Comparing Title Only and Full Text Indexing to Classify Web Pages into Bookmark Categories

Chris Staff and Charlie Abela

Department of Artificial Intelligence, University of Malta
{chris.staff|charlie.abela}@um.edu.mt

Abstract. Web browser bookmark files are used to retain and organise records of web sites that the user would like to revisit. However, bookmark files tend to be under-utilised, as time and effort is needed to keep them organised. We use two methods to index and automatically classify documents referred to in 80 bookmark files, based on document title-only and full-text indexing, respectively. We evaluate the indexing methods by selecting a bookmark entry to classify from a bookmark file, and re-creating the bookmark file so that it contains only entries created before the selected bookmark entry. Classification based on full-text indexing generally outperforms that based on document title only indexing. The ability to recommend the correct category at rank 1 using full-text indexing ranges from 20% to 41%, depending on the number of category members. However, combining the approaches results in a increase to 37% — 59%, but we would need to recommend up to two categories to users. By recommending up to 10 categories, this increases to 58% — 80%.

1 Introduction

Web browser bookmark files are intended to be a repository of web sites that the user would like to revisit. However, bookmark files tend to be passive. Typically, user effort is required to keep a bookmark file organised, and usually, bookmark files become disorganised over time [AB97,ABC98]. Recommendations for assisting with bookmark file organisation include "filing" new bookmark entries [AB97,ABC98]. Although many contemporary web browsers allow users to choose a destination bookmark category for the new bookmark entry, the first category presented to the user is either the top-level one (Mozilla, Firefox, Internet Explorer) or the last category in which a bookmark was created (Safari). In previous work [Bug06,SB07] we presented HyperBK, a bookmark management system that is able to recommend a destination bookmark category (folder). Only a small number of bookmark files were used in the evaluation of HyperBK. Here, we have modified our approach to indexing and classification (HyperBK2), and we have evaluated the new approach with 80 bookmark files. Whereas with HyperBK we selected up to 10 bookmark entries from each bookmark file and attempted to classify them into the original categories, this time we re-create a 'snapshot' of the bookmark file containing only those bookmark entries that

existed just prior to the selected web page being bookmarked, which is more realistic. In this way, we do not take future information into account, and we mimic the recommendations made to users at the time the bookmark entry would be created. HyperBK selected the highest weighted 5 terms from bookmarked web pages, combined them into a category description, and then compared them with a similarly selected set of terms from the web page to classify. If a classification was not possible, then other features, such as URLs, etc., were used to determine the recommended category. HyperBK2 uses Term Frequency and a Normalised Document Frequency (see section 3) to weight terms, creates a centroid category representation based on the term weights occurring in the category, and uses the Cosine Similarity Measure to recommend membership of a web page.
In section 2 we discuss similar systems. HyperBK2's indexing and classification approach is discussed in section 3. The approach to evaluation is described in section 4, and the results are presented and discussed in section 5. Finally, section 6 outlines our future work and conclusion.

2 Background and Similar Systems

Bookmarking is one of the most popular ways in which people store and organise information for later use [Bru04]. However, drawbacks exist, especially when the number of bookmarks increases over time [Bru01]. Numerous tools, called bookmark managers, exist to help support users in creating and maintaining, effectively-reusable bookmarks' lists. These can be categorised either as *"centrally store and browse"* [Ben06], such as HyperBK [SB07], Conceptual Navigator [Cna04] and Check&Get [Chg07] or as *collaborative*, such as Delicious [Del07] and BookmarkTracker [Bmt07]. Bookmark managers need to provide support when it comes to effectively classify newly created bookmarks.
Xia *et. al.*'s approach to bookmark classification classifies a web page by considering information from neighbouring pages in the link graph [Xia06]. Neighbouring pages are referred to as parent, child, sibling and spouse. The adopted method does not rely on the appearance of labelled pages (i.e. pages whose categories are already known) in the neighbourhood of a specific page and this leads to a wider applicability. Evaluations showed that classification increased from around 70% to more than 90%, using pages from the Open Directory Project [Odp06].
Delicious [Del07], which is an online service, allows users to share bookmarks. Categorisation is aided by the use of tags, which users associate with their bookmarks. However there are no explicit category recommendations when a new bookmark is being stored. InLinx [InL03] provides for both recommendations and classification of bookmarks into "globally predefined categories". Classification is based on the user's profile and the web-page content. Classification of a new bookmark, according to [Ben06], is a matter of first establishing a similarity in the interests of two users and then finding a mapping between the folder location of a bookmark in the collaborator's bookmarks' list and that of the target-user's bookmark hierarchy. The similarity between two user profiles is computed through the classical cosine vector-similarity. The generation of a

recommendation about the best folder in which to place a new bookmark is dependent on similarity between the collaborating partners, the folder similarity between collaborative partner and target user, and the number of times that a folder was recommended. Recommendations are however based on a computed combined user-similarity (i.e. all users who have recommended a specific folder) and a combined folder similarity (computed recommendation based on all folders the recommended folder was mapped from). A new folder is created when the recommended folder is the target user's root folder and when the total similarity of the recommended folder falls below a certain threshold.

