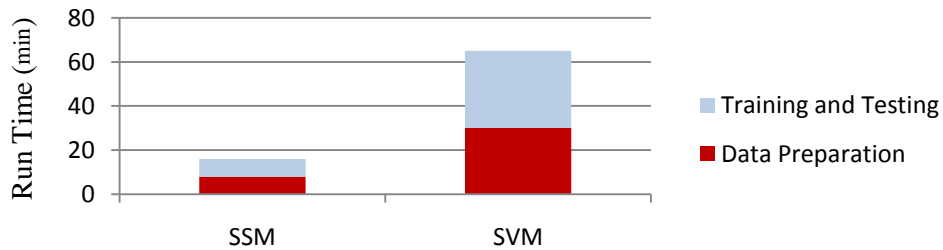


Figure 61 Illustration of the computational complexity of SSM and SVM

## Results

---

### Chapter 6. Discussion

#### 6.1. Significance

We have demonstrated that Semantic Space models based on HAL can provide features that support classification of consumer health webpages on metadata attributes of relevance to consumer health portals. We have achieved good levels of performance (rising to the low 90% range) using a few different classifiers: a novel method based on HAL vector similarity summation (SSM), an adaptation of decision forests for HAL (RRDF), and application of the well-established SVM method to the cells of the HAL matrix (or alternatively to appropriately conditioned word frequencies). We applied these methods to the well-known Reuters21578 data set for comparison. In addition to simulations that randomise training and testing data with balanced numbers of cases from the classes of interest, we have provided results of classification in the natural order. In these experiments we have shown performance as the classifier learns one case at a time in the order the webpages were actually assigned metadata by the consumer website curator. This simulates the accuracy the curator would have experienced had our algorithm been available as part of their metadata encoding tool suite. While the rate of improvement in accuracy varies depending on the metadata attribute in question, and appears to be slightly less with natural order than an ideal balanced mix of cases from each class, we believe the performance is consistent with useful support within the scale of typical consumer health web portal development projects.

Two distinct key user groups for this technology are: (1) ‘curators’ of consumer health information portals (i.e. the staff who maintain the portal, selecting sites to index and choosing values of metadata attributes); and (2) the portal end users (i.e. health consumers searching for information for themselves and family members). From the curator’s point of view, an accurate automated classifier can save them from reading, and making decisions to classify, hundreds and thousands of webpages manually. It can also prevent human errors such as misunderstanding, accidental error, or individual opinions; a computer algorithm, while it may be imperfect in other ways, provides speed, objectivity and consistency. The maintenance of the portals is

burdensome, but vitally important, work. Frequently links become no longer valid or are automatically forwarded to something other than the original content; for example, in the database of the BCKO portal created in Sep 2008, there are 66 valid links out of 129 in the link list for mutually exclusive (ME) supportive web pages and 311 valid ME medical links out of 696 in that link list, revealing that almost half of the links are invalid only a couple of months after their creation. A tool such as introduced in this project will make the work much easier.

From the health consumer's point of view, a reliable, automated, text classifier can make live searching less daunting: it can group the contents, filter or tag them. Furthermore, a good text classifier can help in "digging out the lesser voice." The BCKO team found that breast cancer sufferers wanted to find real-life personal stories, blogs and other such supportive material. Those things, however, are ranked lower by Google and rarely climb up to the first ten pages in Google search results. By grouping or filtering on a particular metadata value (e.g. supportive sites), it could be much easier for the user to find such materials if that is what they were seeking at a particular time.

Spam filtering might be a further use of the text classifiers.

### **6.2. Comparison to related work**

Eysenbach et al [149] present a systematic review on the topic of assessing the quality of online health information for consumers. With analysing seventy-nine studies of health resources, they identify a number of quality criteria commonly used to assess the online contents, including accuracy, completeness, readability, design, disclosures and references. Also they find that quality is a problem on health webpages and that the criteria for measuring quality varies widely in different circumstances. While this 'quality' dimension is of itself quite complex, BCKO and my present work focuses on the 'qualities' of online articles (e.g. tone and disease stage appropriateness) rather than the type of quality that focuses around accuracy per se [140]. My focus on author credential perhaps overlaps the quality interests a la Eysenbach et al. and the interest in supporting user search preference qualities as has characterised the BCKO project.

A recent issue of *Journal of Biomedical Informatics* focused on biomedical text processing and illustrated the wide range of methods and applications currently being

pursued, as well as the powerful tools that are making it possible to achieve increasingly impressive results in this domain [150]. A number of studies address objectives and employ methods similar to the present work. Deshazo and Turner [151] applied rule based processes and SVM to classify diseases based on discharge summary texts. Focusing specifically on webpage classification, Zhang et al. [152] classified Yahoo webpages in the three categories of health, shopping and education; they used word frequency features selected according to information gain, then employed Principal Component Analysis to those features, and finally used a C4.5 decision tree, yielding classification accuracies slightly over 80%. An alternative source of classification features is exploited by Ypma et al. [153] who applied a Markov model to the click-stream log to identify the type of a webpage. Also exploiting the ‘web’ features of webpages, Attardi et al. [154] based their classifier on context, utilising the hyperlinks to the webpages, and the context in the referring webpage. In a similar vein, Roy et al [155] attempt to infer the ‘source’ or ‘sponsor’ of a webpage by way of analysing its incoming and outgoing links. In another case, Mase [156] automatically derives weights for keywords for webpages, which then allows classification in up to 15 distinct categories with accuracy of 70% to 86%. Perhaps most similar to the present work in terms of feature set, Cohen et al [17], employ a Semantic Space model with a permutation based word order encoding [157] that allows remarkably insightful results on MEDLINE data, such as correctly inferring asthma treatments based on the association of the Unified Medical Language System (UMLS) ‘TREATS’ and ‘Asthma’ concepts.

A number of approaches to dimensional reduction have been taken to deal with the large, and sparse, matrices resulting for term-term models. I took a computationally simple approach of picking the  $n$  most frequent columns with highest HAL sum as the reduced matrices. The InfoMap system, on the other hand, uses a two-step process where 1) it picks 1000 most frequent words as ‘content bearing’ words and 2) it applies SVD on the matrix built based on the co-occurrence with the ‘content bearing’ words. A recent approach is Random-Indexing (RI) [158]; this works by projecting a high-dimensional space into a relatively smaller space and thus reduces the dimensionality. It has been shown that RI approximately preserves the distances between vectors after the projection [158].



