Beau Crawford
CS 5761 Final Project
May 7, 2004

Gmail: A New Breed of Email

Background

Gmail is a newly offered web-based email system from Google. It can be thought of as an extension to its search engine service, which is commonly viewed as the most powerful search option available today. Gmail will operate in a manner similar to that of other web-based email services like Hotmail and Yahoo. Competing services like these typically offer users only a few megabytes of storage capacity. In complete contrast, Gmail encourages users to not "throw anything away" by offering them one gigabyte (1024 megabytes) of email. In having that amount of information stored, Google believes that users will require a non-traditional way of browsing their emails or, as they put it, a "search, don't sort" approach. Instead of relying on multiple sub-directories to house archived email, Google will offer users the capabilities of its search engine technology to sift through their archived email.

Problem Description

As users use Google to explore their archived email, in a manner similar to web searching, the Gmail system will be analyzing both their search criteria and the content of the emails being searched. The returned list of results will then allow Google to generate and display relevant advertisements in a non-invasive manner. For example, searching for an email about an upcoming vacation with a friend is likely to cause ads for hotels and rental cars to be displayed. A similar sequence of events will take place as users read new email and browse old emails on their own. The difference here however will be that the email content will be "blindly" analyzed by the Gmail system – meaning that it doesn't necessarily know what the user is interested in. The Gmail system will attempt to determine the key subjects of the email and then display relevant ads to the user. The focus of this project will be on the "blind" analysis of emails. My system will have the capability to analyze the body of an email and programmatically find its main topics. From there, it will attempt to link them to pre-built advertisements that directly relate.

Henzinger, Chang, Milch, and Brin discuss a system that associates television content with web articles the listener might be interested in [1]. A variety of algorithms, including things like document frequency, are discussed in this article. My system deals with a similar concept except that, instead of TV content, it will be analyzing email and, instead of web articles, it will be displaying ads. My system will not directly implement any of these algorithms. Instead, I will be relying on their system as a conceptual foundation for my own ideas.

Practical Applications

Practical applications for such a system are somewhat limited. This is due to the fact that a system like Gmail requires you to be granted permission to electronically analyze private information. The system itself is basically identical to a Web-based advertisement linking

system that analyzes page content instead of email bodies. The core functionality of my system could be used in a similar manner.

Gmail is a radical attempt to shift the web-based advertising paradigm. On most websites today, "target marketing" is occurring only in an extremely limited form – with ads placed using only a general topic. Gmail is the first attempt to advertise directly to users based on information harvested from personal emails. While this is obviously somewhat controversial, the potential benefits are abundant. Advertisers will undoubtedly jump at the opportunity to reach consumers in a new way.

General Algorithms

The first requirement for this application is that it be able to dynamically associate common words. For example, the word **swim** must be associated with words like **pool**, **lake**, **snorkel**, **dive**, **water**, etc. To accomplish this, the following algorithm is proposed:

Keyword Building

1. Manually define a small set of generalized keywords (or phrases) for each advertisement.
2. Using the Google API, look web pages that contain each word. Utilize the top n page links returned from a query for each keyword specified above.
3. Store every word found on each page in a cumulative hash table, along with its frequency. This frequency will be referred to as the "Web Frequency".
4. Ignore the 100 most common English words.

The generated list will hold Zipifian characteristics. That is, the majority of the words in the list will have very low frequencies and therefore be of little value. To eliminate this, the following step will be included in the algorithm:

5. Remove words with a Web Frequency less than 5

Here is an example of what the complete Keyword Building algorithm might produce. The dynamically generated index contains Web Frequencies and their associated keywords obtained from the Google API:

Keyword Bootstrap	Dynamically Generated Index
1. Pool **2. Swim** **3. Whirlpool**	**1039 Pool** 902 Water **491 Whirlpool** 401 Dive 390 Trunks **110 Swim** 63 Suit 52 Sauna … 5 Snorkel 5 Fish 5 Boat

By utilizing this approach, there would be no need to manually define lengthy word lists for each advertisement. This functionality represents the true muscle of the system. By

harnessing the power of Google to do the grunt work, the user's job is made vastly easier. An added benefit is that, after querying the Google API, the keyword indices will contain words commonly associated with the Keyword Bootstrap. Since information will be obtained directly from Web content – where the writing style is similar to that of everyday email – the results will be extremely valuable in the linking of ads to emails.

