Training Datasets Collection and Evaluation of Feature Selection Methods for Web Content Filtering

Roman Suvorov, Ilya Sochenkov, and Ilya Tikhomirov

Institute for Systems Analysis of Russian Academy of Sciences
{rsuvorov,sochenkov,tih}@isa.ru

Abstract. This paper focuses on the main aspects of development of a qualitative system for dynamic content filtering. These aspects include collection of meaningful training data and the feature selection techniques. The Web changes rapidly so the classifier needs to be regularly re-trained. The problem of training data collection is treated as a special case of the focused crawling. A simple and easy-to-tune technique was proposed, implemented and tested. The proposed feature selection technique tends to minimize the feature set size without loss of accuracy and to consider interlinked nature of the Web. This is essential to make a content filtering solution fast and non-burdensome for end users, especially when content filtering is performed using a restricted hardware. Evaluation and comparison of various classifiers and techniques are provided.

Keywords: Dynamic content filtering, text classification, automatic topic identification, active content recognition, feature selection, TF-IDF, thematic importance characteristic, information gain, focused crawling.

1 Introduction

The problem of improper use of the Web has been worrying rather broad categories of people such as employers and parents since the Web came to each house. A number of various attempts to solve this problem have been proposed by the society: FOSI (former ICRA) content labeling initiative, thematic catalogs of resources, lists of URLs and regular expressions, methods for dynamic content filtering. Due to the nature of the Web only the dynamic content filtering can be considered as an adequate solution: all other approaches require hard and conscientious labor to keep databases up-to-date. The latter task is hardly solvable because of fast growth of the Web and existence of Web-anonymizers.

A good dynamic content filtering system must classify content on-the-fly with high quality. In most cases it is a compromise: the faster method is, the more often it makes errors. There are a number of commercial systems that declare use of dynamic classification: PureSight Owl ©, Blue Coat WebFilter ©, NetNanny © etc. Most systems target web resources grouped not only by theme but also by type (forums, shops etc). In this paper we will refer to such groups as categories.

G. Agre et al. (Eds.): AIMSA 2014, LNAI 8722, pp. 129–138, 2014.
© Springer International Publishing Switzerland 2014

In most cases content filters detect the following categories of content: purchase of alcohol, tobacco and drugs; web-anonymizers and proxies; chats, forums, instant messengers, dating sites and social networks; materials with cruelty and criminal information; suicide methods and stories; religious sects; news sites; file sharing sites, warez, video, image and music hostings, torrent trackers; travelling and entertainment; health and beauty; gambling and online games; popularization of various kinds of discrimination; job search websites; adult content; online shops; hobbies: sports, cars, pets etc; sites about weapons purchase and construction.

To summarize, the problem of content filtering differs from the text classification in the following aspects:

– Processing time and memory consumption is crucial (web filters perform in real time and often run on restricted hardware).
– Rates of various classification errors may vary depending on the situation (filtering may be configured more or less strict).
– The target data constantly changes (new lexis can be introduced in order to bypass filters; such resources as chats and forums don't have any specific fixed lexis).

These differences restrict the classification method that can be used: it cannot use complex techniques for feature extraction and the procedure of retraining of the classifier must be simplified.

The research presented in this paper continues the work [1]. We introduce some extensions to the original method, evaluate them in near-real conditions and compare with other classification methods. Such conditions were established using a simple thematic web crawler. The thematic web crawler is a particular case of a focused crawler that aims at collecting web pages on a certain topic [2].

The rest of the paper is organized as follows: in Chapter 2 we review available information about the dynamic content filtering and the focused crawling; in Chapter 3 we describe the modifications proposed to the original method; in Chapter 4 we discuss difficulties in collecting data for training and evaluation and ways to overcome these difficulties; Chapter 5 presents the experiment setup and the results; in Chapter 6 we sum up the work done and discuss the future research.

2 Related Work

Text categorization is a well explored area and many surveys and comparisons have been published [3, 4]. As mentioned above, the problem of the dynamic content filtering is similar to the text categorization but has several significant distinctions that have not yet been considered in the existing comparisons.

The focused crawler is a program that collects data from the Web according to some complex criteria (e.g. only scientific publications with no regard to their knowledge domain or pages on a certain topic). Rather extensive research was done in this field. According to [5] the following major approaches were developed: ontology-based, metadata abstraction, user modeling-based and other.

