Title	Recommending topics for web curation
Authors(s)	Saaya, Zurina, Schaal, Markus, Rafter, Rachael, Smyth, Barry
Publication date	2013-06-14
Publication information	Saaya, Zurina, Markus Schaal, Rachael Rafter, and Barry Smyth. “Recommending Topics for Web Curation.” Springer, June 14, 2013. https://doi.org/10.1007/978-3-642-38844-6_20.
Conference details	UMAP 2013: The 21st Conference on User Modeling, Adaptation and Personalization, Rome, Italy, 10-14 June 2013
Series	Lecture Notes in Computer Science, Volume 7899
Publisher	Springer
Item record/more information	http://hdl.handle.net/10197/10881
Publisher's statement	The final publication is available at www.springerlink.com.
Publisher's version (DOI)	10.1007/978-3-642-38844-6_20

Recommending Topics for Web Curation

Zurina Saaya, Markus Schaal, Rachael Rafter and Barry Smyth

CLARITY: Centre for Sensor Web Technologies
School of Computer Science and Informatics
University College Dublin, Ireland
{zurina.saaya,markus.schaal,rachael.rafter,barry.smyth}@ucd.ie
http://www.clarity-centre.org

Abstract. A new generation of curation services provides users with a set of tools to manually curate and manage topical collections of content. However, given curation is ultimately a manual effort, it still requires significant effort on the part of the curator both in terms of collecting and managing content. We are interested in providing additional assistance to users in their curation tasks, in particular when it comes to efficiently adding content to their collection, and examine recommender systems in an effort to automate this task. We examine a number of recommendation strategies using live-user data from the popular Scoop.it curation service.

Keywords: curation, recommendation system, machine learning

1 Introduction

The recent information sharing and discovery trend is exemplified by the new generation of curation services (e.g. delicious.com, storify.com, clipboard.com, pinterest.com, scoop.it etc.) which provide users with a set of tools to manually curate and manage topical collections of content. In one sense these services represent a new generation of online bookmarking sites but with an emphasis on providing a much richer curation context supporting a variety of content types and leveraging the power of the social web as a key distribution channel [10]. Interested parties can create content collections of topics of interest, share these collections with friends and followers, and support an active and collaborative model of curation to develop unique collections of topical content; for example, Pinterest and Clipboard allow users to create topical collections of web elements (text, images, etc.). Collections themselves become new types of media; for example Storify allows users to narratively knit together collections of links and content to form unique stories. The work by Duh et al. [4] suggests that there are various reasons for why curation services have become popular such as the ability of curated collections to capture conversational and collaborative endeavours.

Ultimately curation is a manual effort, distinguishing it from more automated content discovery approaches such as search and recommendation [17]. The job of the curator is to identify, organise and frame content that is relevant, interesting and useful. To assist in this effort many curation services provide a set of tools to simplify the curation work-flow and to integrate curation features into a user's normal online activities; for example bookmarklets and browser plugins provide one-click curation by allowing a user to add a web page, image, or article directly to their collection as they browse. Nevertheless curation still requires significant effort by the curator and given that many curators will actively curate multiple collections deciding which collection to *assign* a new article to can introduce further friction to an otherwise streamlined workflow.

In this paper we are interested in providing additional support to curators and helping them to efficiently add content to their collections. To this end we propose to evaluate the use of recommender systems techniques in an effort to automate this assignment task. In short, our aim is to profile the evolving collections managed by a curator and to use these profiles as the basis for matching the target content at assignment time. In this way a curator can assign a new piece of content with a single-click, with the recommender system reliably choosing an appropriate collection without further intervention, or at the very least suggest a short-list of likely collections that the user can quickly chose from.

In what follows we will describe and evaluate a number of recommendation strategies for dealing with this assignment task. For the purpose of this study we will use live-user data collected from the popular *Scoop.it* curation service.

2 Web Curation using Scoop.it

Scoop.it is a web-based curation tool that provides a platform for users to curate all of their favorite resources on a given topic for sharing with interested parties. Users can create their own collections (or *topics* in the Scoop.it parlance) and the service provides a range of tools to help users to identify and filter content for their collections. Creating a collection or topic is a simple matter of providing a name, a short description, and a set of suitable keywords.