The second requirement for the system is that it displays ads based directly on the content of an email. To do this, the system will utilize the pre-built keyword indices for each ad. The keyword indices will each be compared against the email's body and be used to score the parent advertisements. The algorithm to do this is rather straightforward and is described as follows:

Ad scoring

1. Count the frequency of each word in the email body
2. Multiply the word's frequency in the email by its "Web Frequency", which was obtained in the Keyword Building algorithm
3. Add the product from step three to a running sum for each ad
4. Display the top n ads (optionally specified) by using their overall totals obtained from steps and four.

Example: If the word "Sauna" occurred 4 times in an email and had a Web Frequency of 52, then 208 would be added to the cumulative score for the current ad being considered. In contrast, the word "Boat" might occur more frequently in the email, say eight times, but it would receive a lesser score (40) since it only had a Web Frequency of five for the same ad.

Important notes:

- Specific words mentioned in the email would not necessarily have to be listed in the keyword index in order for the email itself to be associated with a particular ad. Instead, all of the words in the email will be considered and scored as a set. By utilizing this approach, the topic of the email itself can be more easily linked to ads in the system. Ads will not be automatically thrown out should they be missing one or two important words.

- Although no "fuzzy matching" algorithm will be utilized, spelling errors will still be considered. In fact, this will happen automatically. Since the Web Frequency of words in the keyword indices will be coming from actual Web content, common spelling errors will be included. This means that if a word is misspelled in the body of an email, it will need to be misspelled in the keyword index as well in order for a match to occur and for that word to be included in the score. This requirement should increase the overall accuracy of the scoring algorithm, since certain spelling errors often occurs consistently – regardless of the author or corpora.

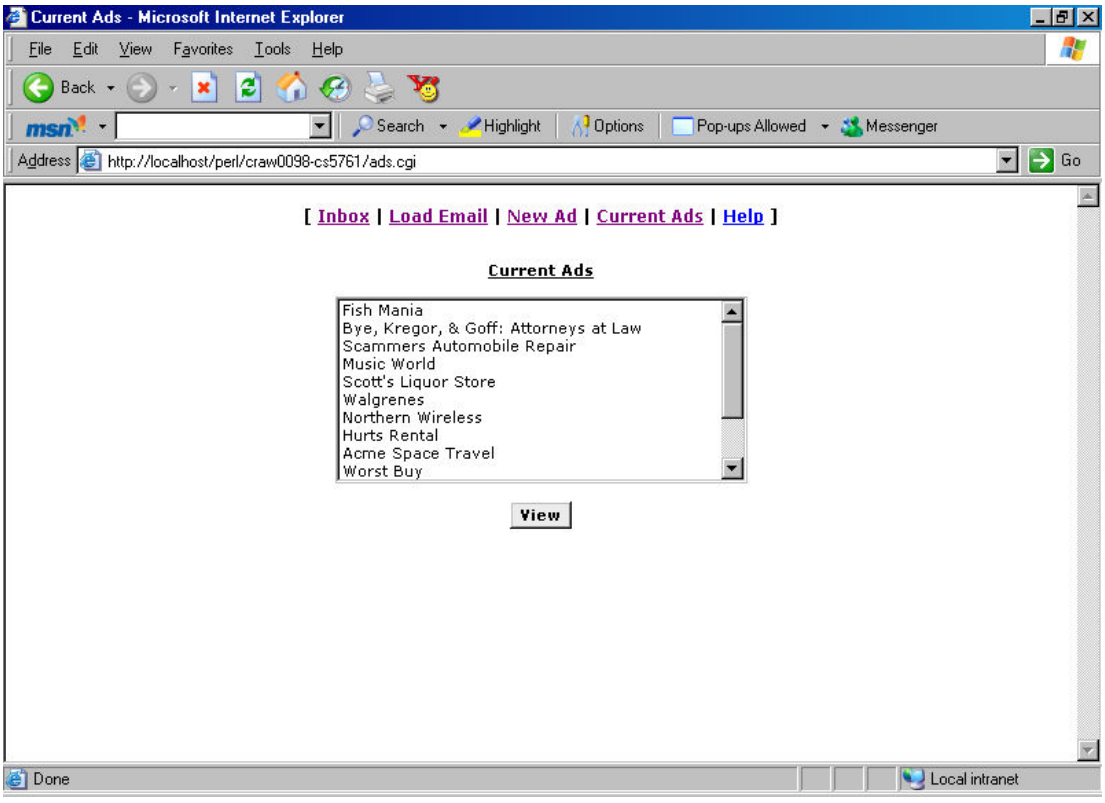
Possible improvements:

- Iwadera and Kimoto have developed a Associated Information Retrieval System (AIRS) [2] that is based on a "dynamic thesaurus". The dynamic thesaurus consists of nodes, which represent each term of a thesaurus, and links, which represent the connections between nodes. Term information that is automatically extracted from

user's relevant documents is used to change node weights and generate links. Similar functionality could be used in my system to extend the word associating. If implemented, the public word database WordNet would be utilized. Such functionality would allow for broader and, possibly, more accurate keyword indices to be built. For example, if the word "chocolate" was part of a keyword index then the system might look for words commonly associated with chocolate and include them in the keyword index for that ad.

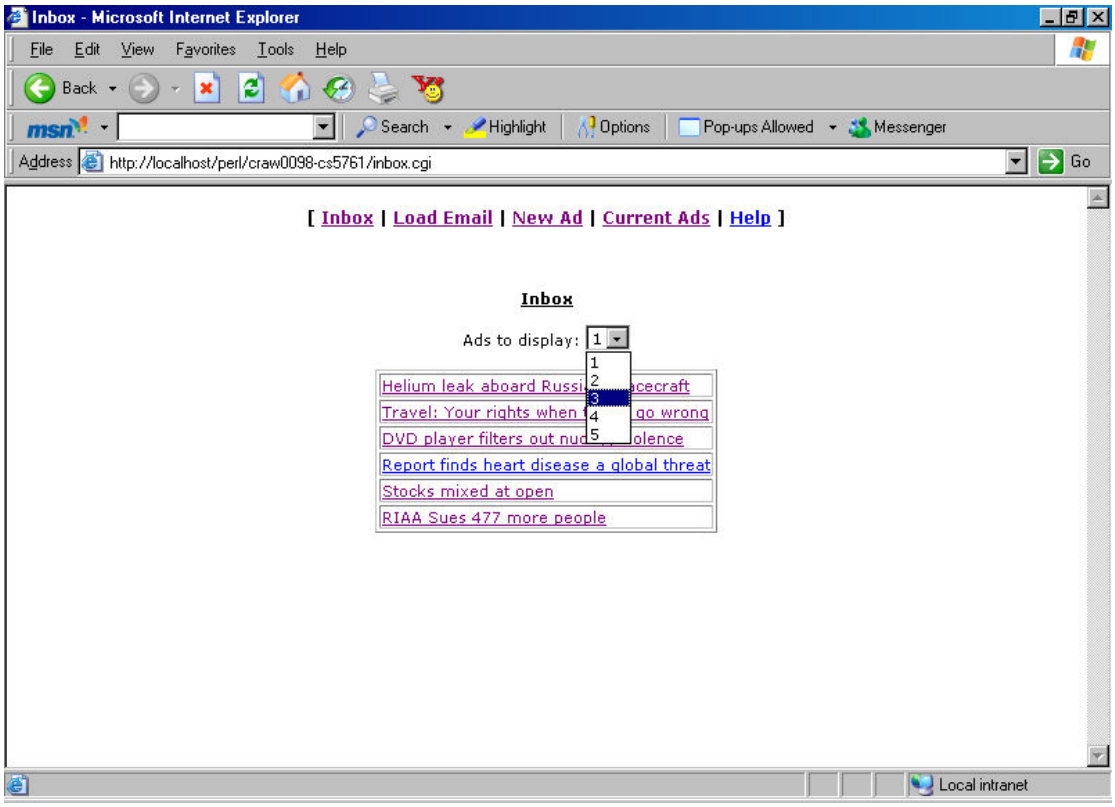
- An alternative scoring algorithm could be utilized. A viable option is to implement a Naïve Bayesian Classifier, similar to the one used in Pantel and Lin's SpamCop program [3]. However, instead of attempting to locate spam, the algorithm could be modified to determine the probability that a particular ad file is associated with content from an email. The benefit of this approach would be that the system would assign a probability to each ad. This probability would represent the likelihood that an ad is related to the content of the email. Utilizing this approach would make it far easier to gauge the overall success of the system.

