# Fighting Web Spam

Marcin SYDOW [a1], Jakub PISKORSKI [b], Dawid WEISS [c], Carlos CASTILLO [d]
[a] *Web Mining Lab, Polish-Japanese Institute of Information Technology, Warsaw, Poland*
[b] *Joint Research Centre of the European Commission, Ispra, Italy*
[c] *Poznań University of Technology, Poznań, Poland*
[d] *Yahoo! Research Barcelona, Spain*

**Abstract.** High ranking of a Web site in search engines can be directly correlated to high revenues. This amplifies the phenomenon of Web spamming which can be defined as preparing or manipulating any features of Web documents or hosts to mislead search engines' ranking algorithms to gain an undeservedly high position in search results. Web spam remarkably deteriorates the information quality available on the Web and thus affects the whole Web community including search engines. The struggle between search engines and spammers is ongoing: both sides apply increasingly sophisticated techniques and counter-techniques against each other.

In this paper, we first present a general background concerning the Web spam phenomenon. We then explain why the machine learning approach is so attractive for Web spam combating. Finally, we provide results of our experiments aiming at verification of certain open questions. We investigate the quality of data provided as the Web Spam Reference Corpus, widely used by the research community as a benchmark, and propose some improvements. We also try to address the question concerning parameter tuning for cost-sensitive classifiers and we delve into the possibility of using linguistic features for distinguishing spam from non-spam.

**Keywords.** Search Engines, Web Spam, Machine Learning, Linguistic Features

## 1. Introduction

Web spamming is any form of manipulating the content, link-structure [1] or other features [2] of Web hosts and documents to mislead search engines in order to obtain undeservedly high ranking in search results. Since high ranking in search engines is positively correlated with high revenues, the motivation is almost purely economical.

Web spam combating has been regarded as the most urgent problem in the Web information dissemination process for many years [3] because it significantly deteriorates the quality of search results and thus affects the whole Web community. In addition, unhappy users can turn to competition and this translates to significant revenue cuts for search engines. Since Web spamming has such significant social and economic impact, the struggle between search engines and spammers is an "arms race": both sides apply increasingly sophisticated techniques and counter-techniques against each other.

Rapid progress of spamming techniques, an increasing number of factors to consider, and the adversarial nature of the spam phenomenon require novel techniques of

[1] The first author was supported by the Polish Ministry of Science grant: N N516 4307 33

dealing with the problem. Recently, machine learning has been successfully applied to support Web spam combating [4,5].

### *1.1. Outline of the paper*

The outline of this paper is as follows. We start by presenting the background in Section 2 which mentions the dominating role of the search engines in the Web and emphasises the role of ranking algorithms in the search process (Section 2.1). Then, we briefly describe the Web economic model (Section 2.2) which clearly explains the motivations behind the Web spamming phenomenon.

In Section 3, we describe what is usually regarded as Web spam, present a Web spam taxonomy (Subsection 3.1) and give some remarks on strategies for combating Web spam (Subsection 3.2).

Section 4 outlines the state of the art in the field of Web spam detection. We mention the reference corpus (Subsection 4.1)[2] prepared recently to help the research community in a systematic comparison of automatic Web spam detection methods and related activities (Subsection 4.2).

Next, we discuss various approaches concerning the use of machine learning with respect to Web spam detection (Subsection 4.3).

The application of the concept of *trust* is separately discussed in Subsection 4.4 due to its important role in automatic spam detection.

Section 5 introduces several open questions concerning usefulness of linguistic features in the context of Web spam classification and unbalanced training class sizes. Some of these questions stem from previous work on the subject (most notably [5]), but we also investigate an unexplored direction of using linguistic features in Web spam detection. The remaining part of Section 5 contains the description of experiments and results achieved.

We conclude and discuss possible future work in Section 6.

### *1.2. Contribution*

Applications of machine learning techniques for fighting Web spam have been in the centre of attention recently (see Section 3). Our contributions presented in this publication are listed below.

- We explore the possibility of using linguistic features contained on Web pages for detecting and classifying spam. In section 5.2 we present the attributes we computed and added to the previous attribute set. Preliminary results show that some of the features are potentially promising and they exhibit some discriminative power in automatic Web spam classification. To our best knowledge, such features have not previously been used in the context of Web spam detection (although they have been applied in other fields).
- We observed that inconsistent labelling present in the reference corpus (see Section 4.1) may lead to unnecessary deterioration of the classification quality. Our results (5.4) indicate that cleaning the data by removing non-univocally human-labelled training examples makes the resulting decision trees much simpler while

[2] A new larger corpus is currently being prepared by the research community, to be available in 2008.

the classification accuracy is not deteriorated. These results seem to be important and applicable to the preparation of future versions of the reference Web spam corpus.

- We repeated classification experiments for a wide range of cost values used in the cost classifier, trying to complete previous research done in [5]. Our results (Section 5.3) shed more light on the impact of the cost parameter on the size of decision trees, accuracy of classification and selection of significant attributes.

## 2. Background

The Web is a large source of information encompassing petabytes of publicly available data on innumerable Web pages. While it is not possible to tell exactly the size of the Web (due to the existence of dynamic documents and the impossibility of taking an instant snapshot of the Web), there are techniques for estimating the size of the "indexable" Web [6]. At the time of writing the number of indexable Web documents is estimated as 25 billion.[3]

### *2.1. The Role of Ranking in Search Engines*

To make any use of that huge amount of available information, Web users use search engines, which became the de facto main gate to the Web. Search engines themselves are huge and complex systems, answering hundreds of millions search queries over hundreds of terabytes of textual corpora daily. Discussion of main architectural and technical issues concerning large search engines can be found in [7],[8], or [9].

Processing huge amounts of data on enormous load rates is a challenge, but the most difficult problem in search technology emerges from the very fact that most users look only at the first page of search results, containing typically 10–20 results. Thus, the primary task of a search engine is to automatically sort all the results according to their relevance, authority and quality so that the best results are at the top of the matching list of thousands or millions matching candidates. This is the task of the *ranking system* module — perhaps the most secret, key component of each search engine.

Ranking algorithms, which determine the order of the returned results, use all the aspects of the information explicitly or implicitly connected with Web documents to decide the ranking position of a search result among the others. These aspects of information include (but are not limited to):

- textual contents of the body, meta-tags and URL of the document, as well as anchor text (the text snippets assigned to the links pointing to the document),
- the link structure of the whole Web graph,
- various statistics derived from query logs,
- estimates of Web traffic.