A unique feature of Scoop.it is that it allows users to create a set of *trigger queries* and assign them to certain *sources* of content online with a view to periodically collecting results from these sources as candidate content for the curator to review. For instance, a curator may create a new *Web Dev* collection as a place to curate all things related to web development. They might set up trigger queries such as *'javascript news'* for Google and *'nodejs tutorials'* for SlideShare to access a daily stream of potential content that they can then filter and add to their collection. Creating a range of these trigger queries and using diverse sources such as Google, SlideShare, Twitter, YouTube at least simplifies the task of locating candidate content, although it does place a significant filtering burden on the curator. Scoop.it also provides more conventional curation tools in the form of a bookmarklet that curators can add to their browser. When browsing a page of interest the curator can select the bookmarklet to *rescoop* the page, and after assigning an appropriate collection the page is added.

For our research Scoop.it is a excellent opportunity to explore different ways to personalize, automate and otherwise support the curation process. It is a popular service attracting more than 1500 new curators every day and approximately

5 million visitors per month resulting in a growing collection of collections and content. Scoop.it has also released a comprehensive API making it possible to work with live data. In this work then we focus on the task of assigning a newly discovered page to a specific collection. For the curator, as mentioned above, this is an entirely manual task, and for curators with many collections it can be time consuming and error prone. Our approach is to treat this as a type of *page classification* task in which the job is to classify a given page (URL) as belonging to one of the curator's collections. And we will do this by capturing the essence of a collection (and a page) using a combination of content features extracted from pages (their URLs, titles, descriptions etc). Ideally, we would like to be able to classify a given page as belonging to a single collection and thus automatically assign the page to this collection. However, this is unlikely to be practical in the face of inevitable imperfections in the classifier models that we learn and so instead we propose to identify a short-list of k suitable collections, ranked by their relevance to the target page, and presented to the user for assignment. So, instead of an active curator having to chose from perhaps 10 or more possible collections she will instead be faced with a choice of only 2 or 3 collections.

3 Related Work

Novel technologies, algorithms, and functionalities for web curation services are demonstrated and explored by upcoming commercial applications such as Storify, Pinterest, or Scoop.it, to name a few. But up to now little has been published which is targeted specifically at the assistance of web curation.

Bookmarking can be seen as an early form of private content curation. Several researchers have examined ways to enhance the organization of one's bookmarks. For example Maarek and Israel [12] describe an automatic URL classification technique to organize documents based on their conceptual similarity. Staff and Bugeja [20] develop a browser extension to classify bookmarks based on the term overlap between the keywords of categories and incoming pages. Private bookmarking later gave way to social bookmarking, typified by sites like Delicious where users collect and share their bookmarks with others. Finally, HeyStaks [19] took the idea of sharing useful web resources even further and introduced the concept of collaborative search. Saaya et al. [16] have explored a solution similar to [20] for identifying the correct content collection for HeyStaks. This was later improved using a machine learning approach [15]. We extend this work in the domain of web curation, specifically for Scoop.it, where the personal topics of the users are presented it in a meaningful and organized way around a specific theme rather than general-purpose content collections.

Our work is also related to tag recommendation, because tags may relate to web pages in a similar way as topics. Various approaches have been proposed for tag recommendation, using methods such as content- and graph-based collaborative filtering and machine learning [11, 8, 14]. However, there are some differences between the tag assignment task and ours. For example, tags usually consist of single keywords, which are redundant and ambiguous and may make it diffi-