System Overview



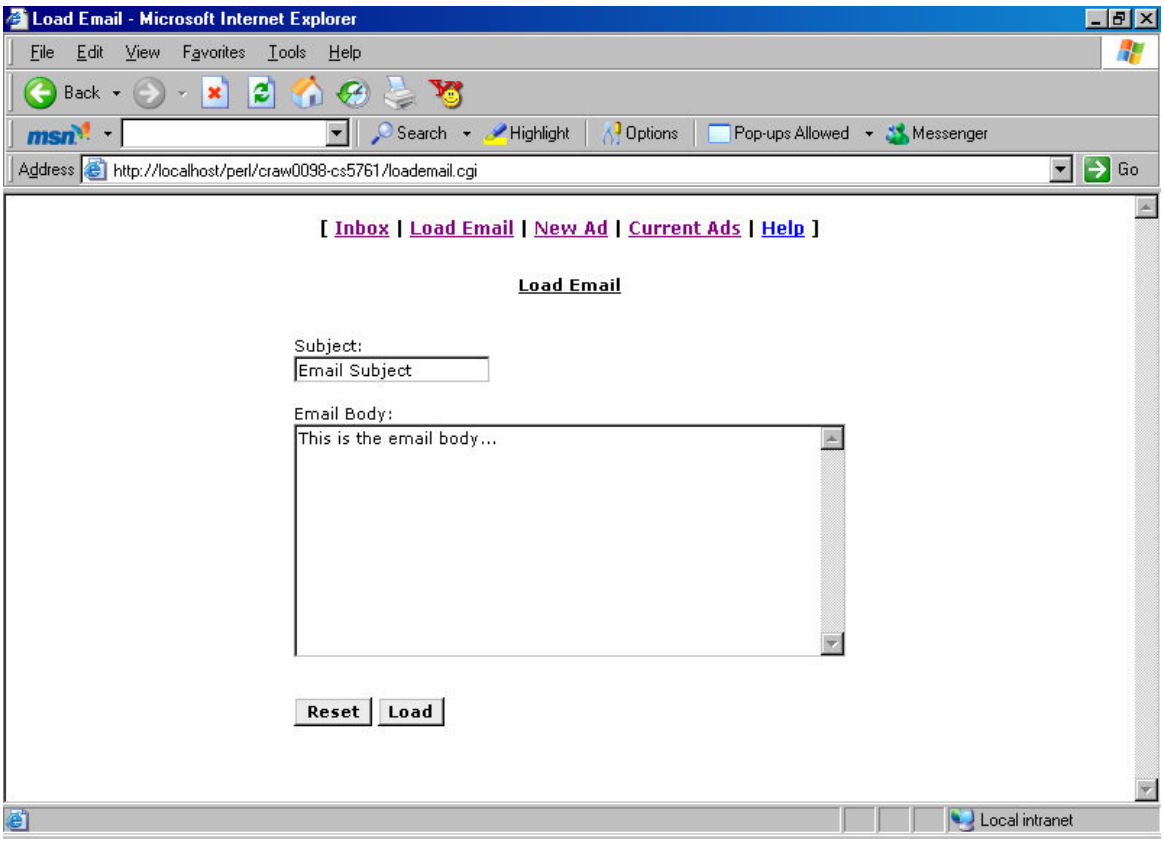
Current Ads interface

This interface lists all of the ads currently loaded into the system. To view the ad driver file for a particular ad, simple select it and then click the "View" button.



