

Filtering Spam Using Search Engines

Oleg Kolesnikov, Wenke Lee, and Richard Lipton
ok,wenke,rjl}@cc.gatech.edu

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

Abstract

Spam filtering approaches constantly face new evasion techniques attempted by the spammers. For example, text-based approaches, including those using Bayesian classifiers on email messages, may be evaded by sending text in images or minimizing the text of an ad and shifting most details to a web site by adding a link. The links (URLs) spammers typically use as a feedback mechanism present perhaps the only piece of "immutable" information in a spam message because each URL must be precisely spelled out to link to a Web page¹. In this paper, we present an approach that automatically mines search engines, such as Google and Yahoo, for category information about URLs in email messages. We then use the information to filter out messages with URLs that point to web sites a user is not interested in seeing. If the search engines cannot provide the category information, our approach uses a Bayesian classifier on the Web site details retrieved from search engines. Our approach is different from approaches that tokenize URLs or use a blacklist of URLs [E.K04, R.04]. We describe the implementation of our approach and discuss lessons in its deployment. Results of our experiments on real-world email (and spam) data show that the accuracy of our approach is comparable or better than that of spamassassin. As spammers continue to develop new techniques to evade text-based filtering, combinations of spam filtering solutions are likely to be employed. Our approach can make such combined filtering solutions more accurate because it relies on an additional information source making it harder for spammers to evade classification.

1 Introduction

Most spam filtering approaches in use today rely on the text of messages for classification. In many cases, these approaches can be easily evaded by spammers. For example, a spammer can shift most of the details of an ad in a message to a Web site, to evade filters. The resulting spam message can be very small, consisting of neutral words (e.g. the ones that commonly occur in legitimate e-mail for most users) and a link to a web site. Spammers can also put ads in images.

We present an approach that complements the existing text-based filtering methods to address the problem. Our approach uses a combination of techniques. The key idea is to filter spam by focusing on messages that contain URLs pointing to Web sites a user is not interested in. We determine categories for URLs using search engines, then use the categories and, if necessary, Bayesian classifier statistics on Web site contents, to define and match users' interests.

One of the advantages of our approach is that it uses two additional sources of information: search engines/website categories and Web site content to improve spam filtering. In our system, users are the ones in control of the kinds of mail and advertisements that get into their mailbox. The only messages with URLs

¹The feedback mechanism must be easy to use. Of course, spammers may require users to perform extra work and form a URL manually by pasting together several pieces of text. However, this would be a self-defeating strategy for a spammer because the harder it is for users to act on messages, the fewer of them will respond.

that pass our filter are the ones that point to sites that are of interest to the users. This helps users regain control of their mailboxes while at the same time encourages a switch to targeted advertisements, which could be a win-win situation for both the users and the legitimate advertisement industry.

As pointed out by Androutsopoulos et al. [IJKC00], a good spam filtering approach must not only have significantly low false positive and negative rates, but also take into account that it is far worse to misclassify a legitimate message than spam. We specifically focus on this requirement in our evaluation. As we show later, our approach satisfies the requirement and its false positive rate is comparable or better than that of spamassassin. We also show that our approach is effective at stopping spam and has a close to zero false negative rate for spam messages that contain URLs.

The remainder of the paper is structured as follows. In Section 2 we give an overview of related work. In Section 3 we describe our approach. Section 4 discusses implementation details. Section 5 evaluates our approach against spamassassin. We conclude in Section 6.

2 Related Work

Filtering is currently the standard means to stop e-mail spam. Most of the current approaches and products use content-based filtering. The content-based approaches include whitelisting, blacklisting, Bayesian classification, heuristics-based filtering [SM03, Cor03], and collaborative filtering [V.V04]. Other classes of filtering approaches include challenge-response, MTA/Gateway filtering (Tarproxy [M.04], greylisting [E.03] etc.), and micropayments [B.T03].

Some content-based approaches rely exclusively on message headers, namely automatic and Bayesian whitelisting [E.02], blacklisting (MAPS, RBL), and others. Their main disadvantage is that spammers can fake a message header. Also, legitimate domains can easily get blacklisted.