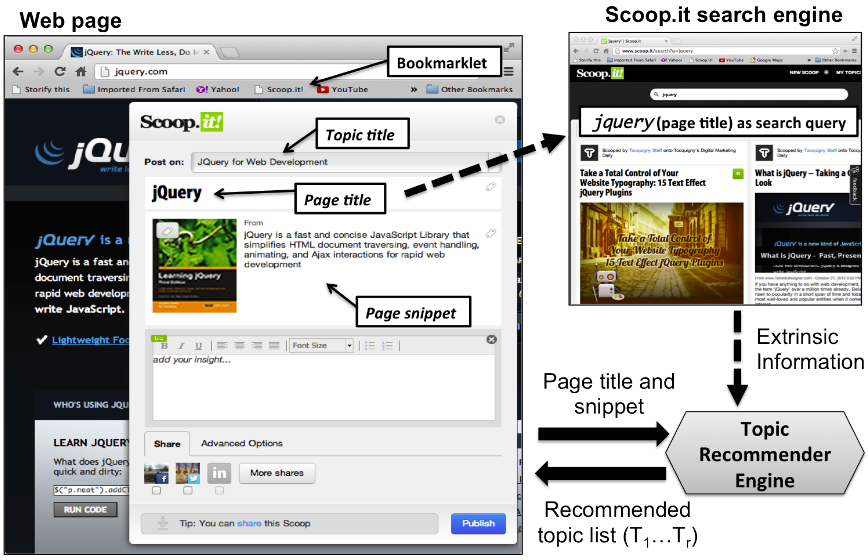


Fig. 1. Recommendation system overview

cult to describe the resources [13]. Opposite to this, topic titles and descriptions in Scoop.it, for example, provide more information, and therefore we consider the recommendation of the right topic for a web page as a crisp classification problem, while tagging is usually seen as a categorization process, see e.g. [6].

There are many web page features that can be exploited for classification. For example, Attardi et al. [1] use the page title and some content to suggest a target page's class. Alternatively Chakrabarti et al. [3] use the content of explicit links within the web page and Shen et al. [18] enhance this approach by combining explicit links found in the web page with implicit links gathered from the results of performing similar search queries. Finally, Baykan et al. [2] propose the URL as a main feature for topic classification. In our work, we use the title and snippet text as features of a page, and we experiment with combinations of intrinsic (found on page) and extrinsic (found on related pages) information.

4 Recommending Curated Collections

To begin we can consider a typical user U as being associated with a set of n collections or *topics* $T_U = T_1 ... T_n$; Scoop.it refers to its curated collections as *topics*, which we shall now adopt for clarity. The assignment task then is to suggest one or more of these topics as a suitable home for a given page or URL, by matching the page with a suitable topic.

This is reflected in the system overview presented in Fig. 1. In this case the curator is looking to add a page about *jQuery* to one of their collections. Two types of features are considered as the basis for recommending a topic. First, there are the *intrinsic* features extracted from the page title and snippet text.

Second, there are *extrinsic* features, which are not derived from the page's own content but rather based on how other Scoop.it users have curated pages like this one in the past. Thus the extrinsic features are determined by using the current page title as a query into Scoop.it. This leads to a collection of Scoop.it results and we use their page titles and snippets as the basis of our extrinsic features. The intuition here is that the combination of intrinsic and extrinsic features will provide a useful way to capture the essence of a new target page for use during topic recommendation. In a sense, the intrinsic features form a basic query for the page in question and the addition of extrinsic feature corresponds to a type of *query expansion* [21], augmenting the limited intrinsic features with additional relevant information. Of course, in a similar way, when it comes to profiling a collection of pages, we can also represent the pages in a collection in terms of their intrinsic and extrinsic features.

Returning to Fig. 1, using the combination of intrinsic and extrinsic features of the page to be curated, the recommendation engine compares these to the topic/collection profiles to produce a ranked list of r topics $T_1...T_r$ for return to the curator, the top one of which can be chosen as the most likely topic; in this case we see that the *JQuery for Web Development* topic has been chosen as the default. Ultimately the question we wish to answer is to what extent these recommended topics might be considered useful recommendations. How often do they contain the correct topic, for example? In addition we will also explore the relative value of intrinsic and extrinsic features (both for the query and collection) in this recommendation task.

In what follows we will describe two complementary approaches to topic recommendation: (1) an information retrieval type approach based on the work of [16] such that each topic/collection is indexed by the pages it contains and at recommendation time the target page is used to retrieve a ranked list of relevant topics; and (2) a machine learning approach in which the target page is *classified* as belonging to one of the curator's collection based on a classification model of these collections.