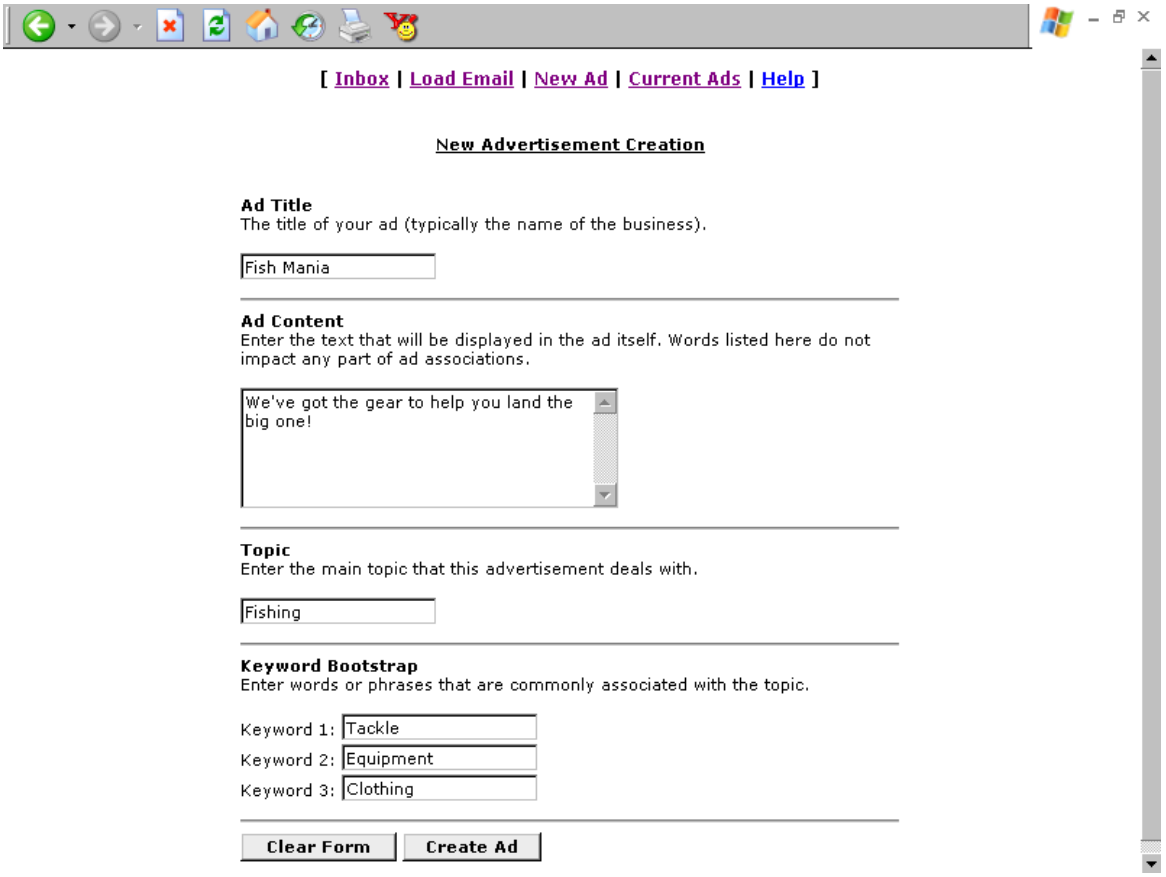
Email Inbox

This interface lists all of the simulated emails that currently reside in the inbox. To view a particular email, simply click its subject. Optionally, testers of the system can specify the number of ads they would like generated for this email by clicking the "Ads to display" dropdown.



