Content-based Web Spam Detection

Gordon V. Cormack
University of Waterloo
gvcormac@uwaterloo.ca

Our 2007 Web Spam Challenge submission used an ensemble of ten content-based classifiers stacked using logistic regression. Each classifier used one of two state-of-the art email filters – DMC [2] or OSBF-Lua [1]– applied to simple text files, with each text file acting as a proxy for a host to be classified. All text files were derived from the home page (including http and redirection logs), the host name, or the host names associated with incoming or outgoing links. Except for the host names of these immediate neighbours, no information about the topology of the corpus was used.

The following ten methods were used to create and classify proxy text files. DMC was applied in nine cases; OSBF-Lua in only one.

1. *homebig.* The http response and text for the host home page was retrieved from the summary corpus. If the text size did not exceed 5000 bytes, the next page (in corpus order) was substituted, and so on, until either a page exceeding 5000 bytes or the largest page on the host was found. Note that corpus order corresponds to a breadth-first traversal of the outgoing links. DMC was applied to the first 2500 bytes of this file.

2. *homebig.tail.* The same file was used as for *homebig*, but DMC was applied to the last 2500 bytes of the file.

3. *httponly.* The http response for the host home page.

4. *bodyonly.* The text for the host home page.

5. *wget.* The live web version of the host home page, including http log and redirection, as fetched by *wget.* DMC was applied to the first 2500 bytes of this file.

6. *wget.tail.* The same file as *wget*, but DMC was applied to the last 2500 bytes of the file.

7. *wget.osbf.* The same file as *wget*, but OSBF-Lua was applied to the entire file.

8. *hostname.* A file containing only the host name, as given in the corpus.

9. *ingraph.* A file containing a list of the host names corresponding to incoming links, separated by spaces.

10. *outgraph.* A file containing a list of the host names corresponding to outgoing links, separated spaces.

The particular methods and the stacking method were derived using 10-fold cross validation on the labeled data. Scores were normalized and combined using log-odds and logistic regression as described by Lynam and Cormack [3].

Method	*AUC*	F_1	*weight*
homebig	.939	.634	.064
homebig.tail	.938	.626	.056
httponly	.867	.481	.124
bodyonly	.933	.627	.184
wget	.942	.622	.121
wget.tail	.942	.619	.135
wget.osbf	.929	.635	.200
hostname	.864	.424	.095
ingraph	.952	.639	.383
outgraph	.834	.289	.021
log-odds	.975	.796	-
logistic	.980	.803	-

Table 1: Cross-validation Results

Table 1 shows the results of applying these methods individually, and stacking them using equal (log-odds) weights as well as weights derived by logistic regression.

1. REFERENCES

[1] Assis, F. OSBF-Lua – a classification module for Lua: The importance of the training method. In *Proc. 15th Text REtrieval Conference (TREC 2006)* (Gaithersburg, MD, November 2006).

[2] Bratko, A., Cormack, G. V., Filipic, B., Lynam, T. R., and Zupan, B. Spam filtering using statistical data compression. *Journal of Machine Learning Research 7* (2006), 2673–2698.

[3] Lynam, T. R., and Cormack, G. V. On-line spam filter fusion. In *29th ACM SIGIR Conference on Research and Development on Information Retrieval* (Seattle, 2006).

AIRWeb 2007, May, 2007, Banff.