Other content-based approaches rely on words in messages and statistics about them. For example, simple *Bayesian Machine Learning* approaches, introduced by Duda et al [DR73] and applied to spam filtering by Sahami et al. [SDHH98], use the conditional probability of words occurring in spam and legitimate messages. The evaluation by Androutsopoulos et al. [IJKC00] showed that these approaches are viable but not without shortcomings. One of the advantages of the approaches is that they are user-specific and may offer low false positive and negative rates after sufficient training with the current generation of spam. The disadvantage is that the training takes time. Also, training statistics cannot be easily re-used or combined for different users. As mentioned earlier, another disadvantage is that spammer may be able to evade the approaches by using common words and shifting the details of an ad to a web-server or by including the text of an ad in an image.

Exchange Intelligent Filter [Cor03] is one of the content-based approaches that uses heuristics based on submissions of thousands of hotmail users. *Spamassassin* is another approach that uses heuristics [SM03]. It calculates a score for every message based on a manually extracted set of features (rule base). The disadvantage of the heuristics-based methods in general is that they are very ad hoc and need to be updated regularly. The updates can be as complex as the filters themselves [J.03]. The current version of spamassassin attempts to deal with the problem by including a number of plugins that support different methods (including Bayesian filtering) to improve the score calculation. One of the plugins used by spamassassin that is related to our approach is SURBL/SpamCopUri [E.K04]. It blocks messages by using a blacklist of URLs. The blacklist is created based on the spam submissions received from users. One disadvantage of this method is that it takes some time for spam to be reported. By the time an update is received, it is already too late. Also, it is very easy for spammers to change the text of a URL and have it point to the same content.

Our approach is different from SURBL in three ways. First, the definition of spam in our approach is personalized. Every user can have a different set of categories he or she is interested in. The information can be easily combined and shared among users. Second, in our system the presence of a URL in a message is a good indicator of spam. Third, in addition to the text of a URL, our approach may also use the information

from the web site pointed to by the URL. For similar reasons, our approach also is different from the URL module used by Brightmail [K.04].

3 Description of Approach

Without a feedback mechanism, spam is useless. The objective of our approach is to take away URLs from spammers as an effective feedback mechanism. The key idea of our approach is to filter messages that contain URLs pointing to sites a user is not interested in. We do so by combining several techniques, namely using categories of URLs retrieved from search engines, dynamically classifying the contents of the Web sites spammers use for feedback, and keeping history of the URLs seen in different types of messages. We describe the techniques in more detail later in this section.

3.1 Types of URLs

We distinguish two types of URLs in messages. The first type comprises *categorized* URLs. These are the URLs present in at least one of the Web directories we query, namely Yahoo Directory [Yah04] and Google Directory/Open Directory Project [dmo04]. To illustrate, a URL may be categorized as Arts_and_Humanities/Philosophy/Philosophers, which is a category from the Yahoo Directory². The second type of the URLs comprises *uncategorized* URLs. These are the URLs that are not listed in any of the Web directories we use.

3.2 Configuration and Training

Because we view spam as a user-specific entity, our approach can have different configurations for different users. By default, the configuration is performed automatically but it can also be easily performed manually. The per-user configuration consists of two main parts:

- List of acceptable categories $A_{categories}$, see appendix for an example.
- List of regular expressions A_{regex} for acceptable URLs e.g. .edu, .mil, and so forth.

Our approach is configured/trained as follows. First, to obtain the list of acceptable categories, we extract URLs from legitimate messages in a user's mailbox. We then check a local cache for categories and, if necessary, find categories for URLs by querying Web directories using search engines. For uncategorized URLs in the user's mailbox, we remove duplicates and add regular expressions for every unique URL to A_{regex} . We also retrieve the contents of the Web sites pointed to by the URLs from search engine caches. We then train a simple Bayesian classifier on the content.

In addition to processing the user's mailbox, we can also load URLs from other sources, such as *Bookmarks* list from the user's browser. The extracted URLs that do not have a category are sorted by frequency. Then, the regular expressions for all or most frequently occurring URLs are automatically added to A_{regex} .

After the automatic configuration, users may edit and verify both $A_{categories}$ and A_{regex} to make sure their interests are properly reflected. Adding categories manually is very easy. For example, users interested in data mining can search for "data mining conferences" and then add the category Computers/Software/Databases/Data_Mining/Events³.

In addition to automatic and manual configuration options, our approach offers three pre-defined profiles for academia, business, and home users. An example of the *Academia* profile is given in the Appendix.

²Yahoo Directory has 13 top-level categories plus a *Regional* category. The top-level categories are: Arts and Humanities, Business and Economy, Computers and Internet, Education, Entertainment, Government, Health, News and Media, Recreation and Sports, Reference, Science, Social Science, Society and Culture.