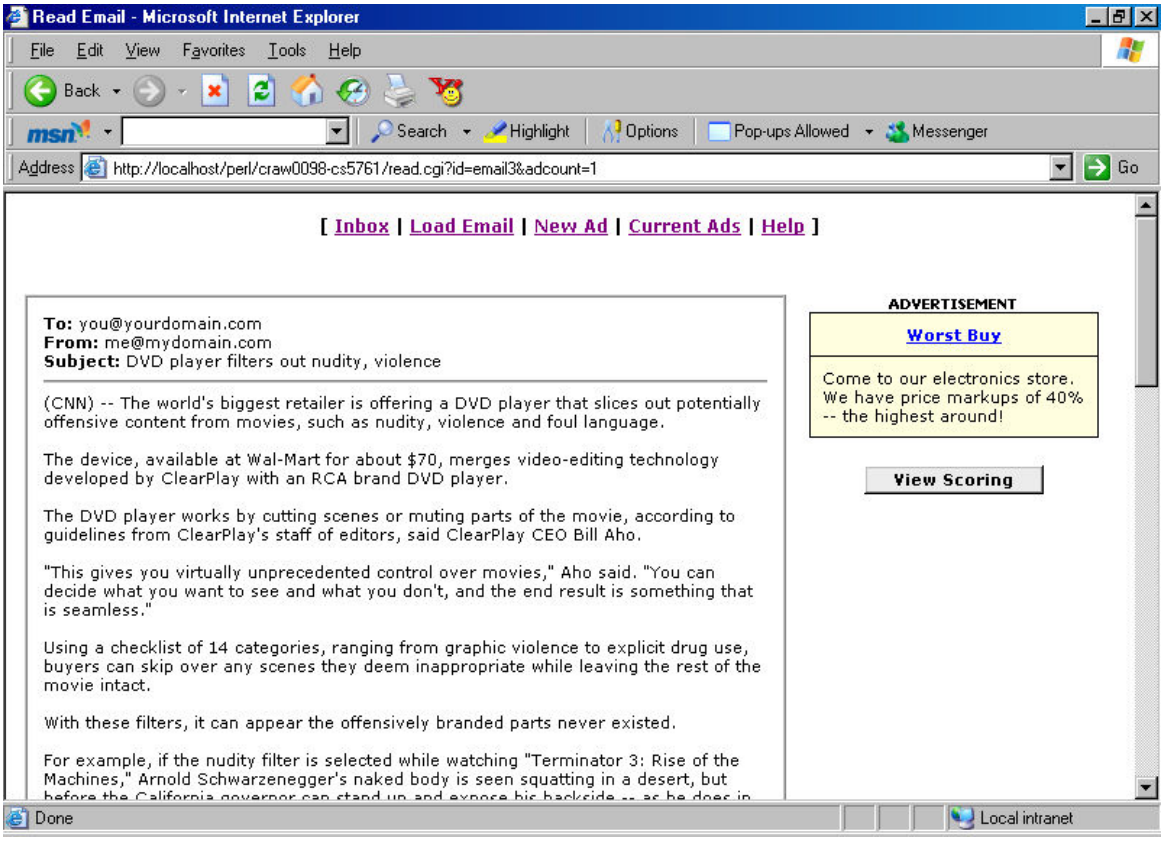
Load Email Interface

Here, a tester of the system can load a new email into the system. This functionality simulates the reception of a new email in a normal, everyday email client. The email subject and body will be saved for future use.