Classification is usually based on a set of features. However the number of features considered varies between the different classification approaches taken. In the approach suggested by [Jen01] these features for classification are reduced through the use of rough sets. This approach considers the indiscernability between two objects and tries to reduce the number of objects by keeping only one tuple in a minimal set, which is called a reduct.

Robertson's approach to bookmark organisation and classification focuses on categorisation through visualisation [Rob98] using a technique called *Data Mountain*, which allows users to place documents in any position on an inclined plane in a 3D virtual environment while using 2D interaction. Data Mountain takes advantage of spatial memory (i.e. the ability to memorise spatial information, such as the geographical layout of things). This technique was compared with the favourites' mechanism in Internet Explorer. The results showed that it provided more personalisation since it presents a whole view of the bookmarks' space and the spatial relationships between the pages.

3 Indexing and Classification Approach

We create two indices for each document in a category and the web page to classify, using either the document title or the full-text of the document, and we compare the performance of each in the evaluation section (section 4).

A bookmark file contains references to a number of web pages that a user has bookmarked. During the indexing stage, we remove stop words, stem the remaining terms using the Porter Stemmer [Por97], and calculate the term frequency for each stem (here, the term frequency is a non-normalised term count).

Once the indexing of bookmark entries is complete, we identify the documents that are to be used to create the description of each category. We take the document index of each document d_1 to d_N in a category and we merge them, calculating a term weight by summing the term frequencies (TF) of each term j_1 to j_m in each document in the category, and multiplying it by the Normalised Document Frequency, $\sum_{d=1}^{N} TF_{j_i,d} \times NDF_{j_i}$, where N is the number of documents in the category. This has the effect of reducing the weight of terms that occur in few documents in the category.

We create a representation of the web page to classify in the same way, although, obviously, the NDF is 1, so the weights of terms are their TF. We then use the Cosine Similarity Measure [SB87] to measure the similarity between the

bookmark entry to place and each category in the bookmark file snapshot. The highest ranking category, if there is one, is recommended.

3.1 Processing Steps

HyperBK2 is a series of Python 2.3.5 programs that process bookmark files to access and download bookmark entries; create representations of and data files for categories and bookmark entries in each bookmark file; remove script tags from each downloaded HTML file; determine the order in which bookmark entries are created within the bookmark file and category; create a full-text index (of stemmed words without stopwords, using Gupta's Python implementation of the Porter Stemmer[1]) of each downloaded HTML file; create evaluation platforms according to given criteria 4; and run and analyse the evaluation platform. Whenever we download a bookmarked web page, we create a full-text index for it, comprising the unique stems of terms, and their frequency. We also keep track of bookmark entries in the same category that may have been created during the same session. For instance, entries created up to 30 minutes apart may be considered to have been created as part of the same session. Jansen and Spink report that web researchers use a session length of anywhere between 5 minutes and 3 hours [JS06]. We use a session duration of at least 30 minutes and at least 3 hours. A group of bookmark entries created in the same session is a 'set'.
Once we have determined the set and category members, we can create a term weight vector description of each category or set that exists in the bookmark file by merging the indices of bookmark entries of each entry in the set or category into a centroid representation, or average pseudo-document. We also create, on-the-fly, term weight vectors based on the titles of category or set members.
To make a category recommendation we take the full-text index of the bookmark entry to classify and an index derived from the title only of the entry to classify, and we compare these against each bookmark file snapshot category and set vector descriptions using the Cosine Similarity Measure [SB87]. A score of 0 means that no recommendation was made, otherwise the category with the highest score is the recommended category. Even when we use sets, we are interested only in assigning the selected bookmark entry to the correct category.