³To make manual editing even easier, we implemented a web-based proxy. All users need to do is to type in several keywords, then click on the categories they would like to add to $A_{categories}$.

3.2.1 Performance and Caching

Since it may take several seconds to query search engines for each URL, our approach maintains a global cache for query results. The cache contains category and other information retrieved from search engines. Entries in the cache are periodically expired to make sure the information is current. The cache is consulted prior to any search engine request to improve performance. To further improve performance, we execute search engine queries in parallel. Note that it is important to consider the possibility of Denial of Service attempts against our approach because we query search engines for URLs in messages. However, further discussion of this question is out of scope for this paper.

3.3 Mail Classification

Our classifier works as follows. When a message arrives, it is scanned for URLs. If a message does not contain URLs it is classified as legitimate. Otherwise, the classifier checks every URL in the message. It skips (removes from consideration):

- All URLs that match one or more regular expressions in A_{regex} .
- All URLs that occurred in messages previously classified as legitimate (see *Incremental Learning* in Section 4).
- All categorized URLs categories for which are in $A_{categories}$

At this point, the set of remaining URLs U_r may include only categorized URLs with categories not in $A_{categories}$ or uncategorized URLs never seen before in legitimate messages that do not match any of the regular expressions in A_{regex} .

U_r is processed as follows. For each URL $U_i \in U_r$, if U_i has a category not in $A_{categories}$, the process is stopped and the message is classified as spam. For all uncategorized U_i , the classifier loads the contents of the Web sites U_i point to from a search engine cache. It then runs a simple Bayesian classifier on the contents (the classifier has a *conservative* configuration so as to align its false positive rate with other parts of the classification algorithm.) Based on the output of the classifier, the message is classified as either legitimate or spam. If the website contents for U_i cannot be found in the search engine cache (the URLs has not been indexed yet), the classifier either retrieves the contents of the Web site directly or immediately classifies message as spam. The specific decision is based on the user's configuration.

4 Implementation details

Our filtering approach can be implemented in two ways. In a server-based implementation, mail for all users is filtered by a central server. The server keeps track of configurations for all users. In a client-based implementation, each client runs a copy of the filter independently. The main advantage of the server-based implementation is improved performance since a global cache of URL information can be used. The main advantage of the client-based implementation is better privacy protection.

Our implementation is *server-based*. It uses a RedHat Linux v9 machine running Apache with `mod_ssl`. It consists of two parts: a web-based configurator interface and a mail classifier. The mail classifier is implemented using Perl and procmail.

The implementation works as follows. Users register in our system via a web interface. They request automatic configuration/training based on their mailboxes. After the automatic part is complete, users can manually edit $A_{categories}$ and A_{regex} .

Following the configuration, whenever new mail for the users arrives on the server, the mail is automatically classified into "Inbox" and "Spam-can" IMAP folders. When users retrieve messages from the server, the messages are already classified.

Fine-tuning Our system can be fine-tuned by users simply by moving incorrectly classified messages from "Inbox" to "Spam-can" and vice versa. The misclassified messages will be automatically picked up by a checker process that is run on a nightly basis.

Google API Google offers a free licence to access the search engine API. Our mail classifier uses the licence and the API to search for the information about URLs on Google (queries are sent over SOAP). Note that the licence is limited to 1000 queries per user per day. So, for each user of our system, we have a licence that is automatically chosen and used whenever a message destined to the user needs to be processed.

Profiles As mentioned before, our system currently offers three pre-defined profiles: *Academia*, *Business*, and *Home*. Each profile has a list of categories and URLs associated with it. When registering with the system, users can simply choose one of the profiles and then edit the base values. A sample of the *Academia* profile we use is denoted in the Appendix.

5 Experimental Evaluation

To validate our approach, we conducted the following experiments. We tested SpamAssassin v.2.63 and our classifier on two e-mail corpuses. The first corpus (C_l) consisted of 3,560 manually filtered *legitimate* e-mails received by one of the authors over a period of two years. The second corpus (C_s) consisted of 1496 recent *spam* messages obtained from the spamarchive.org public spam database. The results of the comparison are summarized in Table 1.