### 6.3. What we have learned

#### 6.3.1. Feasible

The results support our primary objective of demonstrating the feasibility of classification of relatively subtle attributes of consumer health webpages. That is, beyond determining the topic of a webpage at the level of, say, the disease under discussion, we successfully distinguish the tone of the article, the stage of disease and the credentials of the author.

#### 6.3.2. SSM is an elegant semantic space approach

The SSM algorithm in particular is an elegant utilisation a Semantic Space model, where classification is made based on the context of use of frequent non stop-words without consideration of sentence structure and with a pure machine learning approach.

#### 6.3.3. Semantic Space not demonstrably better on accuracy than other good methods

Conversely, the results do *not* indicate that either a Semantic Space model or our HAL-specific classifiers (SSM and RRDF) are necessary to achieve good results in this domain. In the context of Reuters21578, SVM has been shown previously to be a good classifier [55]. Our results on consumer health webpages show that SVM (using either HAL features or conditioned word frequencies) is a fast learner in most of the experiments we tried when the number of training cases is small. With more training cases, the results of SVM converge with SSM and RRDF.

#### 6.3.4. SSM gives the implementer a new choice with different performance characteristic to SVM

Careful choice of kernel function and tuning of parameters may yield a small improvement in performance for this method; overall, however, we believe, other criteria, such as trade-off of computational space and time for specific applications, may reasonably decide an implementer's choice among these algorithms for the context we have explored. Empirically, we find SSM to be faster than SVM, but acknowledge the difficulties of comparing performance of a generic third party

algorithm to one we have implemented specific to the purposes of this study. According to our computational complexity analysis, SSM should give better performance for large training set sizes, an important and increasingly common case with the continued growth of the web.

### **6.3.5. Less frequent classes**

The BCKO data show that often very imbalanced frequencies of training data are available with respect to different values of a metadata attribute (e.g. we have 311 medical cases and only 66 supportive cases). In this case, the shuffle splitting/oversampling method described in Section 3.3.6 improves performance as shown by the result, in Section 5.2.3.

### **6.3.6. Non-exclusive classes**

In the BCKO database, many of the data attributes allow non-exclusive membership (i.e. the webpages belong to more than one group). For that kind of data, I have attempted flagging the non-exclusive membership as a new class, testing it with the classifier that is trained using the exclusive-only dataset. The result looks promising in terms of separating the non-exclusively coded cases from those exclusively coded; the non-exclusive cases tend to have an SSM score that is intermediate between the two exclusive cases.

The test result shows that RRDF is better tool to deal with non-exclusive classes.

## **6.4. Limitation**

There are many potential barriers, even with the best possible classifiers, with respect to achieving near-perfect classification on consumer metadata attributes using Semantic Space models. These barriers include quotation, enantiosis and drollness in the text, as well as legitimate difference of opinion as to the class of a specific article with respect to subjective aspects such as medical versus supportive article tone. We have restricted presentation in this thesis to 2-class problems. The methods have no inherent limitation in terms of the number of distinct classes; however, unsurprisingly, based on results from Reuters21578 [55] (as well as our own experiences not presented in the results of this article), the larger the number of classes, the more difficult the problem and the lower the accuracy. It is interesting to note that the

difficulty of our consumer health metadata classification problems appears not very different from that of classification in the Reuters data (with the data representations we have chosen).

### **6.5. Conclusion and future direction**

When confronted with a healthcare situation, people are increasingly turning to the Internet for information to aid in understanding diagnoses, deciding on treatment options and seeking psychosocial support for themselves, their family and their friends [1]. Vast quantities of health information are being made available online from a wide range of sources (government agencies, pharmaceutical companies, commercial companies, charity organisations, community groups and individuals). As a result a keyword search using any of the major search engines on most healthcare topics will bring up an unusably large number of hits of varying quality and relevance to a person's particular health and life situation. As I stated in the Introduction, the resulting information overload, where the amount of information exceeds a person's ability to process it [6], can often add stress to an already stressful situation. Consequently there is much concern regarding how the quality, relevance, authority and accuracy of online information can be assessed in a timely manner by both healthcare consumers and medical professionals alike[8, 149].

Many projects have been devised to address information overload and investigate ways in which timely, differentiated access to quality online healthcare resources can be provided. The provision of web portals, centred on particular health topics and/or communities of users, is one such strategy [5-6]. The aim is to provide access to a reduced, more manageably sized corpus of information resources that meet quality and relevance criteria. Portals can be further augmented by capturing and creating descriptive metadata about resources selected for inclusion. This structured, value-added information can then be used by portal users in searching, filtering, ranking, and in making judgements about what information is relevant to their needs and in which they wish to trust.

In this thesis, I used metadata from one such topic-centred consumer health portal: Breast Cancer Knowledge Online Portal (BCKO), developed through collaboration

## Discussion

---

between Monash University, BreastCare Victoria and the Breast Cancer Action Group. The key aim of this thesis was to determine the extent to which I could emulate the metadata classifications that had been painstakingly formulated manually for BCKO's repository, using automatic methods, and thus address the problem of automatically determining metadata attributes for consumer health websites.

Motivated by the results of Burgess and Lund [19] I applied the HAL semantic space model as a source of classification features. I addressed the classification problem by using a decision tree, then extending the solution by creating a decision forest. After carefully analyzing the intermediate processing data and the final output, and considering the nature of the HAL model, I improved the decision forest by summing the similarities to classes for the words in the validation path of individual decision trees instead of using the trees in the traditional manner (i.e. by having trees 'vote' an outcome); this improved performance. Finally, based on my work up to that point on a Round Robin Decision Forest (RRDF) I invented the novel algorithm Summed Similarity Measurement (SSM), which provides both a better solution and very simple implementation. I have achieved an overall performance of over 90% accuracy with a modest amount of training data (on the order of 40 cases per class). The algorithm has similar classification accuracy to SVM and takes less processor time; it also provides an easier implementation of the text classifier. In a realistic application environment, at least to support a metadata coder, the classifier is likely to be handed cases one by one in order, as tested by the natural order experiments I have conducted. For the less frequent classes/unbalance issue, as a variant on the natural order experiment, I have used an oversampling method, the experiment showing it to have good applicability to this problem.

