Vetting the Links of the Web

Na Dai Brian D. Davison
Department of Computer Science & Engineering
Lehigh University
Bethlehem, PA 18015 USA
{nad207,davison}@cse.lehigh.edu

ABSTRACT

Many web links mislead human surfers and automated crawlers because they point to changed content, out-of-date information, or invalid URLs. It is a particular problem for large, well-known directories such as the dmoz Open Directory Project, which maintains links to representative and authoritative external web pages within their various topics. Therefore, such sites involve many editors to manually revisit and revise links that have become out-of-date. To remedy this situation, we propose the novel web mining task of identifying outdated links on the web. We build a general classification model, primarily using local and global temporal features extracted from historical content, topic, link and time-focused changes over time. We evaluate our system via five-fold cross-validation on more than fifteen thousand ODP external links selected from thirteen top-level categories. Our system can predict the actions of ODP editors more than 75% of the time. Our models and predictions could be useful for various applications that depend on analysis of web links, including ranking and crawling.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; H.3.5 [Information Storage and Retrieval]: Online Information Services—*Web based services*

General Terms: Algorithms, Experimentation, Measurements

Keywords: Link analysis, Web decay, ODP, web archives

1. INTRODUCTION

The Web is in constant flux. Page content and links are changed, added, and removed from the Web on a continuous basis [2, 3, 5, 6]. This presents a significant challenge to the fundamental technology of the web—hypertext—as the target of a link, if it exists at all, is often not the same as when the link was first created. About two-thirds of web pages change their content each year. As a result, many links on the Web are obsolete. At best, such links point to forgotten sites that have long been abandoned. At worst, such links point to sites whose original purpose has been subverted, perhaps in an attempt to exploit the value of links to search engine ranking.

High-profile sites must be especially vigilant (and therefore should be carefully maintained) as links from such sites are particularly valuable because of visitor traffic or reputation flow or both. Most existing work on web link maintenance involves identifying error-generating pages by developing tools to find invalid links, to automatically correct links that lead to file not found errors [10] and to find irrelevant (and possibly annoying or offensive) spam pages [12, 17]. However, none consider that the link context changes over time. That is, the two end points of a link may change in different "directions" rendering the link inappropriate.

Therefore, in this paper we introduce a new task—that of vetting web links. Our goal is to better understand which factors are important for detecting out-of-date page citations such that we can build a link vetting system, from which web content providers would benefit tremendously from automatically identifying whether links on their pages need to be re-examined. In particular, large, well-known sites like the Yahoo! directory [18] and the dmoz Open Directory Project [14] involve many editors to maintain the accuracy and representativeness of their categories by manually and periodically checking the quality of the external pages to which they link. Our system can make their maintenance work less expensive.

We state the link vetting problem formally as follows.

Definition 1. A link l is defined as an **outdated** link **iff** its web provider or link maintainer removes or should remove it from the original source web page. Otherwise, we consider it as a fresh link.

We assume the link was fresh when it was first created. Changes in both source page and target page causes the link to decay over time. Therefore, we believe that our definition of an outdated link covers the basic aspects of out-of-date links. However, we only describe a binary judgment on whether a link is outdated or not.

Our link vetting problem is defined as follows:

Definition 2. **The link vetting problem**: Suppose a link l was created at time t_0; determine whether this link is **outdated** at time point t_1, given $t_0 < t_1$.

This problem can have two variants by considering the current time point t_2. When $t_2 = t_1 > t_0$, this problem is to identify the outdated links. When $t_1 > t_2 > t_0$, the problem turns to predict whether a link will become out-of-date at a given future time point. We mainly focus on the former variant in this work; however, we also investigate the models' predictability in Section 3.

2. THE LINK VETTING SYSTEM

We consider the task of determining whether links are outdated as a classification problem. Our link vetting system (LVS) builds up a general classification model by combining multiple groups of temporal features extracted from local and global historical information, such as content, topics, etc. These features can represent basic information about link context change over time.