4.1 An Information Retrieval Approach to Recommendation

Each collection or topic $T \in T_U$ (for User U) corresponds to a set of curated pages P_T and each page $p \in P_T$ can be associated with a set of intrinsic and extrinsic features, $p = \{t_1, ..t_x\}$. In this case, the features are simply terms extracted from the page titles and snippets after stop-word removal. So the intrinsic features of p_i are the words from its title and in its snippet text. And the extrinsic features of p_i are the words from the titles and snippets of the pages returned by Scoop.it's own search function for a query that corresponds to the target page's title. In this way we can produce a *topic summary index* (TSI) from the user's topics by indexing each topic by the combination of terms of its constituent pages. And to retrieve a set of relevant topics for a new page we use p_Q as a retrieval probe against the user's TSI, where p_Q is build from the intrinsic and/or extrinsic features of the new page. Specifically, we use Lucene [7] to do this probe which calculates a TF-IDF relevance score for each of the user's topics ($T \in T_U$) according to Equation 2, in order to produce a ranked list of topics for recommendation to user U according to Equation 1.

$$TopicList(p_Q, T_U, TSI) = \begin{array}{l} SortDesc(Score(p_Q, T, TSI)) \\ \forall T \in T_U \end{array} \quad (1)$$

$$Score(p_Q, T, TSI) = \sum_{t \in p_Q} tf(t, T) \times idf(t, TSI) \quad (2)$$

4.2 A Classification Based Approach to Recommendation

As an alternative to the term-based information retrieval approach to topic recommendation described above, we also consider the use of text classification techniques. In this case we use the terms of those URL's page titles and snippets that are already assigned to the topic to train the classifier that is capable of classifying future URLs. For a given user U who has a list of topics $T_U = \{T_1...T_n\}$ we train the n-class classifier based on the content of these topics. Each topic T_i contributes a set of training data in the form of $\langle p, T_i \rangle$ where $p = \{t_1, ..t_x\}$ is the set of intrinsic and/or extrinsic features for a single page p and T_i is the topic id (class). Note, whereas topic information is effectively compiled into a single document for the information retrieval approach described above, in this classification approach, individual pages are used as separate training instances.

To build our classifiers we consider the following two distinct machine learning techniques for the purpose of comparison; Naive Bayes Multinomial(NBM) [9] and Support Vector Machines (SVM). We use standard reference implementations for these classifiers from the Weka-library [5]. Once a topic classifier has been learned for a given user, it can be used as the basis for topic recommendation. A new page is transformed into a similar feature representation and classified by the user's topic classifier and a ranked list of topics is produced based on the classifier's class predictions and confidence scores.

Note that in the above each user is associated with their own individual topic indexes or classifiers, thus personalising the topic recommendation task with respect to an individual user. It is a matter for future work as to whether there are any meaningful benefits to be gained from a more centralised cross-user approach to recommendation. We have also assumed the availability and use of both intrinsic and extrinsic features when it comes to representing the page query and indexing or learning the topics. As mentioned previously we will go on to explore the relative benefit of using intrinsic and extrinsic information in this way and so it is equally possible to use different combinations of intrinsic and extrinsic features without loss of generality as we shall see.

5 Experiment

Ultimately the success of this work will depend critically on the ability to make accurate topic recommendations at curation time. If the correct topic is likely to be contained in the topic recommendations, then we can reduce the curation friction significantly by limiting the user's topic choice. To test this, and to also evaluate the relative benefits of using intrinsic and extrinsic data, we use a dataset collected from the Scoop.it API during October/November 2012.

5.1 Summary Dataset

In summary this Scoop.it dataset contains 3620 topics (collections) from 560 users. On average these users have 6.5 topics; 92% of users have between 5 and 10 topics, with the remaining users having between 11 and 29 topics. Each topic contains at least 5 pages and in total the 3620 topics contain a total of more than 550,000 pages. For the purpose of this experiment we split the data into training and test data for each topic. In other words, for each topic we identify a set of training URLs and a set of test URLs. In total there are 404,507 training URLs and 155,626 test URLs. The training URLs are used to build the topic indexes and classifier models for each of the users and the test URLs are used to evaluate the generated recommendations; for the test URLs we use the known topic assignment as the ground-truth. We do this by calculating a *accuracy* score as the percentage of times that the correct topic was included in recommendation lists ranging in size from 1 to 5 topics.