4 Evaluation Approach

We collected 80 real bookmark files. Each bookmark file is built according to the Netscape bookmark file format[2], and stores the date that each bookmark entry was created. We use the ADD-DATE field to re-create the bookmark file as a snapshot of its state just prior to the addition of the bookmark to be classified. The basic method of evaluation for HyperBK2 is to select bookmark entries from a number of bookmark files, according to some criteria, and to measure the

[1] `http://tartarus.org/martin/PorterStemmer/python.txt`
[2] `http://msdn2.microsoft.com/en-us/library/Aa753582.aspx`

ability of the indexing and classification methods to recommend their original category. We measure the presence of the target category from ranks 1 to 5.
The criteria we use to select bookmark entries for classification from a bookmark file, to determine the eligibility of the bookmark file snapshot to participate in the particular run, are ENTRY-TO-TAKE, SET, and NO-OF-CATEGORIES. SET is false or true, depending on whether a bookmark entry is to be taken from a category, or whether the entry should be taken from a set of entries created in the same session within a category, respectively. We measure a session over either 30 minutes or 3 hours to determine if there is a significant difference in the ability to classify a web page over a longer session time.
ENTRY-TO-TAKE is the nth entry in a category (or set) that is selected for classification. We expect a category (or set) to contain n-1 entries before we select a bookmark entry for classification. If there is a problem with the bookmark entry selected (i.e., it no longer exists, etc.), then we take the next entry in the set or category, if possible. We ran HyperBK2 with values for ENTRY-TO-TAKE of 2, 4, 6, 7, 8, 9, and 11. For example, in the simplest case (ENTRY-TO-TAKE = 2), the second entry created in a category/set would be selected for classification, and a snapshot of that category would contain only one entry.
Finally, NO-OF-CATEGORIES is the number of categories that must exist in a snapshot of a bookmark file for the bookmark file to participate in the evaluation. We imposed a minimum of 5 categories, which would give a random classifier a maximum 20% chance of correctly assigning a selected bookmark entry to its original category. For instance, if the snapshot of the bookmark file contains less than 5 categories, then we will not include that snapshot in the evaluation. We wanted to see if we would bias results in HyperBK2's favour if we did not impose this minimum, so we removed this constraint for the evaluation platform with the best performing criteria.
In all we ran eighteen evaluation platforms with the following characteristics. All but two of the platforms contained bookmark file snapshots with a minimum of 5 categories. Each run evaluated the algorithms on either categories or sets. Sets were composed mainly of bookmark entries that were created within 30 minutes of each other (two platforms used a session length of 3 hours). The bookmark entry to classify was the 2nd, 4th, 6th, 7th, 8th, 9th, or 11th entry created in the category (or set). If the web page of the bookmark entry no longer exists, or if it was in a format other than HTML or XML, then we selected the next bookmark entry that satisfied the criteria.
The evaluation platform that gave the best results (see section 5) classified the 8th bookmark entry in a category, with a NO-OF-CATEGORIES of 5. To see if the number of categories and the session length had a significant impact on results, we also ran the evaluation platform with a 3 hour session length (maximum of 3 hours between the creation time of bookmark entries in the same category considered to belong to the same set) and no minimum number of categories (so that even if a bookmark file snapshot contained only the category from which the bookmark entry for classification was selected, we still included it in the evaluation). The session length had no significant impact. Removing the mini-

mum number of categories enabled more bookmark entries to participate in the evaluation (table 2), although there was no significant impact on classification accuracy (tables 3 — 5).

5 Results

In this section, we describe the general properties of the bookmark files that we collected, in terms of the number of categories that they contain and we present and discuss the results of the runs (tables 3 to 4) in tabular format, highlighting the best performances. On average, bookmark files used in the evaluation have

Table 1: Submitted bookmark files and their numbers of categories

No. of categories	No. of bookmark files
1	8
2–5	28
6–10	13
11–20	10
21–50	9
51–100	8
101+	4

23 categories, with a minimum of 1 and a maximum of 229. Table 1 gives the approximate number of categories in each bookmark file used in the evaluation. 8 files (10%) contain only one category. 51 files (63.75%) contain between 2 and 20 categories, and 21 (26.25%) contain more than 20 categories. We conducted

Table 2: No. of bookmark entries classified, by Category and Set (Note: each pair of runs shares the same characteristics)