CIKM'09, November 2–6, 2009, Hong Kong, China.

Notation	Meaning
d	A target page
s	A source page
t_0	the time point that the link (d–>s) was created
t_n	the time point that the link (d–>s) is outdated and removed
T_i	the i^{th} time interval/unit
$C(d)_{T_i}$	the snapshot of a target page in the i^{th} time interval/unit
$C(s)_{T_i}$	the snapshot of a source page in the i^{th} time interval/unit

Table 1: Notation definitions.

2.1 Using historical snapshots

In order to map link context changes onto a time axis, we discretize time into multiple intervals. We use a page snapshot at one time point to represent the page situation during the time interval which covers that time point. While the time interval could be measured by any time units, we use **Year** as our basic unit to measure link context change. We base our work on the assumption that a link was fresh when it was first created. By comparing the snapshot for each time unit with that for the first time unit and successive time units, we can know how the page changes over the time units, and potentially know whether such changes make the links between pages stronger or weaker.

Table 1 defines some notation. We use the following metrics to measure the difference between two snapshots:

- **Jaccard Coefficient**: $JC(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|}$
- **Element Removal Coefficient**: $ERC(A,B) = \frac{|A-B|}{|A|}$
- **Element Addition Coefficient**: $EAC(A,B) = \frac{|B-A|}{|A|}$
- L_1 **Distance**: $L_1(\overrightarrow{u}, \overrightarrow{v})) = \sum_{j=1}^{m} |u_j - v_j|$

where A and B are two sets which contain some elements, and $\overrightarrow{u}$ and $\overrightarrow{v}$ are two vectors with m dimensions.

2.2 Features

Our features are extracted from the changes of title, meta information, content, link, topicality (defined in Section 2.2.2) and the fraction of words in our predefined list. These changes are determined by the comparison between different historical snapshots of both source pages and target pages. Most of these features emphasize the comparison between snapshots. The main feature list is listed in Table 2.

2.2.1 Local Features

Local features are organized to represent the characteristics of link context changes over time.

Features based on title, meta information and content. We use the downloaded snapshots for each target page and collect the terms in different page fields, such as title, keywords, description and content. We use JC, ERC and EAC (see Section 2.1) in each field as our features to compare the similarity between two snapshots. For meta information recorded by the IA, we also check finer fields within it, including HTML content base, returned HTTP status code and so on, which reflect the state of the target page.

Features based on time measures. This group of features represents the time information hidden in the content of historical snapshots. We treat "Year" as the atomic time unit, and extract all the time information which is presented by year, such as 1999, 2000 and so on. Combining the last-modified time in meta information (about 60% of the pages show last-modified time in returned HTTP information), we calculate a time-based distribution with respect to each snapshot. We use the L_1 distance between two compared snapshots as our features to represent time evolution.