5.2 Test Conditions

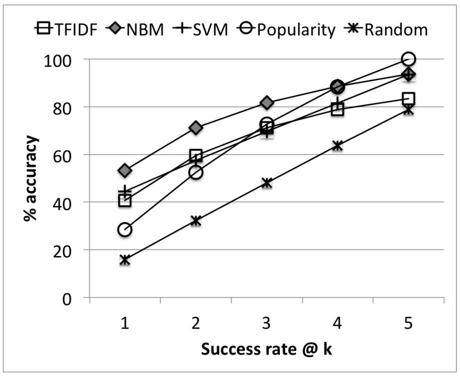
As described previously we use 3 different recommendation techniques, information retrieval approach $TFIDF$, a multinomial naive bayes approach (NBM), and a support vector machine approach (SVM). In addition we also include two baseline recommendation strategies: (1) *Random* simply selects a topic at random from the user's topic-list; and (2) *Popularity* ranks the most popular topics (that is those topics with the most pages) for recommendation.

We also consider the performance of these 5 techniques under 4 different conditions based on the use of combinations of intrinsic and extrinsic features during training and testing.

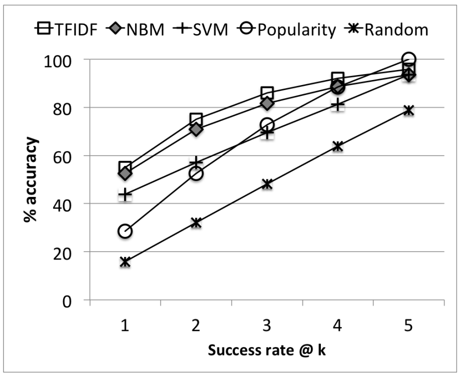
- **I-I**: only intrinsic features are used during training and testing.
- **I-X**: intrinsic features are used during training while both intrinsic and extrinsic features are used for the test data.
- **X-I**: intrinsic and extrinsic features are used during training but only intrinsic features are used for the test data.
- **X-X**: intrinsic and extrinsic features are used during training and testing.

5.3 Results

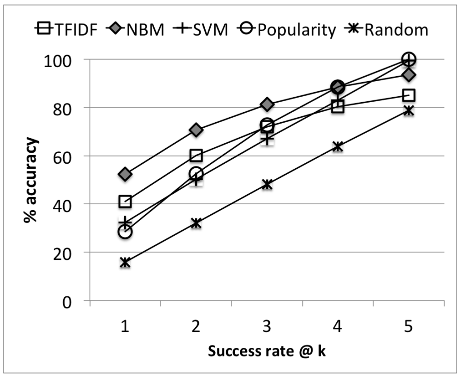
The accuracy results are presented in Fig. 2(a-d) for the different combinations of intrinsic/extrinsic data during training and testing. In each graph we plot the average accuracy against the recommendation list size for each of the 5 recommendation techniques. Overall we can see that the baseline techniques of



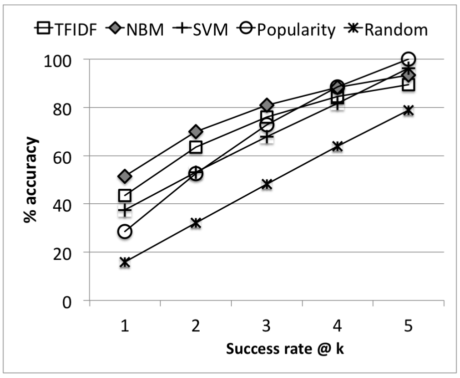
(a) **I-I**



(b) **I-X**



(c) **X-I**



(d) **X-X**

Fig. 2. Accuracy of topic recommendations for different feature conditions

Random and *Popularity* perform poorly, especially for short recommendation lists. For example, they deliver sub-30% accuracy for Top-1 recommendation lists, although it is worth noting that *Popularity* tends to become competitive for recommendation lists of size 3 or more. NBM tends to win out overall, with accuracy rates ranging from 50-85% across all conditions.