5.1 Training

We used the first 1,191 and 493 messages from the C_l and C_s , respectively, as the training sets. Since our filter is user-specific, we used the data for a single user as the training data for legitimate messages. Spamassassin was trained using `sa-learn`, a standard utility included with the package. We used the default settings.

Our classifier was trained as described in Section 3. It learned 230 unique categories during training. The list of categories is denoted in the Appendix. It also learned several most frequently occurred URLs without categories and added regular expressions for those to the whitelist shown in the Appendix.

We found that the list of uncategorized URLs included many duplicates. Most duplicates were what we call *footer* URLs. The most frequently occurring footer URLs were those added by free webmail services, such as Yahoo! and Hotmail. Other types of footer URLs included links to personal Web pages from friends and co-workers as well as URLs added by automatic mail filtering tools, such as the Anomy Sanitizer (an automatic mail virus scanning tool) [B.04].

5.1.1 Legitimate Corpus

Spamassassin classified 56 of 2,369 (3,560-1,191) legitimate e-mails as spam (2% false positive rate). After manual inspection, we found that 11 of the messages were follow-up offers from an on-line merchant. While a generic user may consider the e-mails spam, these e-mails were not spam for the given user, despite the fact that the messages looked similar to generic spam. This emphasizes the importance of treating spam as a user-specific entity and the ability of our approach to address that.

We tested our approach in two modes described below.

5.1.2 Mode 1

The objective of this mode is to see if using URL categories to filter spam is viable and has a low false positive rate. We show that our approach can be effectively used on messages that contain no URLs or categorized URLs.

In this mode, only URL categories were used for classification of e-mails. 1,373 messages of 2,369 in the control set were considered (57%). Of those messages, 14 were classified as spam, the rest was classified as legitimate (0.010% false positive rate). Most of the misclassified messages contained URLs categories for which were not put in the whitelist by the user.

5.1.3 Mode 2

The objective of this mode is to illustrate that with some improvements, our approach can also be effectively used on messages that contain uncategorized URLs. Thus, we are able to consider all 2,369 messages in the legitimate corpus C_l .

As in the first mode, all messages that contained one or more categorized URLs for which a category was not in the user's whitelist were classified as spam. Some messages contained *only* uncategorized URLs. Initially, our approach classified such messages as suspicious if they contained one or more URLs that did not match any categories in the whitelist of URLs. Therefore, 33 messages were classified as spam (1.4% false positive rate) and 165 were classified as suspicious (6.9%+1.5%=8.4% total false positive rate). The main cause for the higher false positive rate was that only a small percentage of URLs are present in Yahoo and Google directories. Therefore, classifying messages containing all uncategorized URLs only using whitelists seems to generate a lot of false positives.

To deal with the problem, we devised an improvement. The idea is to categorize URLs not present in any of the directories by fetching the content of the Web sites pointed to by the URLs from Google's cache. We then run an on-line bayesian classifier on the retrieved data. The bayesian classifier we use is based on the classifier from popfile [P.04], an open-source spam filter. To further improve the accuracy, we also apply the incremental learning improvement described below.

As a result, the false positive rate of the combined approach dropped to 0.0032%, which is significantly better than that of spamassassin. We realize that the rate will vary for each user based on their individual profile but the results we obtained are very encouraging.

5.1.4 Indirect Content Retrieval

In contrast to other approaches, such as Death2Spam [R.04], we chose not to fetch the URL directly. We believe that a legitimate site would have no reason to hide its contents from the search engines.

The algorithm was very simple. For each uncategorized URL observed in training that was not present in the whitelist we trained a simple bayesian filter [P.04] on the cached data for the front page retrieved from the Google's cache. Note that it may not work well for all sites as they may use Flash graphics or be completely image-based. However, it worked quite well in our tests.

5.1.5 Incremental Learning

By incremental learning we mean that our classifier learns good URLs continuously, not just during training. This is done by adding URLs that previously occurred in legitimate messages to A_{regex} . Thus, all messages that *only* contain URLs previously seen in legitimate messages are considered legitimate.

5.1.6 Dealing with Redirects

An immediate question when it comes to URLs is redirection. Clearly, spammers could use multiple redirects to hide the real site they want a user to access. In such a case, classifying a URL that redirects to another site is of little use.

A naive way to deal with this problem would be to have a local spider retrieve each URL in question recursively and look for a meta redirection tag returned by the Web server. However, this could have serious implications for filtering systems.