As we reviewed in Section 2.4, in the domain of the health consumer informatics, providing more personalised information and service is the aim. A proficient classifier can work with most of those applications; for instance, a well-trained classifier could be very helpful for building and maintaining a healthcare portal like BCKO; a good classifier is essential for a content-based recommender system; a computer-tailoring intervention also needs a effective algorithm to best match the user's requirements. For instance, in the CHESS system [133], our classifier could help to support for the social support component.

## Discussion

---

A Java application programming interface (API) for the SSM and RRDF algorithms is available from <http://www.cs.auckland.ac.nz/~gchen/api>. Please cite this thesis and the URL if using the API in research work.

There is no reason why the methods from this thesis should not port readily to other languages. We are exploring a version of the program to deal with Chinese language documents.

### References

1. Kralik D, van Loon A, Telford K. Transitions in Chronic Illness: Booklet 11: Understanding Transition. University of South Australia/Royal District Nursing Service. 2005.
2. Chen G, Warren J, McArthur R, Bruza P, Kralik D, Price K. Understanding Individual Experiences of Chronic Illness with Semantic Space Models of Electronic Discussions. Proceedings of the Twentieth IEEE International Symposium on Computer-Based Medical Systems: IEEE Computer Society; 2007. p. 548-6.
3. Madden M, Fox S. Finding answers online in sickness and in health. Pew Internet and American Life Project, [http://www.pewinternet.org/pdfs/PIP\\_Health\\_Decisions\\_2006pdf](http://www.pewinternet.org/pdfs/PIP_Health_Decisions_2006pdf). 2006. [accessed 2010 Jan 26].
4. Eysenbach G. Consumer health informatics. British Medical Journal. 2000;06/23 ed2000. p. 1713-6.
5. Bomba D. Evaluating the Quality of Health Web Sites: Developing a Validation Method and Rating Instrument. Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 6 - Volume 06: IEEE Computer Society; 2005. p. 139.1.
6. Kim K, Lustria M, Burke D, Kwon N. Predictors of cancer information overload: findings from a national survey. Information Research. 2007;12(4):12-4.
7. Eysenbach G, Powell J, Kuss O, Sa E. Empirical studies assessing the quality of health information for consumers on the world wide web a systematic review. JAMA: Am Med Assoc; 2002. p. 2691-10.
8. Haynes R, Cotoi C, Holland J, Walters L, Wilczynski N, Jedraszewski D, et al. Second-order peer review of the medical literature for clinical practitioners. JAMA. 2006;295(15):1801-8.
9. Moon J, Burstein F. Intelligent Portals for Supporting Medical. Web portals: The new gateways to Internet information and services: IGI Global; 2005. p. 270 - 20.
10. Madden AD. Portals or filters? Identifying quality on the Internet. In: Cox. A, editor. Portals: People, Processes and Technology. London: Facet; 2006.
11. Burstein F, Fisher J, McKemmish S, Manaszewicz R, Malhotra P. User Centred Quality Health Information Provision: Benefits and Challenges. Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 6 - Volume 06: IEEE Computer Society; 2005. p. 138.3.
12. Hunter J. Next Generation Tools and Services: Supporting Dynamic Knowledge Spaces. In: Kapitzke C, Bruce BC, editors. Lib@ ries: Changing Information Space and Practice: Lawrence Erlbaum Associates Inc.; 2006. p. 91 - 22.
13. Huettig F, Quinlan P, McDonald S, Altmann G. Models of high-dimensional semantic space predict language-mediated eye movements in the visual world. Acta psychologica. 2006;121(1):65-80.
14. McDonald S. Environmental determinants of lexical processing effort. Unpublished PhD dissertation, University of Edinburgh. 2000.
15. Cohen T, Widdows D. Empirical distributional semantics: Methods and biomedical applications. Journal of Biomedical Informatics. 2009;42(2):390-405.
16. Widdows D, Ferraro K, Semantic vectors: a scalable open source package and online technology management application. Sixth International Conference on Language Resources and Evaluation (LREC 2008); 2008.

## References

---

17. Cohen T, Schvaneveldt R, Rindfleisch T, Predication-based semantic indexing: Permutations as a means to encode predications in semantic space. Proc AMIA Annual Symposium; 2009: American Medical Informatics Association.
18. Burgess C, Livesay K, Lund K. Explorations in Context Space: Words, Sentences, Discourse. Discourse Processes 1998. p. 211- 47.
19. Burgess C, Lund K. Representing abstract words and emotional connotation in a high-dimensional memory space. Cognitive Science Proceedings, LEA [http://haluc.edu/pdfs/Burgess\\_Lund\(1997b\).pdf](http://haluc.edu/pdfs/Burgess_Lund(1997b).pdf); Lawrence Erlbaum Associates; 1997. [accessed 2010 Jan 26]. p. 61 - 6.
20. Burgess C, Lund K. Modelling parsing constraints with high-dimensional context space. Language and Cognitive Processes: Psychology Press; 1997. p. 177-210.
21. Sahlgren M. The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces: Department of Linguistics, Stockholm University; 2006.
22. Joachims T. Making large-scale SVM learning practical. In: Scholkopf B, Burges C, Smola A, editors. Advances in Kernel Methods – Support Vector Learning. Cambridge, MA: The MIT Press; 1999. p. 169-84.
23. Mitchell T. Decision tree learning. Machine learning. 1997;414.
24. ArzucanOzgur L, Gungor T, Text categorization with class-based and corpus-based keyword selection 2005: Springer.
25. Krause E. Taxicab Geometry. Mathematics Teacher. 1973;66(8):695-706.
26. Garcia E. Cosine Similarity and Term Weight Tutorial. Retrieved May 2006. p. 2009.
27. Rohde D, Gonnerman L, Plaut D. An improved method for deriving word meaning from lexical co-occurrence. Cognitive Science. 2004.
28. Alpaydin E. Introduction to machine learning: The MIT Press; 2004.
29. Zhang R, Shepherd M, Duffy J, Watters C, Automatic Web Page Categorization using Principal Component Analysis 2007.
30. Chen J, Tsai C, Young J, Kodell R. Classification ensembles for unbalanced class sizes in predictive toxicology. SAR and QSAR in Environmental Research: Taylor & Francis; 2005. p. 517-13.
31. Fung G, Yu J, Wang H, Cheung D, Liu H. A Balanced Ensemble Approach to Weighting Classifiers for Text Classification. Citeseer; 2006. p. 869-73.
32. Dewdney N, VanEss-Dykema C, MacMillan R. The form is the substance: Classification of genres in text. Association for Computational Linguistics; 2001. p. 7.
33. Cohen W, Singer Y. Context-sensitive learning methods for text categorization. ACM Transactions on Information Systems (TOIS): ACM New York, NY, USA; 1999. p. 141-73.
34. Efron B. The jackknife, the bootstrap and other resampling plans. Siam; 1982.
35. Kohavi R, A study of cross-validation and bootstrap for accuracy estimation and model selection 1995: In: Proc. IJCAI-95, Montreal, Quebec (1995), pp. 1137–1143.
36. Picard R, Cook R. Cross-validation of regression models. Journal of the American Statistical Association. 1984;79(387):575-83.
37. Ho T, Bootstrapping text recognition from stop words 1998: Procs. ICPR-14, Brisbane 1998, 605-609.
38. Ho T, Fast identification of stop words for font learning and keyword spotting 1999: Procs. ICDAR-5, Bangalore 1999, 333-336.