To summarize these accuracy results and in an effort to better clarify the differences between the use of intrinsic and extrinsic features, Fig. 3 shows the average accuracy for recommendation lists of size 3 for each of the algorithms and test conditions. Overall we can see clearly that NBM performs consistently well across all feature conditions (with accuracy rates of circa 80%), and in fact there is little or no benefit to including extrinsic features when using this technique. $TFIDF$ is less successful but does enjoy a significant benefit when extrinsic features are used for the page query. In this case it delivers an average accuracy of 86%; this is analogous to using a form of query elaboration at retrieval time. From a utility perspective the above results suggest that there is likely to be a curation benefit to using a recommendation approach for topic suggestion. The results suggest that we can present the user with a short-list of 3 topics at

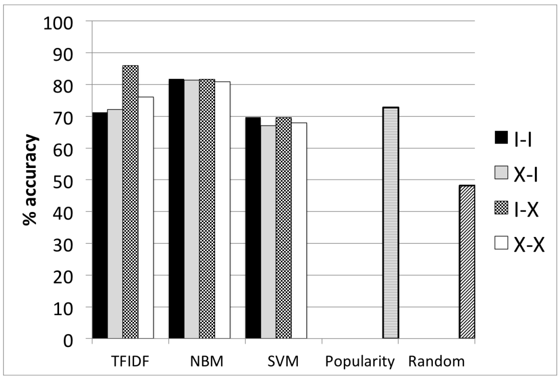


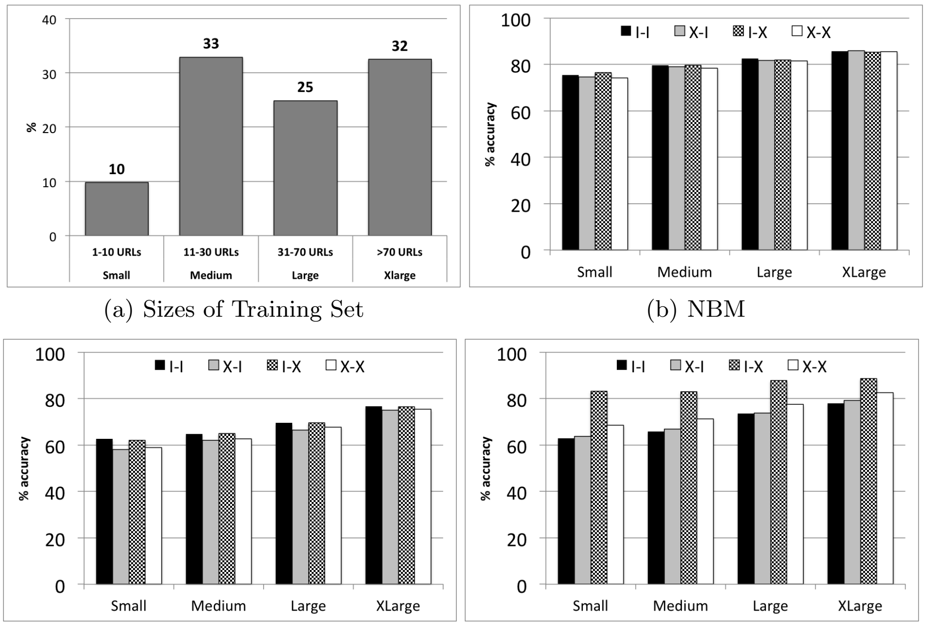
Fig. 3. Average accuracy of each combination of training and test set when $k = 3$

curation time with a success rate of 80% in terms of the correct topic being one of these top-3 topics. For the 20% of cases where the correct topic is not present the user can expand the list for more choices.