Feature	Description
Title, meta information and content	
T/M/C_SC	$JC(C(d)_{T_1/T_{i-1}}, C(d)_{T_i})$, $ERC(C(d)_{T_1/T_{i-1}}, C(d)_{T_i})$, $EAC(C(d)_{T_1/T_{i-1}}, C(d)_{T_i})$ for title/meta/content field
M_UF	Average update frequency of $C(d)_{T_i}$ and $C(s)_{T_i}$
Time measures	
TI_CT	Whether $C(d)_{T_i}$ contain time information
TI_SA	Whether $C(d)_{T_1/T_{i-1}}$ and $C(d)_{T_i}$ have the same distribution on time information
TI_TD	$L_1(C(d)_{T_1/T_i}, C(d)_{T_i})$ of the distribution on time information
Global bi-gram and tri-gram lists	
GL_INF/TRF	the fraction of $C(d)_{T_i}$'s bi/tri-gram words that are in the global "incomplete"/"trustworthy" list
Category	
CA_TOD	$L_1(C(d)_{T_1/T_{i-1}}, C(d)_{T_i})$ of the topic distribution under predefined taxonomy
CA_AC	$L_1(C(s)_{T_1/T_i}, C(d)_{T_i})$ between the topic distribution of the extended anchor text on page s and the content of page d
Outlinks and anchor text	
OA_SC	$JC(C(d)_{T_1/T_{i-1}}, C(d)_{T_i})$, $ERC(C(d)_{T_1/T_{i-1}}, C(d)_{T_i})$, $EAC(C(d)_{T_1/T_{i-1}}, C(d)_{T_i})$ for outgoing links and "frameset" links
OA_NL	The number of "mailto"/"frameset" links in $C(d)_{T_i}$
Topicality inferred from pre-computed language models	
LM_DIST	The distance of $\pi_{C(d)_{T_i},j}$ and $p(\theta_j)$ in the j^{th} hidden topic of the outdated/fresh link context cluster determined by the topic of $C(s)_{T_i}$

Table 2: Main features for each group.

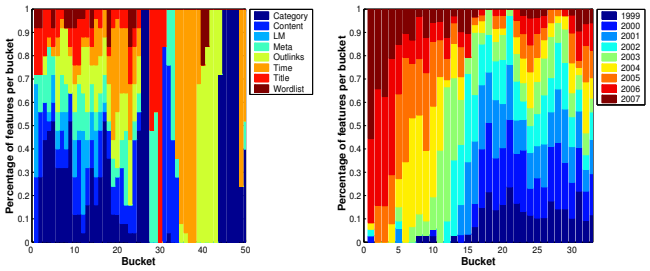
Features based on global bi-gram and tri-gram lists. We manually build up two lists composed of bi-grams and tri-grams. One list records 45 representative phrases which show that the page content is still incomplete (e.g., "under construction", etc.). The other records 45 phrases which relate to the professionality or trustworthiness of the pages/snapshots (e.g., "all right reserved", etc.). The statistics for each of these records the presense (or absence) of that phrase in a page snapshot under consideration. This group of features is based on the comparison of the statistics between snapshots.

Features based on category. Based on the twelve selected top-level ODP topics, we use a well-known naive Bayes classifier "Rainbow" [11] trained on the texts of 1000 randomly selected documents per class from the ODP. We then classify each page snapshot by the trained classifier. For each snapshot, we produce a topic distribution vector, which presents the normalized probability that the snapshot belongs to each topic. This group of features is calculated from the L_1 distance of topic distributions between two compared snapshots, either between source and target page snapshots in the same time unit, or between the same target page in snapshots in two different time units.

Features based on links. This group of features checks the consistency of the outgoing links between two compared snapshots. We also compare the information about "mailto:", which may reflect the page's trustworthiness to some degree. In addition, the change of frameset information gives some clue about the probability that a page becomes a redirection or cloaking page.

2.2.2 Global Features

Global features use the knowledge based on characteristics of members of a calculated group/cluster. They reflect the background



(a) Feature source type per bucket, ordered by information gain.

(b) Feature source year per bucket, ordered by information gain.

Figure 1: Distribution of feature discriminability with respect to their category times and source year.

of the whole group. The deviation of features from such a background will make these extracted features more discriminative.

Features based on topicality inferred from pre-computed language models (LMs). Given a time point t_1, we calculate separate language models based on the content corresponding to outdated link context and fresh link context respectively. Our hope is that we train a set of outdated topics from outdated link contexts, and a set of fresh topics from fresh link context for each time point, and track how close the topical distribution of an unseen link context is to these two sets of topics and how these distances change over time.