[3] http://worldwidewebsize.com

Textual components of ranking techniques are derived mostly from classic IR systems [10,11] and are based on variants of the term-frequency-inverse-document-frequency (*tfidf*) score. More sophisticated link-structure ranking algorithms were introduced to automatic search systems more recently (around 1998; e.g., [12,13]) and are still being intensively developed.

### *2.2. What drives the Web?*

Search engines are the heart of the Web [14]. One can ask about the business model which makes commercial search engines do their business. The answer is that the main source of income for search engines is *advertising*. Search-related advertising can be divided into two main categories: *sponsored links* (paid links to commercial destination pages shown alongside search results) and *contextual ads* (shown on third party Web sites). Both types of advertising rely on the search engine's technology of matching between keywords provided by the advertiser and the context: in the first case an ad is matched against the user query, in the second, against the contents (and other contexts) of the hosting Web page.

The income of search engines increases with the number of advertising customers (ad hosting Web pages share this profit proportionally). There are several different models of charging for ad's appearance: the number of impressions (cost-per-mille; CPM model) of an ad, the number of actual clicks on an ad (cost-per-click; CPC model) or the number of actual transactions made as a consequence of clicking on the ad (still the least popular model, but of increasing interest). Note that search engines try to achieve the best possible matches to make participation in ad programmes commercially attractive, but also to increase their own profit (well targeted ads are likely to be clicked on).

The total income of search-based ads in 2006 in the USA was around $6.7 billion, and constitutes 40% of the total internet-based advertising revenue.[4] Furthermore, this figure grows at a very fast rate—35% in 2006.

As the very important consequence of the Web economic model described above is that Web traffic directly turns into real profit, due to the existence of contextual advertising programs.

Bearing in mind that search engines constitute the actual "main gate" to the Web, we can make the following statements:

- ranking algorithms of search engines determine and influence the actual visibility of particular Web pages,
- the more a Web page is visible (better ranking in search queries) the more traffic goes to it (more users will eventually visit the page due to its presence among the top search results),
- more traffic on the page means more potential income (due to the contextual ad programs).

Thus, to conclude the above: *it is commercially attractive to boost ranking positions in search results*. This is the main rationale behind the existence of the Web spamming phenomenon.

[4] Internet Ad Revenue Reports, http://www.iab.net/.

## 3. Web spam

Web spam (or search engine spam) can be described as any deliberate manipulation of Web documents intended to mislead ranking algorithms of search engines in order to artificially boost the ranking position *without* actually improving the information quality for (human) Web users. Another, somehow extreme, description is: "Web spamming is everything Web authors do only because search engines exist".

The above descriptions are obviously not strict definitions — they leave much room for ambiguity which is inherent to the issue. In practice, most search engines provide their guidelines (for webmasters) to reduce the ambiguities to the minimum about what is considered spam and what is not. Note that spam is consequently punished, usually by removing the documents or hosts from indexes, thus reducing the visibility to zero.

### *3.1. Spam taxonomy*

In [2], spam techniques are classified into two broad categories: boosting techniques and hiding techniques. Boosting techniques influence the ranking used by search engines by altering the contents, links or other features of the pages or hosts. Hiding techniques serve as camouflage for other spamming techniques (e.g., hidden text or links) or provide two different views of a page to the user and to the search engine, e.g. by means of quick and automatic redirect of the user from the page indexed by the search engine to another spam page.

In general, spam techniques aim at modifying the view of documents that search engines use for indexing the pages by modifying the contents, links, etc. of the pages. Content-based techniques include copying or repetition of phrases perceived by the spammer to be relevant for their business, and in some cases, hiding such terms by using hidden text (either very small, or the same colour as the background of the page[5]), or use non-visible parts of the HTML code such as meta tags or alternate descriptions for multimedia objects which are embedded in the HTML.

Link-based techniques aim at link-based ranking algorithms such as PageRank [12] or HITS [13] by manipulating the in-links of a page. This can be done by creating a *link farm*: a tightly-knit community of pages linked to each other nepotistically [15]. The components of the link farm can be pages under the control of the spammer, pages that agree to enter a link-exchange program with the spammer, or external pages in sites that allow world-writable content. The latter is the case of wikis, forums and blogs where, if the appropriate anti-spam measures are not taken, spammers typically post links to sites that they want to boost.

Web spam affects the whole Internet community given that it deteriorates the quality of search results, and thus breaches the trust relationship between users and search engines. It also affects search engines themselves given that it forces them to waste network and storage resources indexing content that is not valuable for its users.

### *3.2. Fighting Web spam*

Defeating Web spam does not require perfection, but only to alter the economic balance for the would-be spammers [16]. A Web site owner will spam if she or he perceives that

[5] This technique used to be popular but now it is quite easy to be detected.

it is economically justified to pay more to spend a certain amount of money in spamming a search engine instead of spending the same amount of money in improving his or her Web site. While a group of spammers perhaps is able to make profit in the short term, this is not true in general and certainly not true in the long term. The first steps decreasing the amount of spam on the Web is to educate users about how to improve their Web sites to make them more attractive to users without using deceptive practises.

Search engines can also explain to users what is regarded as spam and what is not. For example, [17] advocates that search engines develop a clear set of rules and equate these rules to the "anti-doping rules" in sport competitions. Following the same analogy, search engines should increase the cost of spamming by demoting pages that are found to be using spamming techniques. Search engines also maintain spam-reporting interfaces that allow the users of the search engine to report spam results.

Numerous factors of Web documents have to be analysed to decide whether a given document is spam or not. In addition, the process of inventing new spamming techniques by spammers and subsequent updating of the ranking algorithm by search engines (in response) clearly resembles a never ending arms race.

## 4. Web spam detection

The development of an automatic Web spam detection system is an interesting problem for researchers in the data mining and information retrieval fields. It concerns massive amounts of data to be analysed, it involves a multi-dimensional attribute space with potentially hundreds or thousands of dimensions, and is of an extremely dynamic nature as novel spamming techniques emerge continuously.

### *4.1. Public corpus of spam data*

The lack of a reference collection was one of the main problems affecting research in the field of spam detection. This often obliged researchers to build their own data sets to perform experiments, with a twofold drawback. First of all, the data sets were generated to constitute a good representative of the phenomenon researchers were investigating and so, in many cases, had been biased towards it. Second and more importantly, techniques cannot be truly compared unless they are tested on the same collection.

The *webspam-uk2006* dataset described in [18] and available on-line[6] is a large, publicly available collection for Web spam research. It is based on a large crawl of Web pages downloaded in May 2006 by the Laboratory of Web Algorithmics, University of Milan.[7] The crawl was obtained from the `.UK` domain, starting from a set of hosts listed in the Open Directory Project and following links recursively in breadth-first mode.