5.4 Accuracy by Topic Maturity

A factor that is likely to influence recommendation accuracy is the maturity of the topic. Intuitively, mature topics, which contain many URls, are likely to provide a richer index/model for recommendation since they provide more training data. To test this we split the training data into topics of four different sizes to evaluate small topics (5-10 URLs), medium topics (11-30 URls), large topics (31-70 URLs) and very large topics (70+ URLs); see Fig. 4(a) for the relative number of topics of different sizes in the Scoop.it dataset. Next we replayed the above evaluation for the different feature conditions and recommendation strategies. Fig. 4(b-d) show the average accuracy results for recommendation lists of size 3.

Each graph focuses on a particular recommendation strategy and shows average accuracy bars for each of the 4 feature conditions across the 4 different topic sizes. We can see that in general there is a gradual increase in recommendation accuracy as topics increase in size. For example, NBM delivers accuracy rates of approximately 75% for small and medium topics but achieves accuracies of about 85% for the largest topics. Interestingly, we can see that once again a significant accuracy benefit accruing to $TFIDF$ when using extrinsic features for the page query. Even for small topics $TFIDF$ delivers accuracy rates in excess of 80% when extrinsic features are used for the page query. This is important for two reasons. First, it suggests that strong recommendation accuracy can be achieved even for small topics, which will be particularly important in a new-user context. In addition, the $TFIDF$ approach offers certain advantages when it comes to scaling. As topics grow and mature new pages can be incrementally added to



(a) Sizes of Training Set (b) NBM

(c) SVM (d) TFIDF

Fig. 4.

the $TFIDF$ index compared to a more computationally expensive re-training that may be necessary when it comes to updating classification models.

5.5 Accuracy by Topic Count

As mentioned earlier, *Popularity* (and also *Random*) become competitive for recommendation lists of size 3 or more. This is not surprising since the average count of topics across all users is only 6.5 and so the task of selecting the right topic should be easier as we go towards this average with our recommendation list size. To investigate this, we computed recommendation accuracies across four different user groups. The results are presented in Table 1 for each of the 3 recommendation techniques (NBM, SVM and $TFIDF$) averaged across all feature conditions (**I-I**, **X-I**, **I-X**, and **X-X**). We can see that the performance of $TFIDF$ for the **I-X** feature condition is not only consistently better than all other techniques and conditions, but also that the relative benefit versus all other techniques increases with increasing topic count. For the situation of 15+ topics, $TFIDF$ in combination with the **I-X** feature condition is still well above 70% accuracy, while all other techniques and conditions are way below 60%.

Table 1. Accuracy vs. users' topics size when $k = 3$

#topics	Popularity	NBM				SVM				TFIDF			
		I-I	**X-I**	**I-X**	**X-X**	**I-I**	**X-I**	**I-X**	**X-X**	**I-I**	**X-I**	**I-X**	**X-X**
5	75.7	87.1	86.3	87.1	86.3	73.9	67.7	73.8	70.5	77.5	78.3	**90.4**	83.0
6 - 10	69.0	79.6	79.6	79.8	79.2	66.3	66.5	66.5	66.3	69.5	70.2	**84.5**	73.7
11 - 15	70.2	73.6	73.6	72.4	72.3	62.6	66.7	61.1	60.7	65.0	57.0	**73.7**	59.6
15+	60.0	54.6	53.2	52.4	50.3	42.6	50.0	42.8	40.7	53.4	53.8	**72.9**	56.4

6 Conclusions

Content curation has become an important part of the modern web. Its wide range of tools and services have evolved to help users to curate compelling collections of content. In our work we are interested in supporting the curation process and in this paper we have focused on how we can use ideas from recommender systems to automatically or semi-automatically assign newly curated content to the right collection for a user, thereby reducing an important source of friction from typical curation workflows. We have described a number of different approaches to making collection recommendations and explored different ways to represent curated content at the feature-level. These approaches have been evaluated on live curation data from the popular Scoop.it service and we have demonstrated that it is possible to make recommendations that are reliable enough to be useful in practice. In this paper we have focused on the assignment task in curation, which is of course just one place where improved support might be provided to users. In the future we will consider ways in which other curation tasks may be similarly enhanced including the recommendation of new topics to follow, recommending users to share collections or content with, recommending better tags for content or collections etc.