We first cluster the link contexts into different groups, with each sharing some similar characteristics. The reason is that the outdated topics and/or the fresh topics should be different among groups. Hence, we group link contexts according to the category of source pages or the anchor text on them. For each group, we compute separate sets of latent topic-specific language models from all the snapshots of the target pages within the link contexts in that group, with one calculated by using those involved in outdated link contexts and the other one by the target page snapshots in fresh link contexts. Define w as a word in the dictionary. Define $\theta_1, \ldots, \theta_k$ as unigram language models for k topics. $\pi_{d,j}$ is the mixture weight ($\sum_{j=1}^{k} \pi_{d,j} = 1$). We use the EM algorithm in pLSA [7] to estimate the parameters $\pi_{d,j}$ for each snapshot and $p(w|\theta_j)$.

Next, we define the topic centroid of the outdated or fresh link contexts of the i^{th} cluster/group as $\frac{1}{|C_{i,s'}|}\sum_{d \in C_{i,s'}} \pi_{d,j}$, where $s' \in \{\text{fresh, outdated}\}$ and $C_{i,s'}$ represents the corpus composed of all the page snapshots involved in the link contexts of the i^{th} cluster. We use the topic centroid as our estimation of $p(\theta_j)$. Given a unseen link context, we estimate $\pi_{d,j}$ of involved page snapshot content by the pre-computed language models. Bayes' rule infers

$$p(\hat{\theta}_j|w) = \frac{p(w|\theta_j)p(\hat{\theta}_j)}{p(w)} = \frac{p(w|\theta_j)p(\hat{\theta}_j)}{\sum_{j'=1}^{k} p(w|\theta'_j)p(\hat{\theta}'_j)}$$

where $p(\hat{\theta}_j|w)$ unravels the contribution of w to the j^{th} hidden subtopic. Therefore, the $\hat{\pi}_{d,j}$ can be given by $\frac{1}{|d|}\sum_{w \in d} p(\hat{\theta}_j|w)$. We can use $|\hat{\pi}_{d,j} - p(\hat{\theta}_j)|$ to represent the distance of the j^{th} subtopic distribution within a link context from the background of its clusters. Thus, this group of features tracks how far the hidden topic distribution of a link context is from the outdated/fresh background distribution and how prominent a link context is on a particular fresh/outdated subtopic when considering the background in different time units.

2.3 Classification Algorithms

We explore a variety of classification algorithms with respect to their capability to detect outdated links. Specially, we select 37 classifiers implemented in the Weka toolkit [15] and evaluate them on the proposed task in Section 3. Our selected classification algorithms include multiple classifiers in the decision tree family, support vector machine, NaiveBayes, rule generators, boosting and other meta-learning methods. We believe these classification algorithms can represent the state-of-the-art. Thus, by exploring these classification algorithms, we find which classification algorithms are suitable for this task and how well LVS can be generalized.

3. EXPERIMENTS

3.1 Data sets

We use the ODP data set, which is based on the external pages cited by dmoz Open Directory Project and corresponding historical snapshots provided by the Wayback Machine service offered by the Internet Archive [8].

By exploring the historical ODP RDF file and content file, we get some statistics about the external link removal. In order to set up a realistic link vetting task, we define our task as to *determine whether the links to selected external pages will be removed by 2007 (decisions by ODP editors)*. We first use the 2008 ODP RDF file to extract active categories. There are 756,439 categories in total. We randomly select 15,910 external pages among these categories, which have complete historical snapshots from the year in which they are first observed in the ODP directory to the year 2007 as our data set for training and testing the classifier. The removed external pages (outdated external links) are labeled as positive examples, while those remaining are negative.

3.2 Evaluation

We first select the most suitable feature set for this task. By basing our feature selection on Information Gain (IG), we can understand which features are more discriminative for the task. We sample from a larger data set, excluding the 15910 examples, and perform feature selection on it. The distributions of feature information gain with respect to their category and time units are depicted in Figure 1. The x-axis is the buckets from the ranking of feature IG values. From Figure 1(a), the most discriminative features focus on content, LMs and title related features. Especially the top 10 features are all from the LM group covering from 2006 to 2007. The category and meta related features gradually dominate the bucket with the decrease of IG value rankings. Interestingly, the features about outgoing links don't show good discriminability in this task. Figure 1(b) demonstrates that the feature discriminability highly correlates with the time point from which the features formed. Earlier features show poorer discriminability.