## References

---

39. Ho T. Stop word location and identification for adaptive text recognition. *International Journal on Document Analysis and Recognition*. 2000;3(1):16-26.
40. Silva C, Ribeiro B, The importance of stop word removal on recall values in text categorization 2003.
41. Lewis D, Yang Y, Rose T, Li F. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*. 2004;5:361-97.
42. Greenberg J. Understanding metadata and metadata schemes. *Cataloging & classification quarterly*. 2005;40(3):17-36.
43. Harvey D, Hider P. Organising knowledge in a global society: principles and practice in libraries and information centres: Centre for Information Studies, Charles Sturt University, Wagga Wagga, NSW. .
44. Lassila O, Swick R. Resource description framework (RDF) model and syntax Specification. World Wide Web Consortium, <http://www.w3.org/TR/WD-rdf-syntax>. (W3C Recommendation 22 February 1999).
45. Klyne G, Carroll J, McBride B. Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation. 2004;10.
46. Miller E. An introduction to the resource description framework. *Journal of Library Administration*. 2001;34(3):245-55.
47. Bharati A, Chaitanya V, Sangal R, Ramakrishnamacharyulu K. *Natural Language Processing: PHI*; 2000.
48. Allen J. *Natural language processing*. John Wiley and Sons Ltd.; 2003. p. 1218-22.
49. Joshi A. *Natural language processing*. Science. 1991;253(5025):1242.
50. Jurafsky D, Martin J, Kehler A. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*: MIT Press; 2000.
51. Pereira F, Grosz B. *Natural language processing*: MIT Press Cambridge, MA, USA; 1994.
52. Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine\* 1. *Computer networks and ISDN systems*. 1998;30(1-7):107-17.
53. Salton G. *Automatic text processing: the transformation. Analysis and Retrieval of Information by Computer*. 1989(Addison-Wesley Publishing Co., Reading, MA, 1989).
54. Aizawa A. An information-theoretic perspective of tf-idf measures\* 1. *Information Processing & Management*. 2003;39(1):45-65.
55. Debole F, Sebastiani F. An analysis of the relative hardness of Reuters-21578 subsets. *Journal of the American Society for Information Science and technology*: Wiley Subscription Services, Inc., A Wiley Company Hoboken; 2005. p. 584-13.
56. Salton G. *Automatic text processing: the transformation. Analysis and Retrieval of Information by Computer*. 1989.
57. Miller D, Leek T, Schwartz R, A hidden Markov model information retrieval system. *Proceedings of SIGIR Conference*; 1999: ACM
58. Landauer T, Foltz P, Laham D. An introduction to latent semantic analysis. *Discourse Processes*: ALEX PUBLISHING CO; 1998. p. 259-26.
59. McArthur R, Bruza P, Warren J, Kralik D, Projecting computational sense of self: A study of transition in a chronic illness online community. *Proceedings of the 39th Hawaii International Conference on System Sciences (HICSS-39)*; 2006.
60. McArthur R, Bruza P, Discovery of implicit and explicit connections between people using email utterance. *Proceedings of the Eighth European Conference of*



## References

---

- Computer-supported Cooperative Work; 2003: Kluwer Academic Publishers, Helsinki, pp. 21–40.
61. Mc Arthur R, Bruza P. Discovery of tacit knowledge and topical ebbs and flows within the utterances of online community. *Chance Discovery*.
  62. Landauer T. *Handbook of latent semantic analysis*: Lawrence Erlbaum; 2007.
  63. Vanteru B, Shaik J, Yeasin M. Semantically linking and browsing PubMed abstracts with gene ontology. *BMC genomics*. 2008;9(Suppl 1):S10.
  64. Chute C, Yang Y, Evans D, Latent Semantic Indexing of medical diagnoses using UMLS semantic structures1991: American Medical Informatics Association.
  65. Elvevåg B, Foltz P, Weinberger D, Goldberg T. Quantifying incoherence in speech: An automated methodology and novel application to schizophrenia. *Schizophrenia research*. 2007;93(1-3):304-16.
  66. Hofmann T, Probabilistic latent semantic analysis1999: Citeseer.
  67. Blei D, Ng A, Jordan M. Latent dirichlet allocation. *The Journal of Machine Learning Research*. 2003;3:993-1022.
  68. Sethuraman J. A constructive definition of Dirichlet priors. *Statistica Sinica*. 1994;4(2):639-50.
  69. Hofmann T. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*. 2001;42(1):177-96.
  70. Bó I, Siklósi D, Szabó J, Benczúr A, Linked latent Dirichlet allocation in web spam filtering2009: ACM.
  71. Hofmann T, Collaborative filtering via gaussian probabilistic latent semantic analysis2003: ACM.
  72. Ginter F, Pahikkala S. *Advances in Natural Language Processing*: Springer-Verlag Berlin/Heidelberg; 2006.
  73. Goldwater S. *Natural Language Understanding*, 09-10.
  74. Chen Y, Tsai F, Chan K. Machine learning techniques for business blog search and mining. *Expert Systems with Applications*. 2008;35(3):581-90.
  75. Lund K, Burgess C, Atchley R, Semantic and associative priming in high-dimensional semantic space. *Proceedings of the Cognitive Science Society (1995)*, ; 1995: pp. 660–665.
  76. Mirman D, Magnuson J, The impact of semantic neighborhood density on semantic access. *Proceedings of the 28th Annual Cognitive Science Society Meeting*; 2006: ( pp. 1823–1828). Mahwah, NJ: Erlbaum.
  77. Quinlan J. Discovering rules from large collections of examples: a case study. *Expert Systems in the Micro-electronic Age*. 1979:168-201.
  78. Breiman L. Bagging predictors. *Machine learning*. 1996;24(2):123-40.
  79. Quinlan J, Bagging, boosting, and C4. 5. *Proceedings of the Thirteenth National Conference on Artificial Intelligence*; 1996: pages 725-730.
  80. Ho T. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1998;20(8):832-44.
  81. Gao Q, Wang Z. Center-based nearest neighbor classifier. *Pattern Recognition*. 2007;40(1):346-9.
  82. Li S, Lu J. Face recognition using the nearest feature line method. *Neural Networks, IEEE Transactions on*. 2002;10(2):439-43.
  83. Vincent P, Bengio Y. K-local hyperplane and convex distance nearest neighbor algorithms. *Advances in Neural Information Processing Systems*. 2002;2:985-92.
  84. Zheng W, Zhao L, Zou C. Locally nearest neighbor classifiers for pattern classification. *Pattern Recognition*. 2004;37(6):1307-9.