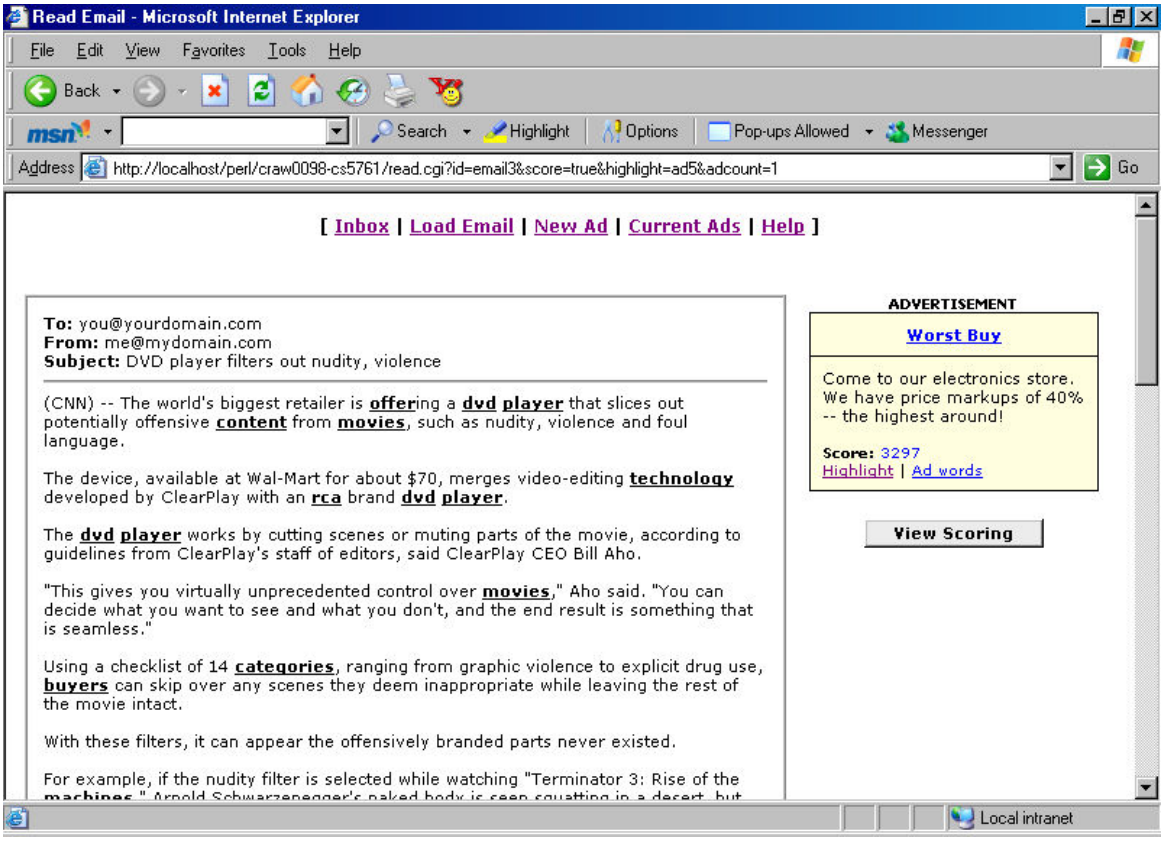
Ad Creation Interface

With this interface, testers of the system are offered the functionality to create new advertisements. To do so, simply follow the onscreen instructions. Specifying an accurate ad topic and keyword bootstrap are important for accurately linking ads to email content at a later time.



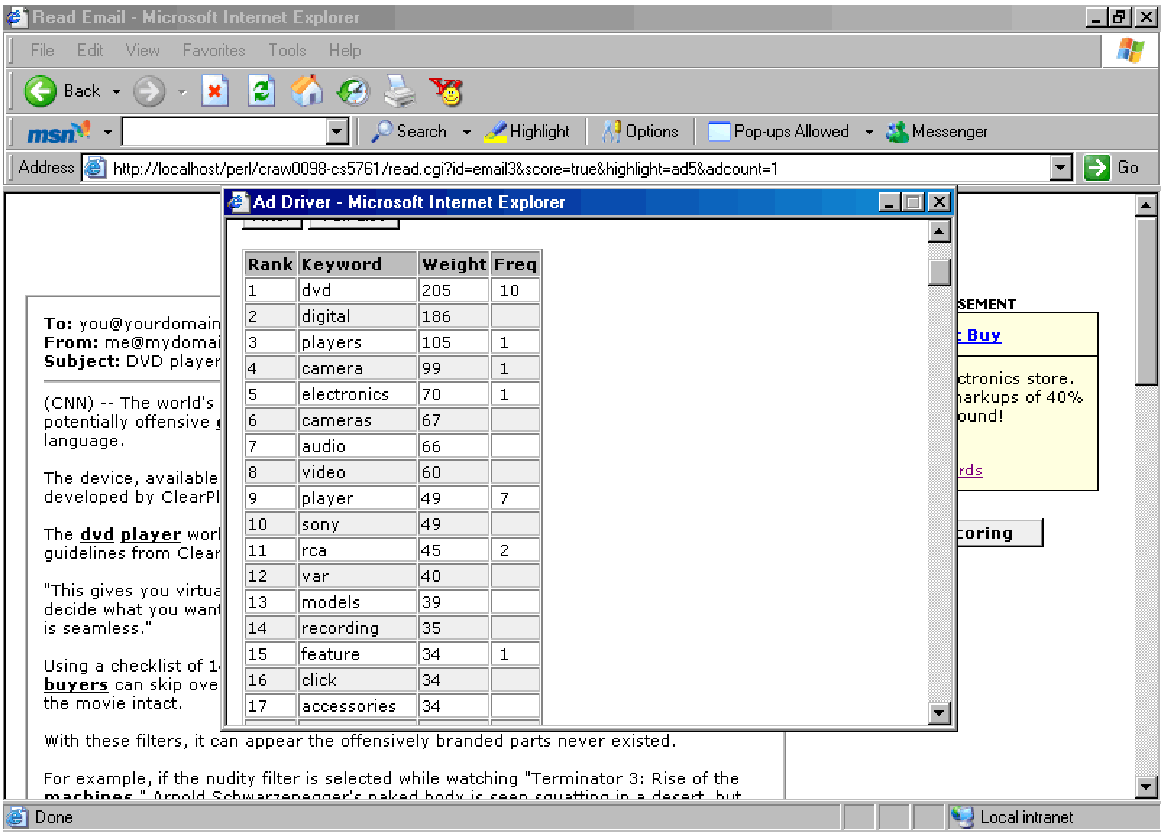
Email Reading

This interface simulates what a user would be looking at as he/she reads an email. Here, the user would see the email and an ad that directly relates to its content. An administrative option available is the "View Scoring" button. Clicking this button will bring up several options available for testing the system (see below).