We tried 27 different feature set sizes, and trained the 37 classifiers provided by Weka. We found most of these classifiers get their best performance when using the top 200 discriminative features selected by IG values. Hence, we choose to present the classification performance based on these 200 features. The comparison among multiple classifiers are all based on five-fold cross-validation, shown in Table 3. The top four classifiers for this task are EnsembleSelection, Bagging, DecisionTable, and REPTree, where EnsembleSelection gets the best performance on all the four metrics we used. We also list the performance of some traditional classifiers, including C4.5 decision tree. The RandomForest classifier in decision tree family can achieve a F-measure of 0.749 while NB has the lowest F-measure and accuracy. We also

Classifier	including 2007		excluding 2007	
	F-Meas.	Accu.	F-Meas.	Accu.
LVS_EnsembleSelection	0.782	0.764	0.686	0.663
LVS_Bagging	0.771	0.753	0.689	0.665
LVS_DecisionTable	0.756	0.740	0.692	0.687
LVS_REPTree	0.756	0.740	0.668	0.648
LVS_RandomForest	0.749	0.717	0.701	0.633
LVS_J48	0.738	0.710	0.675	0.653
LVS_Logistic	0.722	0.699	0.679	0.645
LVS_SMO	0.717	0.699	0.670	0.634
LVS_AdaBoostM1	0.702	0.698	0.683	0.673
LVS_NaiveBayes	0.542	0.596	0.437	0.555
404Checker	0.155	0.459		
Default (Majority)	0.714	0.556		
Random	0.526	0.500		

Table 3: Classification performance results on ODP dataset.

found the performance of many classifiers is quite close to the highest one. The deviation of the top 4 classifiers on F-measure is only 0.0127, which demonstrates that the system can provide a general classification model independent of the specific classifier.

We explore the predictability of classification models by removing all the features involved in the information from 2007 (Table 3, rightmost columns). From Table 3, all classifiers reveal an inferior performance to those trained by all the features on both F-measure and accuracy. In particular, Adaboost shows the best capability of prediction of future outdated links, with only a 2.7% decrease in F-measure performance. In contrast, NB shows the worst predictability performance since its F-measure score decreases 19.4%.

4. RELATED WORK

Web link maintenance involves significant manual labor to detect outdated links based on complex and diverse criteria. Existing research work on web link maintenance, using only a snapshot of the current web, typically focuses on one specific criteria for detecting pages/links which violate it. Tools such as W3C Link Checker [16] can automatically identify error generating pages. Some recent research work extends this task by automatically correcting broken links on the web [9, 10].

Some researchers extracted useful temporal data for web information retrieval tasks. Nunes [13] identifies temporal web evidence by using two classes of features based on individual information and global information. They also propose several sources of temporal web evidence, including document-based and web-based evidence, which can be utilized in improving multiple retrieval tasks. Researchers at Google filed a patent [1] on using historical information for scoring and spam detection. Berberich et al. [4] propose two temporal link analysis ranking algorithms which incorporate pages' temporal freshness (timestamps of most recent updates) and activity (update rates), and improves ranking performance. Yu et al. [19] incorporate temporal factors to overcome the problem that traditional link analysis approaches favor old pages by introducing a temporal weight into the PageRank algorithm, which decreases exponentially with citation age.