## References

---

85. Yang T, Kecman V. Adaptive local hyperplane classification. *Neurocomputing*. 2008;71(13-15):3001-4.
86. Kononenko I. Estimating attributes: Analysis and extensions of RELIEF. *Lecture Notes in Computer Science*. 1994:171-.
87. Vapnik V. The nature of statistical learning theory: Springer Verlag; 2000.
88. Chapelle O, Vapnik V. Model selection for support vector machines. *Advances in Neural Information Processing Systems*. 1999;12:230–6.
89. Hearst M, Dumais S, Osman E, Platt J, Scholkopf B. Support vector machines. *IEEE Intelligent systems*. 1998;13(4):18-28.
90. Steinwart I, Christmann A. Support Vector Machines: Springer Verlag; 2008.
91. Huang T, Kecman V, Kopriva I. Kernel based algorithms for mining huge data sets: supervised, semi-supervised, and unsupervised learning: Springer Verlag; 2006.
92. Dumais S. Using SVMs for text categorization. *IEEE Intelligent systems* 1998;13(4):21-3
93. Joachims T, Nedellec C, Rouveirol C, Text categorization with support vector machines: learning with many relevant 1998: Springer.
94. Weston J. Multi-class support vector machines. *Proceedings of ESANN99???*, D. 1998.
95. Weston J, Watkins C, Support vector machines for multi-class pattern recognition. *Fourth International conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*; 1999; Brighton, UK.
96. Cristianini N, Shawe-Taylor J. An introduction to support Vector Machines: and other kernel-based learning methods: Cambridge Univ Pr; 2000.
97. Schölkopf B, Burges C, Smola A. Introduction to support vector learning: MIT Press; 1999.
98. Joachims T. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*. 1998:137-42.
99. Byvatov E, Schneider G. Support vector machine applications in bioinformatics. *Applied bioinformatics*. 2003;2(2):67.
100. Bhasin M, Raghava G. SVM based method for predicting HLA-DRB1\* 0401 binding peptides in an antigen sequence. *Bioinformatics*. 2004:4241.
101. Guo J, Chen H, Sun Z, Lin Y. A novel method for protein secondary structure prediction using dual-layer SVM and profiles. *PROTEINS: Structure, Function, and Bioinformatics*. 2004;54(4):738-43.
102. Miranda E. Site Dependent Strength Reduction Factors. *Journal of Structural Engineering*. 1993;119:3503.
103. Joachims T, A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *International Conference on Machine Learning (ICML)*; 1997: 143-151.
104. Schapire R, Singer Y, Singhal A, Boosting and Rocchio applied to text filtering 1998: ACM.
105. Sebastiani F. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*. 2002;34(1):1-47.
106. Dumais S, Platt J, Heckerman D, Sahami M, Inductive learning algorithms and representations for text categorization 1998: ACM.
107. Lewis D, Evaluating text categorization 1991.
108. Androutsopoulos I, Koutsias J, Chandrinos K, Paliouras G, Spyropoulos C, An evaluation of naive bayesian anti-spam filtering 2000: Citeseer.
109. Drucker H, Wu D, Vapnik V. Support vector machines for spam categorization. *IEEE Transactions on Neural networks*. 1999;10(5):1048-54.

## References

---

110. Sakis G, Androutsopoulos I, Paliouras G, Karkaletsis V, Spyropoulos C, Stamatopoulos P. Stacking classifiers for anti-spam filtering of e-mail. Arxiv preprint cs/0106040. 2001.
111. Lewis D. Reuters-21578 text categorization test collection. AT&T Labs Research. 1997.
112. Johnson D, Oles F, Zhang T, Goetz T. A decision-tree-based symbolic rule induction system for text categorization-References. IBM Systems Journal 2002.
113. Manaszewicz R, Williamson K, McKemmish S. Breast cancer knowledge online: Towards meeting the diverse information needs of the breast cancer community. Proceedings of the Electronic Networking-Building Community. 2002.
114. Malhotra P, Burstein F, Fisher J, McKemmish S, Anderson J, Manaszewicz R. Breast cancer knowledge on line portal: An intelligent decision support system perspective 2003: Citeseer.
115. Resnick P, Varian H. Recommender systems. Communications of the ACM. 1997;40(3):58.
116. Perugini S, Gonçalves M, Fox E. Recommender systems research: A connection-centric survey. Journal of Intelligent Information Systems. 2004;23(2):107-43.
117. Konstan J. Introduction to recommender systems: Algorithms and evaluation. ACM Transactions on Information Systems (TOIS). 2004;22(1):1-4.
118. Kumar R, Raghavan P, Rajagopalan S, Tomkins A. Recommendation systems: A probabilistic analysis. Journal of Computer and System Sciences. 2001;63(1):42-61.
119. McDonald D, Ackerman M. Expertise recommender: a flexible recommendation system and architecture 2000: ACM New York, NY, USA.
120. Schafer J, Frankowski D, Herlocker J, Sen S. Collaborative filtering recommender systems. Lecture Notes in Computer Science. 2007;4321:291.
121. Pazzani M, Billsus D. Content-based recommendation systems. Lecture Notes in Computer Science. 2007;4321:325.
122. Burke R. Knowledge-based recommender systems. Encyclopedia of Library and Information Systems. 2000;69(Supplement 32):175-86.
123. Foner L, Yenta: a multi-agent, referral-based matchmaking system 1997: ACM.
124. Tang T, McCalla G. Smart recommendation for an evolving e-learning system. International Journal on E-learning. 2005;4(1):105-29.
125. Kurapati K, Gutta S, Schaffer D, Martino J, Zimmerman J. A multi-agent TV recommender. Workshop on Personalization in Future TV, User Modeling; 2001: Sonthofen, Germany.
126. De Vries H, Brug J. Computer-tailored interventions motivating people to adopt health promoting behaviours: introduction to a new approach. Patient Educ Couns. 1999;36(2):99-105.
127. Dijkstra A, De Vries H. The development of computer-generated tailored interventions. Patient Education and Counseling. 1999;36(2):193-203.
128. Jones B, Abidi S, Ying W. Using computerized clinical practice guidelines to generate tailored patient education materials 2005.
129. Davis S, Abidi S. Tailoring Cardiovascular Risk Management Educational Interventions: A Synergy of SCORE Risk Assessment and Behaviour Change Model.
130. Campbell M, Honess-Morreale L, Farrell D, Carbone E, Brasure M. A tailored multimedia nutrition education pilot program for low-income women receiving food assistance. Health Education Research. 1999;14(2):257.