The labelling was the result of a collaborative effort. A group of volunteers was shown a list of Web sites — the "host" part of the URLs — and asked for each host, if there were *spamming aspects* in the host. A list of typical spamming aspects were available to guide the assessment. Some of the aspects often found in spam hosts were: large sets of keywords in the URL and/or the anchor text of links, multiple sponsored

[6] http://www.yr-bcn.es/webspam
[7] http://law.dsi.unimi.it/

links and/or ad units, plus text copied from search engine results. Eventually, the corpus[8] contained 8123 hosts tagged as *normal*, 2113 hosts tagged as *spam* and 426 tagged as *undecided* (borderline).

### *4.2. The Web Spam Challenge*

Using this collection, the Web Spam Challenge series was started.[9] Two challenges were ran during 2007. The first Web Spam Challenge took place simultaneously with AIRWeb 2007[10]; six teams participated and were given a graph, the contents of a sample of 400 pages for each host, and a list of features described in [16,17]. Participants were allowed (and encouraged) to compute their own features and build classifiers that were later tested on a test set obtained from the same collection.

The second Web Spam Challenge took place during GraphLab 2007.[11] This second challenge was aimed mostly at machine learning research groups. Six teams participated (2 that also had participated in the first challenge) and were given just the graph and a set of features. Participants were not allowed to use any external source of information.

The set of features used in the first Web Spam Challenge was composed of 236 features. These features included content-based features such as average word length, number of words in the title, content diversity, term popularity and others proposed in [16]; as well as link-based features such as PageRank, number of neighbours, and others proposed in [17].

### *4.3. Web spam and machine learning*

It has been observed that the distribution of statistical properties of Web pages can be used for separating spam and non-spam pages. In fact, in a number of these distributions, outlier values are associated with Web spam [19]. Several research articles in the last years have successfully applied the machine learning approach to Web spam detection [16,20,21,4].

Building a Web spam classifier differs from building an e-mail spam classifier in a very important aspect: aside from statistical properties from the contents of the messages/pages, we also have a directed graph on the data. Furthermore, there are linking patterns that can be observed in this graph: for instance, non-spam hosts rarely link to spam hosts, even though spam hosts do link to non-spam hosts.

In the scientific literature, there are several ways in which this Web graph has been exploited for Web spam detection.

A first option is to analyse the topological relationship (e.g., distance, co-citation, etc.) between the Web pages and a set of pages for which labels are known [22,23].

A special group of Web-graph topology-based techniques, which deserves for a separate discussion, is based on a notion of *trust* which originates from the social network analysis. This topis is discussed in a subsection 4.4.

Another option is to extract link-based metrics for each node and use these as features in a standard (non-graphical) classification algorithm [17]. Finally, it has been

[8] Counts of `webspam-uk2006-set1-labels.txt` and `webspam-uk2006-set2-labels.txt` combined.

[9] http://webspam.lip6.fr/

[10] http://airweb.cse.lehigh.edu/2007/

[11] http://graphlab.lip6.fr/

shown that the link-based information can be used to refine the results of a base classifier by perturbing the predictions done by the initial classifier using propagation through the graph of hyperlinks, or a stacked classifier [5,24].

### *4.4. The Concept of Trust in Web Spam Detection*

Among the best features used in the machine-learning approach to Web spam classification are those based on the notion of *trust* or *distrust*. The concept is widely known in the social-network research community. A general survey of the trust management techniques can be found in [25].

Due to the adversarial nature of the Web, making use of the concept of trust or distrust when assessing the quality of linked Web pages proved to be a successful idea. In particular, it concerns automatic identification of the Web spam documents.

In the context of directed graphs representing virtual social networks (similar to that of the linked Web pages), a systematic approach for computing or *propagating* the trust through the edges of the graph is discussed in [26]. Various schemes for trust and distrust propagation, which are mathematically represented by the properly modified adjacency matrices and some multiplicative operations are proposed and experimentally studied with the use of some real datasets concerning virtual communities.

While in social networks the concept of trust concerns the users of the system, and models the degree of belief about the honesty of other users, in the context of the Web, the idea is slightly different. Namely, the link between two pages $p$ and $q$ is simplifically interpreted as the belief of the author of the page $p$ about the *good quality* of the page $q$. An alternative approach, however, was proposed in [27], where an extended linking language is proposed with some experiments done with the use of the `Epinions.com` dataset. The latter approach proposes to distinguish between the "appreciating" and "criticising" links between the pages by a proper extension of the markup language.

One of the first works concerning the application of the notion of trust in successful automatic identification of Web spam documents is [28]. The paper proposes an algorithm called "TrustRank" which uses a seed set of some "trusted" pages (which practically mean the pages labelled by human experts as non-spam pages) and the trust propagation algorithm derived from the classic PageRank [12] algorithm. The idea is based on the observation that non-spam pages usually link to other non-spam pages. Noteworthy, the values computed by the TrustRank algorithm (or derived from them) are found to be among the best attributes used in the machine-learning approach to Web spam classification.

The extension of the ideas discussed in [26] and [28] concerning various methods of trust and distrust propagation in the context of Web spam detection is presented in [29]. In particular, the paper proposes novel methods for splitting trust and distrust through the links as well as for aggregating the incoming values.

The "Topical TrustRank" algorithm is proposed in [30]. It overcomes two vulnerabilities of the TrustRank algorithm [28]: its bias towards more tightly-knit Web pages in the Web graph and the problem of the usual under-representation of the various categories of Web document in the human labelled, trusted seed set. The experimental results in that paper prove that introducing the topical context into the trust-computation framework significantly improves the original TrustRank's idea.

An interesting transformation of the TrustRank algorithm, which propagates the "trust" forward, through the links between "non-spam" pages is presented in [31].

Namely, the idea is similar but inverted here. The proposed algorithm, named AntiTrustRank, is based on the analogous observation: spam pages are usually *linked by* other spam pages in the Web graph. Thus, the algorithm proposes to propagate distrust backward, through links incoming to initially labeled spam pages. The experimental evaluation [31] proves that such approach outperforms that of the TrustRank algorithm.

## 5. Experiments

This section reports on our explorations of deploying linguistic features for Web spam classification using a machine learning paradigm. Further, we investigate issues concerning unbalanced training class sizes and we analyze the learned decision trees.

### *5.1. Questions and goals*

We outline the questions and goals driving the experiments presented here. Many of these questions arose as a consequence of previous research on the subject — the *webspam 2006 challenge* and [5].