Run	1/2	3/4	5/6	7/8	9/10	11/12	13/14	15/16	17/18
ENTRY-TO-TAKE	2	4	6	7	8	8	8	9	11
Session length (mins)	30	30	30	30	30	180	30	30	30
NO-OF-CATEGORIES	5	5	5	5	5	5	1	5	5
totalEligibleEntriesSet	1567	626	383	318	259	261	304	207	147
totalEligibleEntriesCat	1373	813	563	470	395	395	470	332	248
inBothTotal	1064	567	372	310	253	255	281	204	144
inCatOnlyTotal	309	246	191	160	142	140	189	128	104
inSetOnlyTotal	503	59	11	8	6	6	23	3	3
Percentage inBoth	57	65	65	65	63	64	57	61	57

the evaluation as follows. From each bookmark file, all the bookmark entries that satisfied the criteria (section 4) were extracted, and a snapshot of the bookmark

file was created per selected bookmark entry. The category and set evaluations may have selected different, but possibly overlapping, bookmark entries from the same bookmark file. On average, more bookmark entries were selected by category than by set. 60%+-5% were selected by both, and the focus of the evaluation in this paper is on the results of these (table 2). There is no significant increase in the number of bookmarks created within the same category in the same session when the session length is 3 hours (compare Runs 9/10 and 11/12 in table 2), suggesting that bookmarks are created in bursts.
Table 3 shows the percentage of correct category recommendations at rank 1 and the cumulative percentage of correct categories appearing in the recommendations from ranks 2 to 5 for the approach to classification using full-text indexing for both sets and categories. The ranks of correct results returned by

Table 3: Comparison of results of full-text indexing for sets (FTS) and categories (FTC) (percent) [E=ENTRY-TO-TAKE/ S=SESSION/ N=NO-OF-CATEGORIES]

E/S/N	2/30/5	4/30/5	6/30/5	7/30/5	8/30/5	8/180/5	8/30/1	9/30/5	11/30/5
FTS Rank 1	24	27	38	38	43	43	44	44	35
FTC Rank 1	**24**	**27**	**38**	**38**	**43**	**43**	**44**	**44**	**36**
FTS Rank 2	30	35	47	50	49	50	53	53	48
FTC Rank 2	**31**	**36**	**48**	**50**	**50**	**50**	**53**	**54**	**48**
FTS Rank 3	34	42	55	57	57	58	60	58	54
FTC Rank 3	**36**	**44**	**56**	**56**	**57**	**57**	**60**	**60**	**55**
FTS Rank 4	38	46	60	62	60	61	63	63	58
FTC Rank 4	**40**	**48**	**61**	**62**	**59**	**59**	**62**	**64**	**59**
FTS Rank 5	41	50	64	63	64	65	66	66	62
FTC Rank 5	**43**	**52**	**65**	**63**	**64**	**64**	**66**	**70**	**63**

each approach are shown in alternating lines to demonstrate that there is a similarity in performance throughout. Basically, it does not seem to matter whether the indexing is based on a set or on a category, because the recommendation made is very nearly the same. Similar behaviour is displayed when a title-only approach to indexing is taken, although the percentages are generally slightly worse (table not shown due to space restrictions). As the performance of the algorithm appears to be independent of sets and categories, we will concentrate on only the full-text and title-only indexing and classification approach based on categories (table 4). We see that from rank 2 onwards, there is an advantage of the full-text indexing approach over the title-only approach (table 4). It also turns out that the title-only approach and the full-text indexing approach are frequently making different recommendations (table 8), even though either approach based on categories or sets make similar recommendations[3]. Ideally,

[3] Full-text indexing on a set and full-text indexing on a category will make the same recommendations R_1. Title-only indexing on a set will make the same recommendations R_2 as title-only indexing on a category, but R_1 and R_2 will not be identical.

the top ranking recommendation is the target category. However, the best performance was 44% and worst was 24% (FTC rank 1, both table 3) and 44% and 24% respectively for classification based on sets (FTS rank 1, also table 3). When we merge the results (table 5) we see a significant increase in accuracy — although users would need to be shown a larger number of recommended categories. In future work, we intend to analyse the documents to discover if we are able to predict which approach to use so that we may reduce the number of recommended categories. In tables 6 and 7 we show, for the best performing

Table 4: Comparison of results of full-text indexing (FTC) and title-only indexing (TC) for categories (percent) [E=ENTRY-TO-TAKE/ S=SESSION/ N=NO-OF-CATEGORIES]