## References

---

131. Strecher V, Shiffman S, West R. Randomized controlled trial of a web-based computer-tailored smoking cessation program as a supplement to nicotine patch therapy. *Ann Arbor*. 2005;1001:48109-0471.
132. Spittaels H, De Bourdeaudhuij I, Vandelanotte C. Evaluation of a website-delivered computer-tailored intervention for increasing physical activity in the general population. *Preventive medicine*. 2007;44(3):209-17.
133. McTavish F, Gustafson D, Owens B, Wise M, Taylor J, Apantaku F, et al., CHES: An interactive computer system for women with breast cancer piloted with an under-served population 1994: American Medical Informatics Association.
134. Ma C, Warren J, Phillips P, Stanek J. Empowering patients with essential information and communication support in the context of diabetes. *International Journal of Medical Informatics*. 2006;75(8):577-96.
135. Koch T, Kralik D, Van Loon A, Mann S. Participatory action research in health care: Blackwell Pub; 2006.
136. Kralik D, MRCNA M. The quest for ordinariness: transition experienced by midlife women living with chronic illness. *Journal of Advanced Nursing*. 2002;39(2):146.
137. Kralik D, Price K, Warren J, Koch T. Issues in data generation using email group conversations for nursing research. *Journal of Advanced Nursing*. 2006;53(2):213.
138. Kralik D, Van Loon A, Visentin K. Resilience in the chronic illness experience. *Educational Action Research*. 2006;14(2):187-201.
139. Chen G, Warren J, Evans J. Automatically generated consumer health metadata using semantic spaces. *Proceedings of the second Australasian workshop on Health data and knowledge management - Volume 80*. Wollongong, NSW, Australia: Australian Computer Society, Inc.; 2008. p. 9-15.
140. Chen G, Warren J, Evans J. 'Qualities' not 'Quality'—Text Analysis Methods to Classify Consumer Health Websites. *electronic Journal of Health Informatics*. 2009;4(1):e5.
141. Sahlgren M, Cöster R. Using bag-of-concepts to improve the performance of support vector machines in text categorization. *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*; 2004: Association for Computational Linguistics.
142. Ling C, Li C. Data mining for direct marketing: Problems and solutions. *International Conference on Knowledge Discovery Data Mining*; 1998: AAAI Press, Menlo Park, CA, pp 73-79.
143. Chen G, Warren J, Yang T, Kecman V. Adaptive K-Local Hyperplane (AKLH) Classifiers on Semantic Spaces to Determine Health Consumer Webpage Metadata. *Proceedings of the 2008 21st IEEE International Symposium on Computer-Based Medical Systems - Volume 00*: IEEE Computer Society; 2008. p. 287-9.
144. Lund K, Burgess C. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments & Computers*. 1996;28(2):203-8.
145. Pagh R. Low redundancy in static dictionaries with O (1) worst case lookup time. *Automata, Languages and Programming*. 1999:701-.
146. Fotakis D, Pagh R, Sanders P, Spirakis P. Space efficient hash tables with worst case constant access time. *Theory of Computing Systems*. 2005;38(2):229-48.

## References

---

147. Osuna E, Girosi F. Reducing the run-time complexity of support vector machines. *Advances in Kernel Methods: Support Vector Learning*, MIT press, Cambridge, MA. 1999:271-84.
148. Lawrence N, Seeger M, Herbrich R. Fast sparse Gaussian process methods: The informative vector machine. *Advances in Neural Information Processing Systems: Citeseer*; 2003. p. 625-8.
149. Eysenbach G, Powell J, Kuss O, Sa E. Empirical studies assessing the quality of health information for consumers on the world wide web: a systematic review. *Jama*. 2002;287(20):2691.
150. Chapman WW, Cohen KB. Current issues in biomedical text mining and natural language processing. *J Biomed Inform*. 2009 Oct;42(5):757-9.
151. Deshazo JP, Turner AM. An interactive and user-centered computer system to predict physician's disease judgments in discharge summaries. *J Biomed Inform*. 2009 Sep 3.
152. Zhang R, Shepherd M, Duffy J, Watters C, Automatic Web Page Categorization using Principal Component Analysis. 40th Hawaii International Conference on System Sciences; 2007: IEEE.
153. Ypma A, Heskes T. Automatic categorization of web pages and user clustering with mixtures of hidden Markov models. *Lecture Notes in Artificial Intelligence 2703*: Springer; 2003. p. 35-49.
154. Attardi G, Gulli A, Sebastiani F, Automatic Web page categorization by link and context analysis. *Proceedings of THAI'99*; 1999; Varese, Italy.
155. Roy S, Joshi S, Krishnapuram R, Automatic categorization of web sites based on source types. *HT '04*; 2004; Santa Cruz, CA: ACM.
156. Mase H. Experiments on Automatic Web Page Categorization for IR System: Stanford University, 1998.
157. Sahlgren M, Holst A, Kanerva P, Permutations as a means to encode order in word space. *Proc 30th Annual Meeting of the Cognitive Science Society (CogSci'08)*; July 23-26, 2008; Washington D.C.: Cognitive Science Society.
158. Sahlgren M, An introduction to random indexing 2005. *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*, August 16, Copenhagen, Denmark.