1. Linguistic text features (lexical diversity, emotiveness; more details in the next section) provide very different class of information compared to graph and traditional content features. They should be good discriminators of "real", human-written content and automatically generated (or structured) gibberish. If we add linguistic features to the set of input attributes, will they help to improve the classification accuracy? What is the distribution and relationship between certain linguistic features vs. spam-normal classes?

2. A number of hosts and pages in the *webspam-uk2006* corpus are marked as "borderline" or received an inconsistent note from human evaluators. We suspect that training a classifier on this "noisy" data can mislead the learning algorithm, resulting in poorer performance and proliferation of attributes which are not truly relevant to evident spam hosts. Would initial pruning of the training data (by selecting "strong" examples of non-spam and spam hosts) improve the classification results? What will happen to the size of the resulting decision trees?

3. The two classes of Web sites (spam and normal) are highly unbalanced in size. In [5] authors use a cost-sensitive classifier to cater for this problem, suggesting that cost coefficient $R$ equal to 20 worked best in their case.[12] How sensitive is the classification depending on the actual setting of $R$? Given the same input data, is $R = 20$ really the best value to pick and why?

To address the above questions we decided to perform several new experiments using the training and test data obtained from the *webspam-uk2006* corpus. Using this particular reference data also lets us compare against the results reported in [5].

The remaining sections describe the arrangement and results of each experiment.

[12] Cost-sensitive classifiers take into account the minimum expected misclassification cost. In our case the cost coefficient $R$ is the cost given to the spam class, and the cost of the normal class is fixed to 1.

**Table 1.** Selected linguistic features used in our experiments. The "number of potential word forms" used for computing lexical validity and text-like fraction of the text refers to the number of tokens which undergo morphological analysis — tokens representing numbers, URLs, punctuation signs and non-letter symbols are not counted as potential word forms. The term "number of tokens which constitute valid word forms" refers to the number of potential word forms, which actually are valid word forms in the language, i.e., they are recognized by the morphological analyser as such word forms.

| feature name | formula | value range |
|---|---|---|
| *Lexical diversity* | $= \frac{\text{number of different tokens}}{\text{total number of tokens}}$ | $[0, 1]$ |
| *Lexical validity* | $= \frac{\text{number of tokens which constitute valid word forms}}{\text{total number of potential word forms}}$ | $[0, 1]$ |
| *Text-like fraction* | $= \frac{\text{total number of potential word forms}}{\text{total number of tokens}}$ | $[0, 1]$ |
| *Emotiveness* | $= \frac{\text{number of adjectives and adverbs}}{\text{number of nouns and verbs}}$ | $[0, \infty]$ |
| *Self referencing* | $= \frac{\text{number of 1st-person pronouns}}{\text{total number of pronouns}}$ | $[0, 1]$ |
| *Passive voice* | $= \frac{\text{number of verb phrases in passive voice}}{\text{total number of verb phrases}}$ | $[0, 1]$ |

### *5.2. Linguistic features*

There is a number of aspects that can be measured and extracted from the text apart from simple occurrence statistics. Certain language features, such as expressivity, positive affect, informality, uncertainty, non-immediacy, complexity, diversity and emotional consistency (discussed in [32]), turned out to have some discriminatory potential for human deception detection in text-based communication. Intuitively, they might also be useful in differentiating Web spam from legitimate content and, to our best knowledge, so far they have not been exploited in this context.

There are various ways of how the aforementioned features can be computed. For instance, for estimating a text's complexity, the average sentence length or the average number of clauses per sentence could be considered. In case of expressiveness, one could give a preference for certain part-of-speech categories to others (e.g., giving higher weight to adjectives and adverbs). Further, non-immediacy is indicated by usage of passive voice and generalising terms.

For our experiments, we have selected and adapted a subset of feature definitions described in [32]. In particular, we considered only features, whose computation can be done efficiently and does not involve much linguistic sophistication since the open and unrestricted nature of texts on the Web indicates that utilization of any more error-prone higher-level linguistic tools would introduce more noise. Table 1 lists the features and formula's used to calculate their value.

Two NLP tools were used to compute linguistic features: *Corleone* (Core Linguistic Entity Extraction) [33], developed at the Joint Research Centre, and Alias-i's *LingPipe*.[13] We processed only the summary version of the *webspam-uk2006* collection. It contains circa 400 pages for each host. It is important to note that solely the body of each page was taken into account. The aggregate for a host was calculated as an arithmetical average

[13] http://www.alias-i.com/lingpipe

**Table 2.** Results of classification with and without linguistic features on a full data set (all instances) and on a data set from which instances with unknown attribute values have been removed.

| | full data | | data w/o missing values | |
|---|---|---|---|---|
| | with l.f. | without l.f. | with l.f. | without l.f. |
| instances | 8 411 | 8 411 | 6 644 | 6 644 |
| attributes | 287 | 280 | 287 | 280 |
| classified ok | 91.14% | 91.39% | 90.54% | 90.44% |
| misclassified | 8.85% | 8.60% | 9.45% | 9.55% |

of values of all its pages. Interestingly, it turned out that 14.36% of the pages had no "textual" content at all and many pages simply indicated HTTP errors encountered during the crawl (404, page not found).

*Classification with linguistic features*

In our first experiment, we have tested the usability of linguistic features simply by adding them to the set of existing features in the *webspam-uk2006* collection. Surprisingly, adding these new features did not yield significantly different results (see Table 2). In case of the full data set, adding linguistic features degraded classification accuracy slightly. We were able to get a small improvement in quality by pruning the data set from instances with empty values of attributes (see Table 2), but the improvement is very little.

The first intuitive conclusion was that the new features are not good discriminators or there is some strong correspondence between them and the "original" content-based features included in *webspam-uk2006* collection (e.g., compression ratio is in a way similar to lexical diversity). We decided to take a closer look at the distribution of linguistic features with regard to the input classes.

*Distribution of linguistic features in the data set*

To get a better understanding of our previous results and the relationship between spam and linguistic features, we explored the distribution of those features in the corpus. Figure 1 depicts the distribution of lexical diversity, lexical validity, text-like fraction, emotiveness, self-referencing, and passive voice respectively.

Each diagram in Figure 1 on the following page consists of a bar graph and a line graph. The bar graph reflects the distribution of a given feature in the input corpus of Web pages. The horizontal axis represents a set of feature value ranges (bins). For example, in the first diagram on the left, the first range holds the pages, whose lexical diversity is between 0.0 and 0.05. The values on the left vertical axis correspond to the fraction of pages that fell into a particular range. The right vertical axis corresponds to the graph line, and represents the fraction of pages in each range that were classified as spam.