E/S/N	2/30/5	4/30/5	6/30/5	7/30/5	8/30/5	8/180/5	8/30/1	9/30/5	11/30/5
TC rank 1	**26**	**32**	33	**41**	39	38	38	29	**40**
FTC rank 1	24	27	**38**	38	**43**	**43**	**44**	**44**	36
TC rank 2	31	**39**	39	50	46	45	45	41	48
FTC rank 2	31	36	**48**	50	**50**	**50**	**53**	**54**	48
TC rank 3	33	42	44	53	49	48	48	44	51
FTC rank 3	**36**	**44**	**56**	**56**	**57**	**57**	**60**	**60**	**55**
TC rank 4	33	44	46	54	52	51	51	45	53
FTC rank 4	**40**	**48**	**61**	**62**	**59**	**59**	**62**	**64**	**59**
TC rank 5	34	44	47	55	52	51	51	45	55
FTC rank 5	**43**	**52**	**65**	**63**	**64**	**64**	**66**	**70**	**63**

evaluation platform configuration (taking the 8th entry in a category to classify), the relationship between accuracy from rank 1 to 5 and the number/percentage of categories in the corresponding bookmark file snapshot respectively.

Table 5: Merging recommendations from different approaches gives higher precision/recall (percent) [E=ENTRY-TO-TAKE/ S=SESSION/ N=NO-OF-CATEGORIES]

E/S/N	2/30/5	4/30/5	6/30/5	7/30/5	8/30/5	8/180/5	8/30/1	9/30/5	11/30/5
Rank 1	37	43	52	56	59	59	59	53	53
Rank 2	46	52	61	67	67	67	69	63	64
Rank 3	51	59	68	74	74	74	76	69	68
Rank 4	55	64	73	78	77	77	78	73	72
Rank 5	58	66	75	79	80	80	80	76	76

Discounting bookmark files with only one category, the overwhelming majority of bookmark files (71%) have between 2 and 20 categories (table 1). Accuracy is highest at rank 1, with an accuracy of 65.6% and 53% for bookmark files containing 6–10 and 11–20 categories respectively. With bookmark files that

Table 6: Relationship between highest rank of correct recommendation and no. of categories in bookmark file snapshot

Categories	Not Found	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Totals
2–5	2	1	0	3	1	1	8
6–10	2	21	3	2	2	2	32
11–20	2	27	7	10	2	3	51
21–50	14	29	9	4	2	6	64
51–100	12	14	3	3	3	3	38
101+	13	29	9	6	2	1	60
Totals	46	121	31	29	12	16	253
Percentage	18	48	12	11	5	6	100

contain 6–10 and 11–20 categories (32% of submitted bookmark files with more than one category and 33% of bookmark file snapshots), accuracy at rank 5 is 93.7% and 96.5% respectively. This means that for bookmark files with these

Table 7: Relationship between highest rank of correct recommendation and no. of categories in bookmark file snapshot (percent)

Categories	Not Found	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Totals	Top 5
2–5	25	12.5	0	37.5	1	12.5	100	75
6–10	6.25	65.6	9.4	6.25	6.25	6.25	100	93.73
11–20	3.5	53	14	20	3.5	6	100	96.5
21–50	23	45	14	6	3	9	100	77
51–100	32	37	7.5	7.5	7.5	7.5	100	68
101+	22	48	15	10	3	2	100	78

numbers of categories, the ability to place the target category into the top 5 recommendations is almost complete. Of course, showing the user a maximum of 10 recommended categories when the bookmark file contains only 10 categories appears to be self-defeating, but at least the recommendations would be ordered, with the user's preferred category ranked 1 about 66% of the time. The worst performing class of bookmark file contains between 51 and 100 categories (68% at rank 5). As part of future work, we intend to measure the consistency (or cohesiveness) of bookmark file categories. This will allow us to measure the average similarity of entries in a bookmark file, to see if there is a relationship between classification accuracy and cohesiveness.

Table 8 gives the performance of title-cat and full-text-cat indexing and recommendation approaches, again broken down by numbers of categories in the bookmark file snapshots. Full-Text indexing recommends the user's preferred category in a higher rank more frequently than title-only indexing, except, surprisingly, when a bookmark file contains more than 50 categories.

Table 8: Which of Title-Cat (TC) and Full-Text-Cat (FTC) ranks higher?