## Appendix A

---

### Appendix A: Stop Words list

<i>a</i>	<i>do</i>	<i>instead</i>	<i>particularly</i>	<i>thorough</i>
<i>a's</i>	<i>does</i>	<i>into</i>	<i>per</i>	<i>thoroughly</i>
<i>able</i>	<i>doesn't</i>	<i>inward</i>	<i>perhaps</i>	<i>those</i>
<i>about</i>	<i>doing</i>	<i>is</i>	<i>placed</i>	<i>though</i>
<i>above</i>	<i>don't</i>	<i>isn't</i>	<i>please</i>	<i>three</i>
<i>according</i>	<i>done</i>	<i>it</i>	<i>plus</i>	<i>through</i>
<i>accordingly</i>	<i>down</i>	<i>it'd</i>	<i>possible</i>	<i>throughout</i>
<i>across</i>	<i>downwards</i>	<i>it'll</i>	<i>presumably</i>	<i>thru</i>
<i>actually</i>	<i>during</i>	<i>it's</i>	<i>probably</i>	<i>thus</i>
<i>after</i>	<i>e</i>	<i>its</i>	<i>provides</i>	<i>to</i>
<i>afterwards</i>	<i>each</i>	<i>itself</i>	<i>q</i>	<i>together</i>
<i>again</i>	<i>edu</i>	<i>j</i>	<i>que</i>	<i>too</i>
<i>against</i>	<i>eg</i>	<i>just</i>	<i>quite</i>	<i>took</i>
<i>ain't</i>	<i>eight</i>	<i>k</i>	<i>qv</i>	<i>toward</i>
<i>all</i>	<i>either</i>	<i>keep</i>	<i>r</i>	<i>towards</i>
<i>allow</i>	<i>else</i>	<i>keeps</i>	<i>rather</i>	<i>tried</i>
<i>allows</i>	<i>elsewhere</i>	<i>kept</i>	<i>rd</i>	<i>tries</i>
<i>almost</i>	<i>enough</i>	<i>know</i>	<i>re</i>	<i>truly</i>
<i>alone</i>	<i>entirely</i>	<i>knows</i>	<i>really</i>	<i>try</i>
<i>along</i>	<i>especially</i>	<i>known</i>	<i>reasonably</i>	<i>trying</i>
<i>already</i>	<i>et</i>	<i>l</i>	<i>regarding</i>	<i>twice</i>
<i>also</i>	<i>etc</i>	<i>last</i>	<i>regardless</i>	<i>two</i>
<i>although</i>	<i>even</i>	<i>lately</i>	<i>regards</i>	<i>u</i>
<i>always</i>	<i>ever</i>	<i>later</i>	<i>relatively</i>	<i>un</i>
<i>am</i>	<i>every</i>	<i>latter</i>	<i>respectively</i>	<i>under</i>
<i>among</i>	<i>everybody</i>	<i>latterly</i>	<i>right</i>	<i>unfortunately</i>
<i>amongst</i>	<i>everyone</i>	<i>least</i>	<i>s</i>	<i>unless</i>
<i>an</i>	<i>everything</i>	<i>less</i>	<i>said</i>	<i>unlikely</i>
<i>and</i>	<i>everywhere</i>	<i>lest</i>	<i>same</i>	<i>until</i>
<i>another</i>	<i>ex</i>	<i>let</i>	<i>saw</i>	<i>unto</i>
<i>any</i>	<i>exactly</i>	<i>let's</i>	<i>say</i>	<i>up</i>



## Appendix A

---

<i>anybody</i>	<i>example</i>	<i>like</i>	<i>saying</i>	<i>upon</i>
<i>anyhow</i>	<i>except</i>	<i>liked</i>	<i>says</i>	<i>us</i>
<i>anyone</i>	<i>f</i>	<i>likely</i>	<i>second</i>	<i>use</i>
<i>anything</i>	<i>far</i>	<i>little</i>	<i>secondly</i>	<i>used</i>
<i>anyway</i>	<i>few</i>	<i>look</i>	<i>see</i>	<i>useful</i>
<i>anyways</i>	<i>fifth</i>	<i>looking</i>	<i>seeing</i>	<i>uses</i>
<i>anywhere</i>	<i>first</i>	<i>looks</i>	<i>seem</i>	<i>using</i>
<i>apart</i>	<i>five</i>	<i>ltd</i>	<i>seemed</i>	<i>usually</i>
<i>appear</i>	<i>followed</i>	<i>m</i>	<i>seeming</i>	<i>uucp</i>
<i>appreciate</i>	<i>following</i>	<i>mainly</i>	<i>seems</i>	<i>v</i>
<i>appropriate</i>	<i>follows</i>	<i>many</i>	<i>seen</i>	<i>value</i>
<i>are</i>	<i>for</i>	<i>may</i>	<i>self</i>	<i>various</i>
<i>aren't</i>	<i>former</i>	<i>maybe</i>	<i>selves</i>	<i>very</i>
<i>around</i>	<i>formerly</i>	<i>me</i>	<i>sensible</i>	<i>via</i>
<i>as</i>	<i>forth</i>	<i>mean</i>	<i>sent</i>	<i>viz</i>
<i>aside</i>	<i>four</i>	<i>meanwhile</i>	<i>serious</i>	<i>vs</i>
<i>ask</i>	<i>from</i>	<i>merely</i>	<i>seriously</i>	<i>w</i>
<i>asking</i>	<i>further</i>	<i>might</i>	<i>seven</i>	<i>want</i>
<i>associated</i>	<i>furthermore</i>	<i>more</i>	<i>several</i>	<i>wants</i>
<i>at</i>	<i>g</i>	<i>moreover</i>	<i>shall</i>	<i>was</i>
<i>available</i>	<i>get</i>	<i>most</i>	<i>she</i>	<i>wasn't</i>
<i>away</i>	<i>gets</i>	<i>mostly</i>	<i>should</i>	<i>way</i>
<i>awfully</i>	<i>getting</i>	<i>much</i>	<i>shouldn't</i>	<i>we</i>
<i>b</i>	<i>given</i>	<i>must</i>	<i>since</i>	<i>we'd</i>
<i>be</i>	<i>gives</i>	<i>my</i>	<i>six</i>	<i>we'll</i>
<i>became</i>	<i>go</i>	<i>myself</i>	<i>so</i>	<i>we're</i>
<i>because</i>	<i>goes</i>	<i>n</i>	<i>some</i>	<i>we've</i>
<i>become</i>	<i>going</i>	<i>name</i>	<i>somebody</i>	<i>welcome</i>
<i>becomes</i>	<i>gone</i>	<i>namely</i>	<i>somehow</i>	<i>well</i>
<i>becoming</i>	<i>got</i>	<i>nd</i>	<i>someone</i>	<i>went</i>
<i>been</i>	<i>gotten</i>	<i>near</i>	<i>something</i>	<i>were</i>
<i>before</i>	<i>greetings</i>	<i>nearly</i>	<i>sometime</i>	<i>weren't</i>
<i>beforehand</i>	<i>h</i>	<i>necessary</i>	<i>sometimes</i>	<i>what</i>
<i>behind</i>	<i>had</i>	<i>need</i>	<i>somewhat</i>	<i>what's</i>