	<i>spamassassin</i>	<i>Our approach</i>	
	<i>v1.63</i>	<i>Mode 1</i>	<i>Mode 2</i>
False positives rate	2%	0.010%	0.003%

Table 1: Comparison of false positive rate of spamassassin and our approach in two modes

We dismissed the idea because we believe that by doing that we would give out control to spammers again and provide them with new possibilities for experimentation, such as Denial-of-Service, feedback loops and so forth.

We chose to use search engines to deal with redirects. We found that Google spiders would follow redirects and would keep the end-site in the cache rather than the redirecting page. Thus, we are able to retrieve the redirect-free page directly from Googles cache using the *info:* query.

5.1.7 Spam Corpus

Spamassassin correctly classified all 1496 spam messages in C_s (0% false positive rate). Our system also had close to perfect accuracy of 99.998% on messages that contained URLs. There were 1313 (88%) such messages in C_s .

Note that the percentage of messages without URLs in C_s we used is lower than the average 95% [K.04]. Manual analysis of the corpus have shown that many of the emails were partially filtered or were simply meaningless email junk submissions with no contact information whatsoever (C_s is based on voluntary submissions with limited filtering.)

As described earlier, our approach does not offer information about messages that contain no URLs so if it were to be used in a standalone mode, it would be useful for around 95% of spam messages. Therefore, other filtering approaches can be used to deal with the remaining messages.

5.1.8 Remarks on False Negative Rate and Objectives

Our approach only considers messages containing URLs and therefore has higher false negative rate than spamassassin. However, the approach still achieves its goals. First, it takes URLs away from spammers as an easy-to-use feedback mechanism. To remain in business, spammers may try using other feedback mechanisms, which is likely to make spam much less effective. Alternatively, spammers may try to send users categorized URLs that users *will* be interested in. This is the other goal of our approach which will encourage the transition from spam to *user-specific* advertising.

Thus, *users* will have control over what gets into their mailbox simply by changing the list of categories in which they are interested. This is similar to Google's *Adwords(TM)* approach, whereby users only see ads related to their searches.

6 Conclusions

In this paper, we discussed the idea and an implementation to improve spam filtering by fetching the information about URLs in messages and the web sites they point to from Search Engines. We showed that the idea is viable and can be used both in a stand-alone configuration and as part of a combined spam filtering solution. As a future direction, we plan to study other types of information available through Search Engines that can be useful in spam filtering.

References

- [B.04] Einarsson B. The anomy mail tools. <http://mailtools.anomy.net/>, 2004.
- [B.T03] B.Templeton. E-stamps. <http://www.templetons.com/brad/spam/estamps.html>, 2003.
- [Cor03] Microsoft Corporation. Exchange intelligent message fi lter. <http://www.microsoft.com/exchange/techinfo/security/imfoverview.asp>, 2003.
- [dmo04] Death2spam e-mail fi ltering software. <http://death2spam.com/>, 2004.
- [DR73] Hart P.E. Duda R.O. Bayes decision theory. In *Pattern Classification and Scene Analysis*, pp.10-43, 1973.
- [E.02] Kidd E. Bayesian whitelisting: Finding the good mail among the spam. <http://www.randomhacks.net/stories/bayesian-whitelisting.html>, 2002.
- [E.03] Harris E. The next step in the spam control war: Greylisting. <http://projects.puremagic.com/greylisting/>, 2003.
- [E.K04] E.Kolve. Spamcop uri/spam uri realtime blocklist. <http://www.surbl.org/>, Apr 2004.
- [IJKC00] I.Androutsopoulos, J.Koutsias, K.V.Chandrinou, and C.D.Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam fi ltering with personal e-mail messages. In *Workshop on Machine Learning in the New Information Age*, 2000.
- [J.03] Goodman J. Spam fi ltering: From the lab to the real world. In *Spam Conference*, 2003.
- [K.04] Schneider K. Brightmail url fi ltering. In *Spam Conference*, 2004.
- [M.04] Lamb M. Tarproxy: Lessons learned and what's ahead. In *Spam Conference*, 2004.
- [P.04] Graham P. Popfi le automatic mail classifi cation. <http://popfi le.sourceforge.net/cgi-bin/wiki.pl>, 2004.
- [R.04] Jowsey R. Death2spam e-mail fi ltering software. <http://death2spam.com/>, 2004.
- [SDHH98] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to fi ltering junk e-mail. In *AAAI Workshop* pp. 55-62, 1998.
- [SM03] Mason J. et al Sergeant M. Spamassassin presentations. <http://eu.spamassassin.org/presentations.html>, 2003.
- [V.V04] V.V.Prakash. Vipuls razor documentation: Collaborative fi ltering. <http://razor.sourceforge.net/docs>, 2004.
- [Yah04] Yahoo! Yahoo directory. <http://dir.yahoo.com/>, 2004.