Most of them require resources (ontologies or models) that are expensive to generate. Babouk [2] is a keyword-based thematic web crawler that needs nearly no pre-training. It starts from a small number of keywords or URLs and crawls documents on the same topic. It uses BootCaT [6] procedure to expand the set of keywords describing the topic of interest. BootCaT itself can be used for crawling but it's not as effective as specialized crawlers are because it retrieves pages only through global search engines (search engines ban clients that try to use them intensively).

3 Method

The method discussed and improved in this paper was described in detail in [1]. According to this method a document is considered to belong to a category if its normalized Thematic Importance Characteristic (nTIC) exceeds the corresponding threshold derived during the training stage. Thematic Importance Characteristic estimates how specific the term to a particular collection of documents is comparing to another collection. To achieve this TIC relies on principles that are similar to ones that information gain relies on. nTIC of a document is estimated as a weighted sum of words TICs in it. We will refer to this method as nTIC.

Let's clarify the terms used. Category is a label that is assigned to a group of documents sharing similar features (e.g. web pages downloaded from a forum about cars have labels "forums" and "cars" assigned to them). To belong to a category is to have the corresponding label. A web site is a group of web pages that are accessible through URLs which have the same host domain name (including subdomains). Web sites and web pages can belong to multiple categories.

In this paper we propose two modifications that take into account interlinked nature of the Web. These modifications are:

- Take into account categories of the Web-sites that the currently classified page refers to.
- Tokenize URLs found on the classified page and treat these tokens as the usual lexis.

The first modification roots in the so-called thematic isolation: web pages often refer to pages from the same website or to thematically similar resources. This principle can be generalized by introducing a frequency distribution of topics of the referred resources. We propose to treat topics of resources as usual lexical features in context of the TIC-based classifier [1]. The corresponding part of the feature set is generated according to the following algorithm.

1. Extract URLs from the body of a page.
2. Extract server domain names from the retrieved URLs.
3. Map domains to categories labels using a gazetteer.
4. Calculate weights as if these labels were usual words.

The most crucial part of this algorithm is the domain-category mapping. It can be initially constructed from a catalog such as Open Directory Project [7] (e.g. an online tobacco shop would get a label "url_shopping_tobacco"). Later on it can be iteratively expanded with domains of pages that got class label with high confidence (margin between the rating of a page and the corresponding threshold is large). Each domain may correspond to multiple topics. Each topic is represented by a unique label constructed from a prefix "url_" and a title of the topic (e.g. adult, chats etc.). The goal of using the prefix is to distinguish topics of the referred pages from the usual lexis.

The second modification makes sense because human-readable URLs become more widespread (e.g. http://example.com/catalog/pages-on-some-topic instead of http://example.com/catalog.php?topicId=31415). Similar ideas were developed in [8, 9]. We propose to tokenize URLs found in the body of the page and to treat all the extracted tokens as a part of usual lexis. By tokenization we mean splitting the URL using a set of delimiters, e.g. an URL "http://mega-news/catalog/news?q=tech" is converted to the following list of tokens: url_mega, url_news, url_example, url_catalog, url_news, url_tech. Short tokens and numbers are ignored.

4 Data Sets

A good evaluation of a method for content filtering is not a trivial task because of absence of reliably marked data sets. Such standard data sets for text categorization and clustering as 20 Newsgroups and Reuters-21578 don't fit the task because they contain only textual features (no hyperlinks and markup) and sets of labels used in these corpora differ from ones that make sense for content filtering (they are less thematic and more associative). Public access lists are updated rarely and contain addresses of pages that have disappeared or have been sold to another owner. Therefore, it is useless to collect pages from such lists.

There are a couple of ways to overcome this issue:

- Use unsupervised methods of machine learning (datasets marking is not necessary).
- Use methods that require few examples to learn (in this case datasets can be marked manually).
- Introduce a technique of collecting datasets that does not require much additional manual marking.

In modern conditions such corpora cannot be easily created manually because the Web changes rapidly and it is necessary to retrain the filter periodically in order to fit it to the current state of the Web. Moreover, when training on a small dataset one cannot guarantee that recall of the lexis-based filter in real life will be the same as during the experimental evaluation.

Therefore, we have chosen the third way. We created a special web crawler to collect web pages only on the topics of interest. This crawler addresses the focused crawling problem [5, 10, 11]. However, this problem in general is very difficult. Our goal was to create a rather simple system that collects pages on the specified topic and needs no or almost no manual configuration. Babouk [2] is a thematic web crawler that is similar to the one proposed in this paper. Our system differs from it in the web resouces walk order and the decision rule.

The general idea of data collection was to reproduce behavior of the end users. To achieve this, experts tried to find pages of each category of interest using global search engines (Google, Bing etc). Then the system automatically addressed other search engines using the most productive queries. Finally, the found pages were crawled. During the crawling, the topic of the downloaded page is compared with topics of the previous pages and only the pages on similar topics are added to the dataset.