As can be observed, not all of the features seem to be good discriminators in spam detection. In particular, emotiveness and self referencing do not seem to be good indicators of spam, i.e., the line graph appears to be quite noisy. Certain value ranges for lexical diversity (0.65–0.80) and passive voice (0.25–0.35) might constitute a weak indication of non-spam. The spam-percentage line for lexical validity seems to have a clear

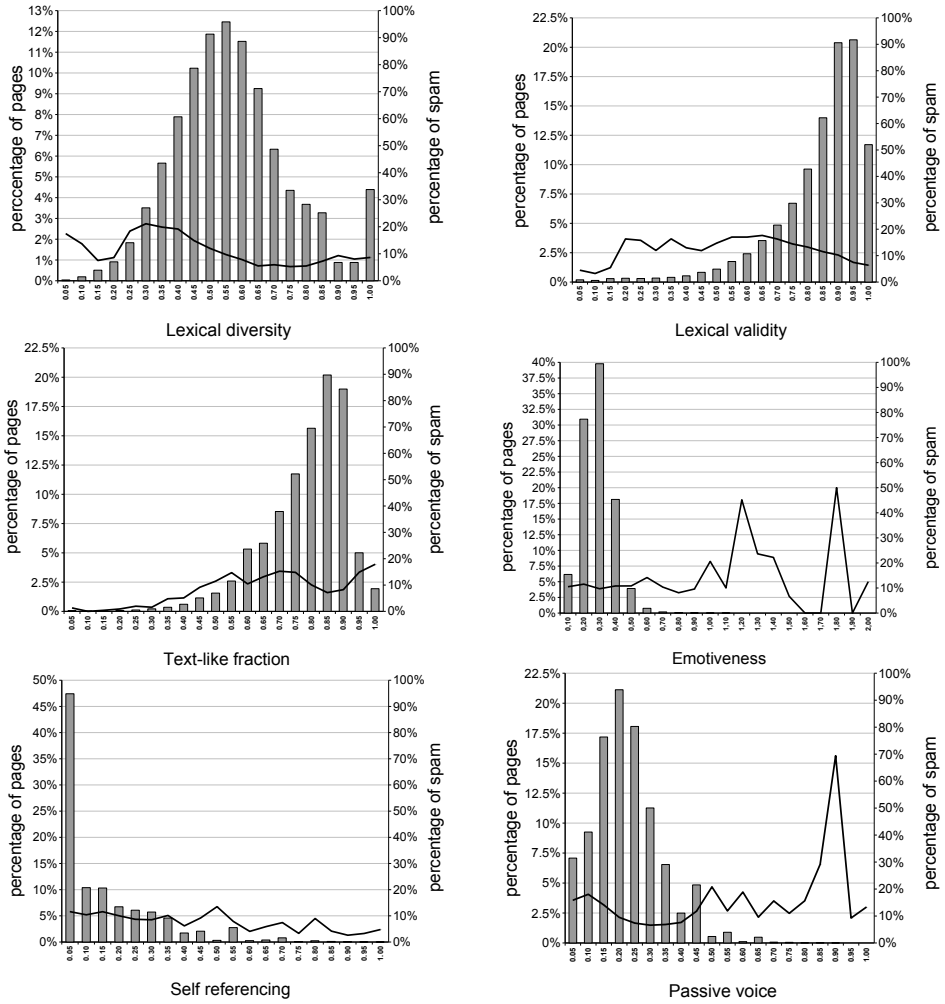



**Figure 1.** Prevalence of spam relative to linguistic features. The bar graph in each diagram reflects the distribution of a given feature in the input corpus of Web pages. The horizontal axis represents a set of feature value ranges (bins). The values on the left vertical axis correspond to the fraction of pages that fell into a particular range. The right vertical axis corresponds to the graph line, and represents the fraction of pages in each range that were classified as spam.

downward trend in the rightmost part of the corresponding diagram. In case of text-like fraction feature, values below 0.3 correlate with low spam probability.

Since many of the pages contained in the "summary" collection happen to be just short messages indicating HTTP errors, we recalculated the distributions discarding all pages with less than 100 tokens. Figure 2 depicts the recomputed distribution for text-like fraction and lexical validity. Some improvement can be observed: left and right-boundary values for lexical validity, as well as text-like fraction values lower than 0.25, correlate with higher probability of non-spam, whereas text fraction of more than 95% implies higher prevalance of spam (50%). However, the assessment of the usefulness of

each feature (the line) should take into account the number of documents in particular range (the bar).

For the sake of completeness, we also provide in figure 3 direct comparison of histograms for the latter attributes in spam and non-spam pages in the reduced corpus.

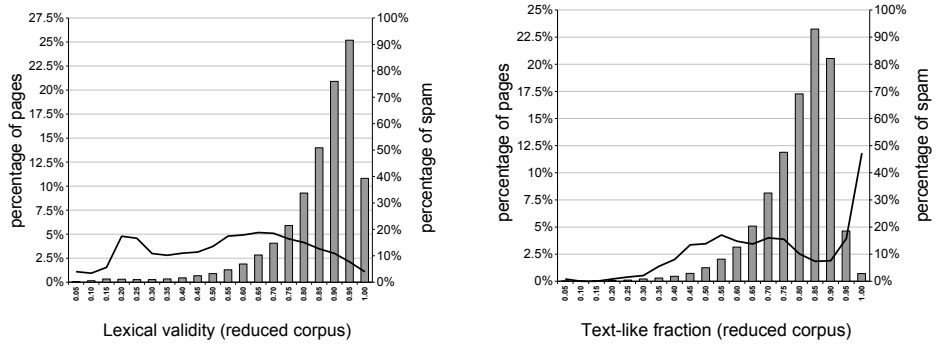



**Figure 2.** Prevalence of spam relative to lexical validity and text-like fraction of the page in the reduced input corpus.

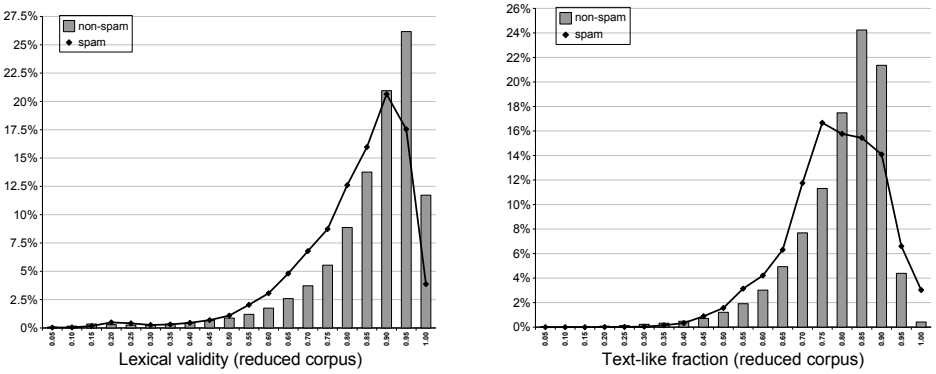



**Figure 3.** Histograms of the lexical validity and text-like fraction in spam and non-spam pages in the reduced input corpus.