5. CONCLUSION

Many web links reflect choices and information that, while valid at time of link creation, are now woefully out-of-date. In this work we have proposed a new web mining task of vetting web links. As an initial attempt to satisfy this task for links of the ODP directory, we have presented a classification performance comparison among a variety of state-of-the-art classifiers on this task, trained by temporal features extracted from the historical link context, including content, category, and link-based information from historical snapshots of both source and target pages. Our proposed system is able to achieve a F-measure of 0.782 when compared to ODP editor removal actions. This evidence suggests that, with sufficient, coherent archival data, it is possible to vet automatically many links of the web, and points the way to new tools to help maintain the accuracy of the Web, benefitting various applications that depend on analysis of web links beyond web site maintenance, including crawling, ranking, and classification.

Acknowledgments

This work was supported in part by a grant from the National Science Foundation under award IIS-0803605 and an equipment grant from Sun Microsystems. We also thank Dennis Fetterly, Liangjie Hong, Gordon Mohr and Xiaoguang Qi for helpful feedback.

6. REFERENCES

[1] A. Acharya, M. Cutts, J. Dean, P. Haahr, M. Henzinger, U. Hoelzle, S. Lawrence, K. Pfleger, O. Sercinoglu, and S. Tong. Information retrieval based on historical data. United States Patent 20050071741, USPTO, Mar. 2005.

[2] R. A. Baeza-Yates, C. Castillo, and F. Saint-Jean. Web structure, dynamics and page quality. In M. Levene and A. Poulovassilis, editors, *Web Dynamics: Adapting to Change in Content, Size, Topology and Use*, pages 93–115. Springer-Verlag, Berlin, 2004.

[3] Z. Bar-Yossef, A. Z. Broder, R. Kumar, and A. Tomkins. Sic transit gloria telae: Towards an understading of the web's decay. In *Proc. 13th Int'l World Wide Web Conf.*, pages 328–337, May 2004.

[4] K. Berberich, M. Vazirgiannis, and G. Weikum. Time-aware authority ranking. *Internet Mathematics*, 2(3):301–332, 2005.

[5] J. Cho and H. Garcia-Molina. Estimating frequency of change. *ACM Transactions on Internet Technology*, 3(3):256–290, Aug. 2003.

[6] D. Fetterly, M. Manasse, M. Najork, and J. L. Wiener. A large-scale study of the evolution of web pages. *Software: Practice and Experience*, 34(2):213–237, 2004.

[7] T. Hofmann. Probabilistic Latent Semantic Analysis. In *Proc. of the 15th Annual Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 289–296. Morgan Kaufmann, July 1999.

[8] The Internet Archive, 2009. http://www.archive.org/.

[9] J. Martinez-Romo and L. Araujo. Recommendation system for automatic recovery of broken web links. In *IBERAMIA '08: Proceedings of the 11th Ibero-American conference on AI*, pages 302–311, Berlin, Heidelberg, 2008. Springer-Verlag.

[10] J. Martinez-Romo and L. Araujo. Retrieving broken web links using an approach based on contextual information. In *Proc. 20th ACM Conf. on Hypertext*, pages 351–352, 2009.

[11] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow, 1996.

[12] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proc. 15th Int'l Conf. on the World Wide Web*, pages 83–92, May 2006.

[13] S. Nunes. Exploring temporal evidence in web information retrieval. In *BCS IRSG Symposium: Future Directions in Information Access (FDIA)*. British Computer Society, 2007.

[14] The dmoz Open Directory Project (ODP), 2009. `http://www.dmoz.org/`.

[15] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition, June 2005.

[16] World Wide Web Consortium. W3C link checker. Online at `http://validator.w3.org/checklink`, visited 2 November 2008, 2006.

[17] B. Wu and B. D. Davison. Identifying link farm spam pages. In *Proc. 14th Int'l World Wide Web Conf.*, pages 820–829, May 2005.

[18] Yahoo!, Inc. Yahoo! directory. `http://dir.yahoo.com/`, 2009.

[19] P. S. Yu, X. Li, and B. Liu. On the temporal dimension of search. In *Proc. 13th Int'l World Wide Web Conf.*, pages 448–449, May 2004.