Appendix

6.1 Academia profile: Whitelist of categories (fragment)

Top/Science/News
Top/Science/Publications
Top/Science/Conferences
Top/Computers/Computer_Science/Organizations
Top/Computers/Computer_Science/Research_Institutes
Top/Computers/Computer_Science/Academic_Departments/North_America/United_States/Georgia
Top/Science/Technology/Academia

Top/News/Colleges_and_Universities/
 Top/Computers/Internet/Policy
 Top/Computers/Internet/Searching/Search_Engines/Google
 Top/Computers/Security/Internet/Privacy/Organizations
 Top/Computers/Security/Intrusion_Detection_Systems
 Top/Computers/Security/Virtual_Private_Networks
 Top/Computers/Supercomputing
 Top/Computers/Systems/
 Top/News/By_Subject/Information_Technology
 Top/News/By_Subject/Information_Technology/Computers
 Top/News/By_Subject/Information_Technology/Internet/Headlines_and_Snippets

6.2 Academia Profile: List of acceptable regular expressions

.edu denotes *.edu, strings beginning with a hash are ignored
 .edu
 .gov
 .org
 .mil
 citeseer.nj.nec.com
 .cos.com
 www.research.ibm.com
 www.research.att.com
 www.research.microsoft.com
 www.google.com
 www.nytimes.com
 www.cnn.com
 .yahoo.com

6.3 List of categories extracted from legitimate corpus URLs (fragment)

230 Unique URL Categories Learned from all sources. We give a fragment of the complete listing below.

Top/Business/Employment/Careers/Job_References/
 Top/Business/Information_Technology/Associations/
 Top/Business/Management/Management_Science/Management_Information_Systems/Institutes_and_Schools/
 Top/Computers/Computer_Science/Academic_Departments/North_America/United_States/California/
 Top/Computers/Hacking/Conventions/
 Top/Arts/Comics/Comic_Strips_and_Panels/B/
 Top/Arts/Online_Writing/E-zines/Non-Fiction/Society_and_Culture/
 Top/Arts/Television/Networks/PBS/
 Top/Business/Accounting/News_and_Media/
 Top/Business/Arts_and_Entertainment/Fashion/Designers/
 Top/Computers/Data_Communications/Vendors/Manufacturers/Cisco_Systems/
 Top/Computers/Ethics/
 Top/Computers/Hardware/Standards/IEEE/
 Top/Computers/Security/Intrusion_Detection_Systems/Free/
 Top/Computers/Software/Operating_Systems/Linux/Security/
 Top/Regional/Asia/India/Business_and_Economy/Shopping/Online_Malls/
 Top/Shopping/Publications/Digital/Professional_and_Technical/
 Science/Computer_Science/Supercomputing_and_Parallel_Computing/Institutes
 Social_Science/Linguistics_and_Human_Languages/Languages/Specific_Languages/English/English_as_a_Second_Language/Teaching
 [...]

6.4 Examples of categories extracted from spam messages

Top/Business/Consumer_Goods_and_Services/Beauty/Cosmetics
 Top/Business/Opportunities/Networking-MLM/C
 Top/Business/E-Commerce/Consultants/W
 Top/Business/Investing/Day_Trading/Brokerages
 Top/Business/Major_Companies/Publicly_Traded/A
 Top/Business/Major_Companies/Publicly_Traded/P
 Top/Business/Marketing_and_Advertising/Internet_Marketing/Marketing_Services/Opt-In_Email
 Top/Business/Marketing_and_Advertising/Market_Research_Suppliers/Online_Surveys/Volunteer_Focus_Groups
 Top/Computers/Hardware/Retailers/Mac
 Top/Shopping/Publications/Books/Pets/Dogs
 Top/Society/Law/Legal_Information/Business_Law/Securities
 Top/Society/Relationships/Dating/Personals/International/A
 Top/World/Chinese_Simplified/
 Top/World/Español/Computadoras/Internet/WWW/Contadores
 Top/World/Polish/Komputery/Internet/Portale