The last experiment shows that a more-sophisticated way of computing some of the linguistic features might be beneficial. In general, however, the linguistic features explored in this article seem to have less discriminative power than the content-based features described in previous work on the subject [34]. This may be a result of the fact that spammers reuse some existing Web content (inject spam content inside legitimate content crawled from the Web).

### *5.3. Looking for the optimum cost value of the cost classifier*

Inspired by the results reported in [5], we wanted to shed some more light on the characteristic of the cost ratio ($R$) between spam and normal classes, given to the cost-sensitive classifier (see footnote 3 on page 9).

**Table 3.** The number of hosts that received an identical number of votes for the "pure" and "purest" sets. The denotations NN, NNN, NNNN mean that a host received a univocal "normal" label from (all) 2,3 or 4 human assessors (respectively). Similarly, the SS denotation concerns the host that obtained "spam" label from both the human assessors.

| data set | votes | | | |
|---|---|---|---|---|
| | NNNN | NNN | NN | SS |
| pure | 13 | 843 | 1850 | 323 |
| purest | 13 | 843 | — | 323 |

We trained and tested the cost-sensitive classifier (with underlying J48 algorithm) for $R$ values ranging between 1 and 200. For each value of $R$, a single training/ testing round was executed until the resulting decision tree does not depend on the order of input data or other parameters.

As it turned out, adjusting the cost of misclassifying a spam page as normal ($R$) does not affect the f-measure as much as one could think (see Figure 4 on page 16). Increasing $R$ beyond 70 does not change the results significantly at all. True positive/ false positive ratio curves are more insightful compared to f-measure — it seems sensible to strike the balance between TP (true positive) and FP (false positive) ratios of spam and normal classes and this happens for value of $R$ somewhere around 20, just as previously reported in [5].

### *5.4. Classification accuracy for classifiers trained on "clean" data*

In this experiment we start from the assumption that the label assigned to training examples (spam/ normal) is not always correct. This was motivated by the analysis of the training data — even though the labels were assigned by humans, there were frequent cases of inconsistent labels between judges, most likely caused by pages or hosts that mixed legitimate content with spam [18]. Instead of training the classifier on this "borderline" data, we decided to extract just the strongest examples of spam and normal hosts and use this subset for learning.

We processed the "judgement" files from the *webspam-uk2006* collection, splitting hosts into subsets that exhibited full agreement of judges. For each host we concatenated all votes it received so, for example, a host marked with `SS` received two "spam" votes, a `NNN` host received three "normal" votes and so on. We then created two sets — "pure" and "purest", consisting of hosts with the following labels:

- `NNNN`, `NNN`, `NN`, `SS` hosts ("pure" set),
- `NNNN`, `NNN`, `SS` hosts ("purest" set).

The number of the hosts in each group is given in Table 3.

Finally, we trained a cost-sensitive classifier on each of these filtered sets, for changing cost value $R$ — this time between 1 and 40. The resulting decision trees were evaluated against the original set of hosts (including those that received mixed votes) to keep the results consistent and comparable with previous experiments.

Figure 5 illustrates the F-measure, area under curve (AUC), true positive (TP) and false positive (FP) ratios for three training data sets — pure, purest and the original unfiltered set. Before we compare the results note that, regardless of the data set, the "opti-

mal" value of the cost $R$ seems to be around the value of 20 — this is where TP/FP meet and the F-measure/ AUC reach their peak values. As for pruning the training data, we can observe a slight boost of quality (F-measure, AUC) for the "pure" set. However, further pruning ("purest" input) does not yield any improvement, even degrades the performance of the final decision tree (note high values in the sub-figure showing true positives).

Summing up, removing the noisy borderline elements from the training data contributes slightly to the accuracy of the final classifier, although leaving out just the strongest examples results in borderline cases to be classified as spam. Not depicted in Figure 5, but of interest and relevance, is the size of the final decision tree, discussed in the next section.

### *5.5. Analysis of the output decision trees*

Figure 6 shows the size of the tree and the number of attributes used for 3 different inputs. We may see that the cleaner the input data, the fewer attributes are needed to properly distinguish between spam and non-spam. Taking into account the fact that there was little difference in quality between the decision tree trained on all instances compared to the "pruned" set, this may mean redundancy of some attributes in the original tree.

We performed the following analysis. For each data set (unfiltered, pure, purest) and for each value of $R$ between 1 and 40, we counted the attributes occurring in the final decision tree (conditions on branches). We then calculated which attributes were used most frequently to decide between a spam host and a regular host. Among the most influential attributes[14], regardless of the value of $R$, were:

- logarithm of the number of different supporters (different sites) at distance 4 from the site's home page,
- logarithm of the trust rank of a given host's home page,
- length of host name,
- top 100 corpus recall, fraction of popular terms that appeared on the page (*STD_83*),
- top 100 corpus precision, fraction of words in a page that appeared in the set of popular terms (*STD_79*),
- compound features such as *log_OP_trustrank_hp_div_indegree_hp_CP* (various coefficients such as trust rank, in degree etc., combined into a single formula).

Actual conditions on these attributes were quite sensible; for example, if length of the host name exceeds 20 characters or the page contains many popular terms (*STD_83*), then it is most likely a spam host.

Note that these attributes were not only the most frequently used for choosing between spam and normal hosts, but were also stable with respect to the change of cost parameter $R$ (as visually depicted in Figure 7).

[14] See [5] for details concerning how these attributes were computed.