## Appendix A

---

<i>being</i>	<i>hadn't</i>	<i>needs</i>	<i>somewhere</i>	<i>whatever</i>
<i>believe</i>	<i>happens</i>	<i>neither</i>	<i>soon</i>	<i>when</i>
<i>below</i>	<i>hardly</i>	<i>never</i>	<i>sorry</i>	<i>whence</i>
<i>beside</i>	<i>has</i>	<i>nevertheless</i>	<i>specified</i>	<i>whenever</i>
<i>besides</i>	<i>hasn't</i>	<i>new</i>	<i>specify</i>	<i>where</i>
<i>best</i>	<i>have</i>	<i>next</i>	<i>specifying</i>	<i>where's</i>
<i>better</i>	<i>haven't</i>	<i>nine</i>	<i>still</i>	<i>whereafter</i>
<i>between</i>	<i>having</i>	<i>no</i>	<i>sub</i>	<i>whereas</i>
<i>beyond</i>	<i>he</i>	<i>nobody</i>	<i>such</i>	<i>whereby</i>
<i>both</i>	<i>he's</i>	<i>non</i>	<i>sup</i>	<i>wherein</i>
<i>brief</i>	<i>hello</i>	<i>none</i>	<i>sure</i>	<i>whereupon</i>
<i>but</i>	<i>help</i>	<i>noone</i>	<i>t</i>	<i>wherever</i>
<i>by</i>	<i>hence</i>	<i>nor</i>	<i>t's</i>	<i>whether</i>
<i>c</i>	<i>her</i>	<i>normally</i>	<i>take</i>	<i>which</i>
<i>c'mon</i>	<i>here</i>	<i>not</i>	<i>taken</i>	<i>while</i>
<i>c's</i>	<i>here's</i>	<i>nothing</i>	<i>tell</i>	<i>whither</i>
<i>came</i>	<i>hereafter</i>	<i>novel</i>	<i>tends</i>	<i>who</i>
<i>can</i>	<i>hereby</i>	<i>now</i>	<i>th</i>	<i>who's</i>
<i>can't</i>	<i>herein</i>	<i>nowhere</i>	<i>than</i>	<i>whoever</i>
<i>cannot</i>	<i>hereupon</i>	<i>o</i>	<i>thank</i>	<i>whole</i>
<i>cant</i>	<i>hers</i>	<i>obviously</i>	<i>thanks</i>	<i>whom</i>
<i>cause</i>	<i>herself</i>	<i>of</i>	<i>thanx</i>	<i>whose</i>
<i>causes</i>	<i>hi</i>	<i>off</i>	<i>that</i>	<i>why</i>
<i>certain</i>	<i>him</i>	<i>often</i>	<i>that's</i>	<i>will</i>
<i>certainly</i>	<i>himself</i>	<i>oh</i>	<i>thats</i>	<i>willing</i>
<i>changes</i>	<i>his</i>	<i>ok</i>	<i>the</i>	<i>wish</i>
<i>clearly</i>	<i>hither</i>	<i>okay</i>	<i>their</i>	<i>with</i>
<i>co</i>	<i>hopefully</i>	<i>old</i>	<i>theirs</i>	<i>within</i>
<i>com</i>	<i>how</i>	<i>on</i>	<i>them</i>	<i>without</i>
<i>come</i>	<i>howbeit</i>	<i>once</i>	<i>themselves</i>	<i>won't</i>
<i>comes</i>	<i>however</i>	<i>one</i>	<i>then</i>	<i>wonder</i>
<i>concerning</i>	<i>i</i>	<i>ones</i>	<i>thence</i>	<i>would</i>
<i>consequently</i>	<i>i'd</i>	<i>only</i>	<i>there</i>	<i>would</i>
<i>consider</i>	<i>i'll</i>	<i>onto</i>	<i>there's</i>	<i>wouldn't</i>



## Appendix A

---

<i>considering</i>	<i>i'm</i>	<i>or</i>	<i>thereafter</i>	<i>x</i>
<i>contain</i>	<i>i've</i>	<i>other</i>	<i>thereby</i>	<i>y</i>
<i>containing</i>	<i>ie</i>	<i>others</i>	<i>therefore</i>	<i>yes</i>
<i>contains</i>	<i>if</i>	<i>otherwise</i>	<i>therein</i>	<i>yet</i>
<i>corresponding</i>	<i>ignored</i>	<i>ought</i>	<i>theres</i>	<i>you</i>
<i>could</i>	<i>immediate</i>	<i>our</i>	<i>thereupon</i>	<i>you'd</i>
<i>couldn't</i>	<i>in</i>	<i>ours</i>	<i>these</i>	<i>you'll</i>
<i>course</i>	<i>inasmuch</i>	<i>ourselves</i>	<i>they</i>	<i>you're</i>
<i>currently</i>	<i>inc</i>	<i>out</i>	<i>they'd</i>	<i>you've</i>
<i>d</i>	<i>indeed</i>	<i>outside</i>	<i>they'll</i>	<i>your</i>
<i>definitely</i>	<i>indicate</i>	<i>over</i>	<i>they're</i>	<i>yours</i>
<i>described</i>	<i>indicated</i>	<i>overall</i>	<i>they've</i>	<i>yourself</i>
<i>despite</i>	<i>indicates</i>	<i>own</i>	<i>think</i>	<i>yourselves</i>
<i>did</i>	<i>inner</i>	<i>p</i>	<i>third</i>	<i>z</i>
<i>didn't</i>	<i>insofar</i>	<i>particular</i>	<i>this</i>	<i>zero</i>
<i>different</i>				