Seed URLs are addresses of pages retrieved from the global search engines. Seed pages are the pages the seed URLs refer to. Root pages are main (home) pages of web sites.

General algorithm of this crawler contains two major steps: seed URLs collection and recursive crawling.

To collect the seed URLs, we applied the following approach. An expert tried to simulate behavior of a user. To accomplish this, the expert looked for web pages on the topic of interest using the global search engines and wrote down the most productive queries. Then the system automatically sent these queries to the other global search engines and collected addresses of the found pages (seeds URLs). Duplicate URLs were then removed from the resulting list.

Before describing the crawling algorithm let us make some definitions. Let K be the number of keywords that should be extracted from the analyzed page. Let T be the list of keywords describing the subject of crawling. It consists of pairs $(word, weight)$ where *weight* is the number of pages that contained *word* since the last reduction of T. T is periodically reduced to increase recall of crawling. P is the list of keywords of the currently processed page.

The system recursively crawls the specified number of pages starting from the seed list according to the following rules:

- Web sites are traversed in the breadth-first order.
- If a seed URL points to a root page (path and query string of its URL are empty) then the system will crawl recursively the referred pages.
- If a seed URL points to a non-root page (path or query are not empty) then the system will download it but will not proceed recursively. We decided to do that to reduce amount of candidate pages.
- If the current page is a seed page then the system will extract K keywords from it and add them to the list T containing keywords that describe the topic of interest.
- If the current page is not a seed page then:
  1. Extract K keywords from the current page and put them to the list P.
  2. If $|P \cap T| < M$, where M is a minimal number of keywords set by an expert, then stop processing the current page.
  3. Update T using P (details of this step will be described below).

References found in the current page are enqueued for crawling with no regard to the topic of the page. It is done so because there is no general topic-relevant order of the web site traversal and thus we cannot guarantee that non-relevant pages do not refer to the relevant ones and vice versa. There are works on more advanced focused crawling techniques [2, 10, 11] that try to reduce amount of considered pages.

The list of keywords of a page consists of K stems of tokens that have the greatest TF-IDF rating [12]. We used Snowball algorithm [13] to extract stems. IDFs were calculated over rather large subsets of English and Russian Wikipedia. If the table of IDFs does not contain a stem, the corresponding token will be ignored. The table of IDFs was built using POS-tagging and contains only stems of verbs, nouns and adjectives. Thus, the stopwords, numbers and misspelled words are naturally filtered out from the list of keywords of a page.

If the current page belongs to the topic of interest and is not a seed page, list of topic keywords T will be updated according to the following algorithm.

1. For each keyword w in the list P of the current page do
   (a) if w in T then increment its weight by 1;
   (b) otherwise add $(w, 1)$ to T.
2. If reduction of T has not been performed for R times, reduce it. The reduction consists of two steps:
   (a) remove from T $|T| - Max_T$ entries that have the smallest weights;
   (b) decrease weights of the rest entries by the maximal weight of the removed ones.

The crawling algorithm has the following configuration parameters:

- K - the number of keywords to extract from each page;
- Max_T - maximal number of terms representing the topic of interest (maximal size of T);
- M - minimal number of keywords of a page that must be in T for this page to be added to the dataset;
- R - the number of pages to process before next reduction of T.

This algorithm allows collecting web pages on the restricted topic that is specified by a set of initial pages (seeds). Periodical updates and reductions of T give some freedom to the crawler: it can slightly diverge from initial topic. The degree of the divergence depends on the configuration parameters. We do not have a technique to estimate these parameters automatically at the moment. Furthermore, we doubt that such technique can exist because of the vicious circle: to build a classifier we already need a classifier. Fortunately, these parameters seem to be rather easy for an expert to set.

5 Evaluation

The aim of the evaluation within this work is to estimate quality of the content filtering in conditions that are rather close to the real life. To achieve this goal, we collected about 170000 web pages on 17 topics (10000 pages per topic) in English and Russian (5000 pages in each language) according to the technique described in Chapter 3. These topics are:

- Adult content - resources with pornographic and erotic videos, images and animations.
- Chats and forums - resources for chatting on non-professional topics and dating including most popular social networks.
- Criminals - resources exposing violent and related to criminal materials.
- Drugs - resources on how to purchase drugs including legal ones (spice, alcohol, tobacco) or how to make them at home.
- Entertainment - websites with advertisements and recommendations on how to spend the spare time.
- Online games - browser games, MMORPG, online shooters, racing games, online casino.
- Health - resources "for women", articles on wellness, fitness, health etc.
- Hostings - sites for sharing files, videos and images (including torrents).
- Jobs - resources with job descriptions and advertisements.
- Pets - resources on pets care.
- Proxy - Web-anonymizers, lists of proxy servers and virtual private networks.
- Sect - materials containing information on possibly dangerous religious sect stories, meetings, ceremonies etc.
- Shopping - online shops, and auctions.
- Sports - news about sports events, articles for sports lovers.
- Suicide - suicide methods, stories of self-murderers.
- Tech - news about modern technologies (IT, cars etc).
- Weapons - information on how to buy, use or create weapons at home.

The topics above are considered as "bad". The system must block documents of these topics. We also added about 20000 "good" pages to the dataset. These pages were collected from Wikipedia and other informational resources and should not be blocked.

We treat this problem as a multi-class and multi-label classification problem. It means that the system must assign zero or more labels to each document. If no labels are assigned to a document, it is considered to be "good" and is not blocked. We reduce this problem to a set of binary problems using "one-vs-all" technique.

Quality of the classification is measured using macro-averaged precision, recall and F1-measure.

The following combinations of features were used during the experiment:

- *Base* - only usual lexis.
- *Cat&Tok* - *Base* feature set extended with categories and tokens of links.

The results are present in table 1.

The most difficult categories were "Jobs", "Adult" and "Sect", probably because of their breadth and fuzzy boundaries. The easiest ones were "Sports", "Games" and "Drugs".

Table 1. Comparison of classifiers and feature extraction techniques

Classifier	Feature set	Precision			Recall			F1		
		Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
nTIC	Base	0.739	0.972	0.895	0.918	0.994	**0.968**	0.819	0.983	0.929
nTIC	Cat&Tok	0.812	0.986	**0.934**	0.909	0.988	0.963	0.878	0.986	**0.948**
SVM	Base	0.98	0.999	**0.996**	0.962	0.996	**0.988**	0.971	0.997	**0.992**
SVM	Cat&Tok	0.98	0.999	0.996	0.953	0.996	0.985	0.973	0.997	0.991

Table 2. Results of experiments on feature selection

Technique	IDF			nTIC			IG		
N	P	R	F1	P	R	F1	P	R	F1
5 000	0.977	**0.948**	**0.962**	**1**	0.852	0.921	0.99	0.83	0.908
10 000	0.983	**0.962**	**0.972**	0.981	0.955	0.968	**0.99**	0.908	0.951
100 000	0.992	**0.975**	**0.984**	0.995	0.951	0.972	**0.997**	0.958	0.977

As one can see, the extended feature set yields 7% increase in precision and 2% increase in F1. SVM performs better than nTIC on most categories, but it trains about 3 times slower and requires 2 times more memory in production. Memory consumption can become a bottleneck when working with many categories.

Often content filtering solutions work on hardware that is far from state-of-the-art (e.g. in schools, on smartphones). It means that consumption of the computational resources should be reduced as much as possible. Each category needs memory for feature selection and classifier model: e.g. if we have about 100000 features, we need about 500KB for classifier model and about 100KB to represent a document. This means that if we have 50 categories (as most modern content filters do), we need about 25MB of working memory only to classify a page (besides additional memory needed for preprocessing). Memory is a very scarce resource on most mobile devices. Furthermore, smaller feature set allows the system to work faster.

The second series of experiments addresses this issue: its goal is to determine how quality of classification depends on the feature selection technique used. During this series quality indices were evaluated over four-dimensional grid using SVM classifier with linear kernel. Dimensions were:

- Category (the same as above).
- Technique for feature selection (Inverse Document Frequency, Thematic Importance Characteristic, Information Gain) [14, 15].
- N - the number of top-rated features that must be included into the resulting feature set;
- Threshold - minimal number of documents that must use a feature in order to the feature to be significant (not considered as noise).

Results of the second series of evaluation are present in table 2. The present quality indices are macro-averaged over categories. Threshold values are also omitted for brevity (only the best values are taken into account). The numbers present in Table 2 are slightly smaller than in Table 1 because the original feature set contained about 600000 features.

6 Conclusion

In this work we evaluated and compared two classifiers, two techniques for feature extraction and three techniques for feature set reduction (feature selection) in near-real conditions. Techniques for feature extraction address the idea of using interlinked nature of the Web and the thematic isolation to improve the classification quality. We think that adding categories and tokens of URLs to the feature set does not improve the SVM accuracy because of optimization (it is already rather good). Taking into account tokens and categories of URLs may be useful to filter results of image search like Google Images. Such pages usually contain almost no text and refer to the search engine itself, but references often include addresses of the found web resources as parameters. Without the URL tokenization there are no other text-based features to classify such page.