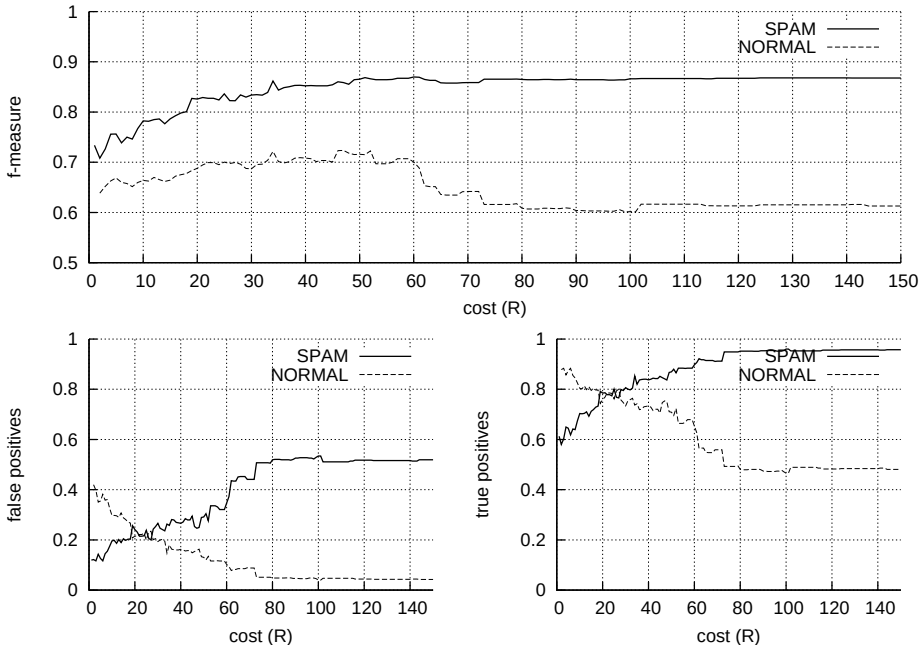



**Figure 4.** F-measure and TP/FP rates for changing misclassification cost $R$ (continuous lines for clarity).

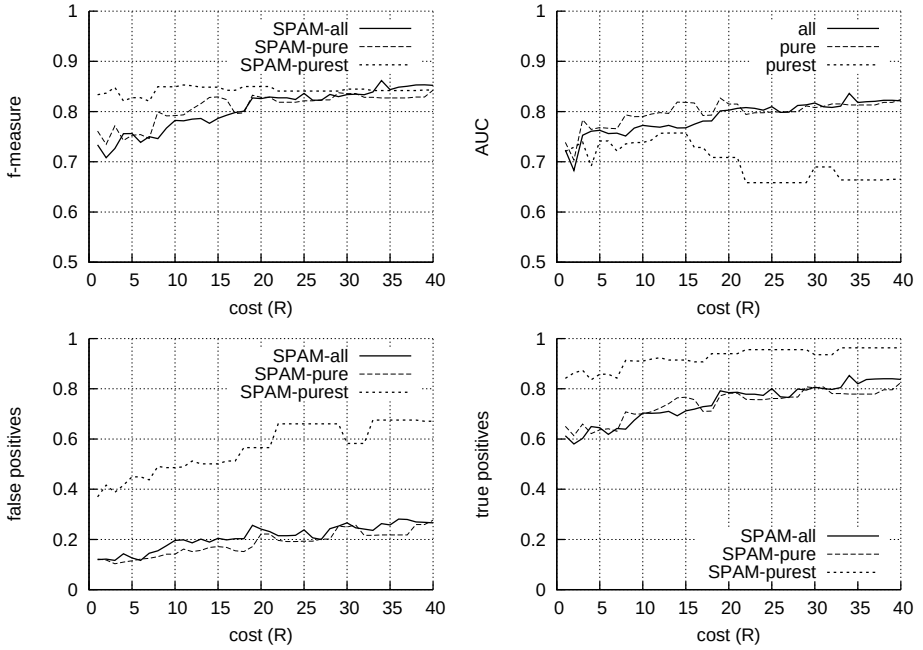
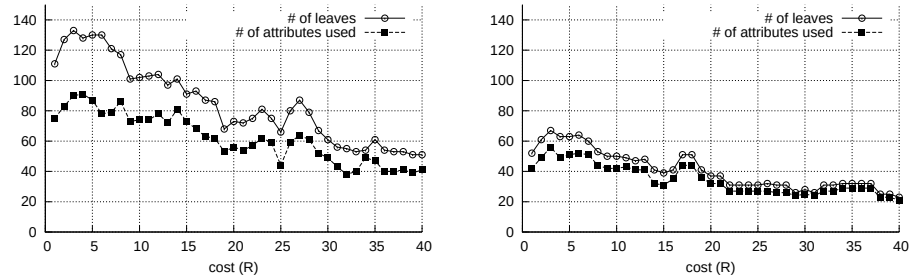



**Figure 5.** F-measure, TP/FP rates and AUC for changing misclassification cost $R$, results shown only for the spam class, all three inputs on one chart: pure, purest and unfiltered training set.




a) unfiltered training data b) "pure" set of hosts

**Figure 6.** Size of the decision tree (number of leaf nodes and attributes used) depending on the training set.

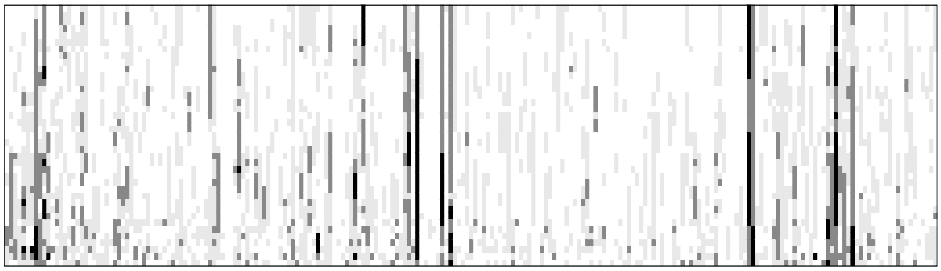


**Figure 7.** Visualisation of attributes used inside decision trees. Each "column" in the horizontal axis represents a single attribute, vertical axis reflects changing values of $R$ — 1 on the bottom, 40 on the top. A grey square at each intersection indicates the number of times the attribute was used, aggregating over the three training sets; light grey: 1 time, grey: 2 times, black: 3 times. Note nearly solid black vertical lines — these attributes were almost always used for prediction.

## 6. Summary and conclusions

Web spam is any form of manipulation of Web documents intended to mislead ranking algorithms of search engines in order to artificially boost the ranking position without improving the information quality for Web users. Fighting Web spam is being considered as one of the most urgent problems in the Web information dissemination process since it significantly deteriorates the quality of search results and thus affects the whole Web community. In this article, we gave a short overview of the Web spam phenomenon and state-of-the-art techniques for combating it. Further, we tried to answer and verify several open questions by applying machine learning techniques.