Acknowledgments

This work is supported by Science Foundation Ireland under grant 07/CE/I1147, Ministry of Higher Education Malaysia and Universiti Teknikal Malaysia Melaka.

References

1. Attardi, G., Gullì, A., Sebastiani, F.: Automatic web page categorization by link and context analysis. In: Proceedings of THAI. vol. 99, pp. 105–119 (1999)
2. Baykan, E., Henzinger, M., Marian, L., Weber, I.: A comprehensive study of features and algorithms for url-based topic classification. ACM Transactions on the Web (TWEB) 5(3), 15 (2011)
3. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext categorization using hyperlinks. In: ACM SIGMOD Record. vol. 27, pp. 307–318. ACM (1998)
4. Duh, K., Hirao, T., Kimura, A., Ishiguro, K., Iwata, T., Yeung, C.: Creating stories: Social curation of twitter messages. In: Sixth International AAAI Conference on Weblogs and Social Media (2012)

5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11(1), 10–18 (Nov 2009)
6. Halpin, H., Robu, V., Shepherd, H.: The complex dynamics of collaborative tagging. In: Proceedings of the 16th international conference on World Wide Web. vol. 21, pp. 1–220. Citeseer (2007)
7. Hatcher, E., Gospodnetic, O.: Lucene in action. Manning Publications (2004)
8. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. Knowledge Discovery in Databases: PKDD 2007 pp. 506–514 (2007)
9. Kibriya, A., Frank, E., Pfahringer, B., Holmes, G.: Multinomial naive bayes for text categorization revisited. AI 2004: Advances in Artificial Intelligence pp. 235–252 (2005)
10. Liu, S.B.: Trends in distributed curatorial technology to manage data deluge in a networked world. The European Journal for the Informatics Professional 11(4), 18–24 (2010)
11. Lu, Y.T., Yu, S.I., Chang, T.C., Hsu, J.Y.j.: A content-based method to enhance tag recommendation. In: Proceedings of the 21st international jont conference on Artifical intelligence. pp. 2064–2069. Morgan Kaufmann Publishers Inc. (2009)
12. Maarek, Y., Ben Shaul, I.: Automatically organizing bookmarks per contents. Computer Networks and ISDN Systems 28(7), 1321–1333 (1996)
13. Milicevic, A.K., Nanopoulos, A., Ivanovic, M.: Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions. The Artificial Intelligence Review 33(3), 187–209 (2010), http://search.proquest.com/docview/198036064?accountid=14507
14. Pujari, M., Kanawati, R.: Tag recommendation by link prediction based on supervised machine learning. In: Sixth International AAAI Conference on Weblogs and Social Media (2012)
15. Saaya, Z., Schaal, M., Coyle, M., Briggs, P., Smyth, B.: A comparison of machine learning techniques for recommending search experiences in social search. Research and Development in Intelligent Systems XXIX p. 195 (2012)
16. Saaya, Z., Schaal, M., Coyle, M., Briggs, P., Smyth, B.: Exploiting extended search sessions for recommending search experiences in the social web. Case-Based Reasoning Research and Development pp. 369–383 (2012)
17. Scoble, R.: The seven needs of real time curators.e. http://scobleizer.com/2010/03/27/the-seven-needs-ofreal-time-curators/ (2010), [Online; accessed 06-Dec-2012]
18. Shen, D., Sun, J., Yang, Q., Chen, Z.: A comparison of implicit and explicit links for web page classification. In: Proceedings of the 15th international conference on World Wide Web. pp. 643–650. ACM (2006)
19. Smyth, B., Briggs, P., Coyle, M., O'Mahony, M.P.: Google shared. a case-study in social search. In: User Modeling, Adaptation and Personalization. pp. 283–294 (2009)
20. Staff, C., Bugeja, I.: Automatic classification of web pages into bookmark categories. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 731–732. ACM (2007)
21. Xu, J., Croft, W.B.: Query expansion using local and global document analysis. In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 4–11. SIGIR '96, ACM, New York, NY, USA (1996)