Methods for feature selection address the need of deploying content filtering systems on any hardware (including old servers and mobile devices). Thematic Importance Characteristic gains better accuracy with small feature sets. Feature selection technique is not that important with middle-scale and large feature sets (all techniques show similar performance). The difference in quality between various feature sets originates in the way the technique resolves the tradeoff between frequent but non-special lexis and very special but rare. The more special the used lexis is the better precision is but the worse recall is.

Furthermore, a simple and easy-to-tune method for thematic web crawling is proposed and applied for collection of training data. The method is based on metasearch, keyword extraction and IDF term weighting. It is similar to Babouk [2] but due to the dynamic topic representation (i.e. reduction of the list of keywords) should provide higher control over the area of crawling. Evaluation and comparison of such web crawlers should be a subject of another research.

The main direction of the future work is the development of a heterogeneous filter. It should include a functional classifier, an analyzer of graphical content and take into account user behavior. Functional classification should improve detection of forums, shops and other types of resources that do not have very special lexis. It also probably will allow stripping out only parts of pages that contain, e.g. ”dirty” advertisements. Analysis of graphics should improve detection of violent materials and pornography resources that do not use text to describe images or videos.

Acknowledgments. The project is supported by Russian Foundation for Basic Research grant 12-07-33012. The described work is performed within the Exactus Expert [16, 17] project and the results are used in the TSA WebFilter software.

References

1. Suvorov, R., Sochenkov, I., Tikhomirov, I.: Method for pornography filtering in the WEB based on automatic classification and natural language processing. In: Železný, M., Habernal, I., Ronzhin, A. (eds.) SPECOM 2013. LNCS, vol. 8113, pp. 233–240. Springer, Heidelberg (2013)
2. de Groc, C.: Babouk: Focused web crawling for corpus compilation and automatic terminology extraction. In: 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 1, pp. 497–498 (August 2011)
3. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval 1(1-2), 69–90 (1999)
4. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34(1), 1–47 (2002)
5. Dong, H., Hussain, F.K., Chang, E.: State of the art in semantic focused crawlers. In: Gervasi, O., Taniar, D., Murgante, B., Laganà, A., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2009, Part II. LNCS, vol. 5593, pp. 910–924. Springer, Heidelberg (2009)
6. Baroni, M., Bernardini, S.: Bootcat: Bootstrapping corpora and terms from the web. In: LREC (2004)
7. AOL Inc.: Open directory project, http://www.dmoz.org
8. Baykan, E., Henzinger, M., Marian, L., Weber, I.: Purely url-based topic classification. In: Proceedings of the 18th International Conference on World Wide Web, WWW 2009, pp. 1109–1110. ACM, New York (2009)
9. Shih, L.K., Karger, D.R.: Using urls and table layout for web classification tasks. In: Proceedings of the 13th International Conference on World Wide Web, WWW 2004, pp. 193–202. ACM, New York (2004)
10. Aggarwal, C.C., Al-Garawi, F., Yu, P.S.: Intelligent crawling on the world wide web with arbitrary predicates. In: Proceedings of the 10th International Conference on World Wide Web, pp. 96–105. ACM (2001)
11. Jamali, M., Sayyadi, H., Hariri, B.B., Abolhassani, H.: A method for focused crawling using combination of link structure and content similarity. In: IEEE/WIC/ACM International Conference on Web Intelligence, WI 2006, pp. 753–756. IEEE (2006)
12. Salton, G., McGill, M.J.: Introduction to modern information retrieval. McGraw-Hill, New York (1983)
13. Porter, M.F.: Snowball: A language for stemming algorithms (2001)
14. Liu, T., Liu, S., Chen, Z., Ma, W.Y.: An evaluation on feature selection for text clustering. In: ICML, vol. 3, pp. 488–495 (2003)
15. Mitchell, T.: Machine Learning. McGraw Hill (1997)
16. Osipov, G., Smirnov, I., Tikhomirov, I., Vybornova, O.: Technologies for semantic analysis of scientific publications. In: 2012 6th IEEE International Conference on Intelligent Systems (IS), pp. 058–062 (September 2012)
17. Osipov, G., Smirnov, I., Tikhomirov, I., Shelmanov, A.: Relational-situational method for intelligent search and analysis of scientific publications. In: Proceedings of the Integrating IR Technologies for Professional Search Workshop, pp. 57–64 (2013)