Categories	Equal	TC Higher Rank	FTC Higher Rank	No Recommendation
2–5	1	1	4	2
6–10	7	4	19	2
11–20	17	9	23	2
21–50	11	9	30	14
51–100	7	11	8	12
101+	18	18	11	13

6 Future Work and Conclusions

We have extended work previously conducted on HyperBK in the area of automatic bookmark classification by comparing indexing and classification methods based on vector-based full-text and title-only representations of documents in a bookmark category. Additionally, we investigate whether there is any significant advantage to grouping bookmark entries within a category by time of creation (to form a set of bookmark entries created within the same session). We conducted several runs in which the bookmark entry to be selected for classification was the 2nd, 4th, 6th, 7th, 8th, 9th, or 11th entry created in a category (or set). We found that there appears to be no advantage to grouping bookmarks into sets of entries created during the same session. Although there was a notable difference in the numbers of participating bookmark entries (generally, less sets exist than other eligible bookmark entries), there was no noticeable improvement in recommendation accuracy. However, there is a significant difference when title-only or full-text indexing is used. Not only do the different approaches make different recommendations, one of which is likelier to be the correct category, but there appears to be a correlation between the method to use and number of categories that exist in a bookmark file.

Other future work includes determining if the approach to use (title-only or full-text indexing and classification) can be predicted from the type of document to be indexed, and whether it is worth defining a cohesiveness function to measure the relative similarity of documents in a category, to determine the likelihood of the category performing well in classification tasks.

We currently need to recommend a maximum of 10 different categories to the user to obtain the highest accuracy at rank 5, assuming that each approach recommended 5 completely different categories. However, on average, there is an overlap of 65% in recommended categories, so probably only around 7 different categories would be displayed. Ideally, the number of recommended categories could be reduced, using a combination of cohesiveness, prediction based on document features, and the correlation between indexing and classification approach and the number of categories in a bookmark file.

References

[AB97] David Abrams and Ron Baecker. How people use WWW bookmarks. In *CHI '97: CHI '97 extended abstracts on Human factors in computing systems*, pages 341–342, New York, NY, USA, 1997. ACM Press.

[ABC98] David Abrams, Ron Baecker, and Mark Chignell. Information archiving with bookmarks: personal Web space construction and organization. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 41–48, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[Ben06] Benz, D., Tso, K., Schmidt–Thieme, L., *Automatic Bookmark Classification: A Collaborative Approach*, 2nd Workshop on Innovations in Web Infrastructure, IWI2006, May 2006, Edinburgh, UK.

[Bmt07] *http://www.bookmarktracker.com/*

[Bru01] Bruce, H., Jones, W., Dumais, S. *Keeping Found Things Found on The Web*, in Proceedings of the 10th International Conference on Information and Knowledge Management, ACM CIKM 2001, pages 119–126.

[Bru04] Bruce, H., Jones, W., Dumais, S., *Keeping and Re-Finding Information on the Web: What do people do and what do they need?* in ASIST 2004: Proceedings of the 67th ASIST annual meeting. Chicago, IL: Information Today, Inc.

[Bug06] Ian Bugeja. Managing WWW broswer's bookmarks and history. FYP report, Department of Computer Science & AI, University of Malta, 2006.

[Chg07] Check&Get, *http://activeurls.com/*

[Cna04] Concept Navigator, *http://activeurls.com*

[Del07] Delicious, *http://del.icio.us./*

[InL03] Bighini, C., Carbonaro, A., Casadei, G., *InLinx for Document Classification, Sharing and Recommendation*, in the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03), p. 91, 2003

[Jen01] Jensen, R., Shen, Q,*A Rough Set-Aided System for Sorting WWW Bookmarks*, in Proceedings of the First Asia-Pacific Conference on Web Intelligence: Research and Development, pages 96–105, London, UK, 2001.

[JS06] Bernard J. Jansen and Amanda Spink. How are we searching the world wide web?: a comparison of nine search engine transaction logs. *Inf. Process. Manage.*, 42(1):248–263, 2006.

[Odp06] Open Directory Project, *http://www/dmoz.com*

[Por97] M. F. Porter. An algorithm for suffix stripping. *Readings in information retrieval*, pages 313–316, 1997.

[Rob98] Robertson, G., Czerwinski, M., Larson, K., Robbins, D.C., Thiel, D., and van Dantzich, M., *Data Mountain: Using Spatial Memory for Document Management*, in Symposium on User Interface Software and Technology, pg 153–162, 1998.

[SB87] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA, 1987.

[SB07] Chris Staff and Ian Bugeja. Automatic classification of web pages into bookmark categories. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 731–732, New York, NY, USA, 2007. ACM Press.

[Xia06] Xiaoguang, Q., Davison, B.D., *Knowing a web page by the company it keeps*, in Proceedings of the 15th ACM international conference on Information and knowledge management, CIKM '06, Arlington, Virginia, USA, (2006), pg 228–237