First, we explored whether linguistic features, which go beyond classical content-based features (used by others), have any discriminatory power for classifying spam. In particular, we experimented with features like lexical validity, lexical diversity, emotiveness, text-like fraction, passive voice and self reference, which proved to be useful in the process of detecting human deception in text-based communication [32]. Various experiments on including these features for training a classifier did not show any significant improvement in the accuracy, although some of the corresponding distribution graphs revealed some discriminatory potential.

Our second endeavour focused on experimenting with training the classifier on "cleaned" data, i.e., data pruned via removing the "borderline", which were neither clas-

sified as spam nor legitimate pages, and non-univocally labelled instances in the training corpus. Removing such noisy data yielded significantly simpler decision trees without deteriorating the classification accuracy. Presumably some attributes in the trees computed from the original data were redundant. We also observed that there were some attributes which were most influential disregarding the cost coefficient and training dataset used. These included: logarithm of the trust rank of hosts home page, length of host name, logarithm of the number of different supporters, top-100 corpus recall, top-100 corpus precision and some compound features like trustrank combined with indegree.

A continuation of the application of light-weight linguistic analysis in machine learning approach to Web spam detection is envisaged. Most likely, the linguistic features studied in our work duplicate information of the traditional content-based features. In the next step, we intend to train the classifier solely using linguistic features in order to verify the latter assumption. Further, we also intend to explore more sophisticated features like for instance positive affect, syntactical diversity, etc.

The current and future results of our work related to the application of linguistic features for web spam detection will be available at the following URL:
`http://www.pjwstk.edu.pl/~msyd/lingSpamFeatures.html`

## 7. Acknowledgements

The work was additionally supported by the EMM project carried out at the Joint Research Centre of the European Commission (`http://emm.jrc.it/overview.html`) and by the internal grant of Polish-Japanese Institute of Information Technology: ST/SI/06/2007.

## References

[1] Zoltán Gyöngyi and Hector Garcia-Molina. Link spam alliances. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, pages 517–528, Trondheim, Norway, 2005.

[2] Zoltán Gyöngyi and Hector Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*, pages 39–47, Chiba, Japan, 2005.

[3] Monika R. Henzinger, Rajeev Motwani, and Craig Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2):11–22, 2002.

[4] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi. Detecting spam blogs: A machine learning approach. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Boston, MA, USA, July 2006.

[5] Carlos Castillo, Debora Donato, Aristides Gionis, Vanessa Murdock, and Fabrizio Silvestri. Know your neighbors: Web spam detection using the web topology. In *Proceedings of SIGIR*, Amsterdam, Netherlands, July 2007. ACM.

[6] Ziv Bar-Yossef and Maxim Gurevich. Efficient search engine measurements. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 401–410, New York, NY, USA, 2007. ACM.

[7] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, and Sriram Raghavan. Searching the web. *ACM Transactions on Internet Technology (TOIT)*, 1(1):2–43, 2001.

[8] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[9] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan-Kauffman, 2002.

[10] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[11] Gerard Salton and Michael McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1984.

[12] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[13] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[14] Ian H. Witten, Marco Gori, and Teresa Numerico. *Web Dragons: Inside the Myths of Search Engine Technology (The Morgan Kaufmann Series in Multimedia and Information Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[15] Brian D. Davison. Recognizing nepotistic links on the Web. In *Artificial Intelligence for Web Search*, pages 23–28, Austin, Texas, USA, July 2000. AAAI Press.

[16] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the World Wide Web conference*, pages 83–92, Edinburgh, Scotland, May 2006.

[17] Luca Becchetti, Carlos Castillo, Debora Donato, Stefano Leonardi, and Ricardo Baeza-Yates. Link-based characterization and detection of Web Spam. In *Second International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Seattle, USA, August 2006.

[18] Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini, and Sebastiano Vigna. A reference collection for web spam. *SIGIR Forum*, 40(2):11–24, December 2006.

[19] Dennis Fetterly, Mark Manasse, and Marc Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings of the seventh workshop on the Web and databases (WebDB)*, pages 1–6, Paris, France, June 2004.

[20] Tanguy Urvoy, Thomas Lavergne, and P. Filoche. Tracking web spam with hidden style similarity. In *Second International Workshop on Adversarial Information Retrieval on the Web*, Seattle, Washington, USA, August 2006.

[21] Gilad Mishne, David Carmel, and Ronny Lempel. Blocking blog spam with language model disagreement. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Chiba, Japan, May 2005.

[22] András Benczúr, Károly Csalogány, and Tamás Sarlós. Link-based similarity search to fight web spam. In *Adversarial Information Retrieval on the Web (AIRWEB)*, Seattle, Washington, USA, 2006.

[23] Dengyong Zhou, Christopher J. C. Burges, and Tao Tao. Transductive link spam detection. In *AIRWeb '07: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 21–28, New York, NY, USA, 2007. ACM Press.

[24] Qingqing Gan and Torsten Suel. Improving web spam classifiers using link structure. In *AIRWeb '07: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 17–20, New York, NY, USA, 2007. ACM.

[25] Sini Ruohomaa and Lea Kutvonen. Trust management survey. In *Proceedings of the iTrust 3rd International Conference on Trust Management, 23–26, May, 2005, Rocquencourt, France*, pages 77–92. Springer-Verlag, LNCS 3477/2005, May 2005.

[26] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 403–412, New York, NY, USA, 2004. ACM.

[27] Paolo Massa and Conor Hayes. Page-rerank: Using trusted links to re-rank authority. In *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 614–617, Washington, DC, USA, 2005. IEEE Computer Society.

[28] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating Web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, pages 576–587, Toronto, Canada, August 2004. Morgan Kaufmann.

[29] Baoning Wu, Vinay Goel, and Brian D. Davison. Propagating trust and distrust to demote web spam. In *Workshop on Models of Trust for the Web*, Edinburgh, Scotland, May 2006.

[30] Baoning Wu, Vinay Goel, and Brian D. Davison. Topical trustrank: using topicality to combat web spam. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 63–72, New York, NY, USA, 2006. ACM.

[31] V. Krishnan and R. Raj. Web spam detection with anti-trust rank. In *ACM SIGIR workshop on Adversarial Information Retrieval on the Web*, 2006.

[32] A. Zhou, J. Burgoon, J. Nunamaker, and D. Twitchell. Automating Linguistics-Based Cues for Detect-

ing Deception of Text-based Asynchronous Computer-Mediated Communication. *Group Decision and Negotiations*, 12:81–106, 2004.

[33] Jakub Piskorski. Corleone - Core Language Entity Extraction. Technical Report (in progress). Joint Research Center of the European Commission, 2008.

[34] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of WWW 2006, Edinburgh, Scotland*, pages 83–92, 2006.