Keyword Highlighting

With “View Scoring” clicked, the “Highlight” link becomes available. Clicking this link will highlight all of the words contained in the email body that were considered by the scoring algorithm and matched to the keyword index of this ad.



Ad Driver scoring breakdown

With “View scoring” turned on, the “Keywords” link becomes available. Clicking this link will open a popup window that lists the current breakdown of scoring for this ad.

Evaluation

Performing an evaluation of this system is rather straightforward. This is due to the fact that the output of a system like this is very predictable – allowing for a success/failure analysis to be easily made. For example, suppose the system holds ads that deal with topics A, B, and C. If topic C is discussed (directly or indirectly) in an email, the ad for topic C should be displayed. Neither A, B, or C should be displayed if topic D is discussed in an email. The number of pre-built advertisements saved in the system will be highly influential in the accuracy of linking ads to email content. Having few ads in the system means that the ad-email relationship is more likely to be loosely based. However, the opposite is true as well – more ads will lead to a stronger ad-email relationship.

For testing purposes, dummy emails were created and consisted of text from news stories. The context of the email messages are not as important as the requirement that they be based on particular topics. Quantitatively determining the success of the system was difficult. However, comparing the score for a particular ad against that of other ads (generated from the same email) allowed for solid conclusions to be drawn. Another approach used to determine the success was to simply gauge the relationship between the ad and the email content.

Example of Success

Hurts Rental Car

Rank	Keyword	Weight	Freq
1	airport	368	1
2	rental	261	5
20	agency	28	1
33	check	18	1
43	reservation	17	1
75	suv	13	1
84	agencies	12	1
139	fullsize	8	1
141	reserved	8	2
181	agent	6	2
187	options	6	1
194	hotels	6	1
212	agents	6	1
270	confirmation	5	1
283	counter	5	2

In reading an email titled "Travel: Your rights when things go wrong", whose subject matter obviously dealt with vacationing and traveling, an ad for "Hurts Rental Car" company was successfully linked with a score of 1830. The keywords listed to the left highly correspond with the content found in the email.

The main reason this ad was so successful was that keywords obtained by the system were not as likely to be ambiguous with ads for companies from other fields. A rental car company holds a fairly unique niche in the business world compared to some of the other cases described below.

Example of Moderate Success

An example where a moderate amount of success was obtained dealt with an article titled "RIAA sues 477 more people." This article talked about lawsuits being brought against people who download music illegally online. The two highest scoring ads were as follows for "Worst Buy" (441), a fictious electronics store, and for "Bye, Kregor and Goff" (390) attorneys at law. Their

results (in terms of how often words from their respective keyword indices occurred in the email body) are as follows:

Worst Buy

Rank	Keyword	Weight	Freq
14	recording	35	6
38	latest	22	2
40	technology	21	3
67	learn	15	1
102	computer	11	3
120	internet	10	3
140	copyright	9	3
149	software	8	1
226	policies	6	1
281	users	5	1

Bye, Kregor, & Goff: Attorneys at Law

Rank	Keyword	Weight	Freq
15	michigan	47	1
17	lawyers	46	1
42	internet	24	3
44	copyright	22	3
49	software	20	1
88	technology	13	3
106	virginia	11	1
111	connecticut	11	2
165	courts	8	1
213	san	6	1
217	jersey	6	1
221	defendants	6	2
228	texas	6	1
233	filed	6	2
242	lawsuits	6	2
320	learn	5	1

Here we can see that the ads are only somewhat related to the topic being discussed in the email. It would be a stretch to consider a relationships for either – although the email does discuss things based on technology and law.

Example of Failure

All Smiles Dental Service

Rank	Keyword	Weight	Freq
2	health	191	4
23	disease	31	6
42	current	20	1
73	expensive	14	1
124	workforce	11	1
136	cheap	10	1
154	related	9	1
168	team	9	1
175	cheaper	8	1
204	smoking	7	1
264	levels	6	1

An example of what can be considered a failure for this system can be seen here. An reading titled “Report finds heart disease a global threat”, was linked to an ad for the fictitious company “All Smiles Dental Service” and received a score of 1067. Its keyword breakdown is listed to the left.

You can see that the words commonly associated with the dental profession came back as being somewhat ambiguous with those of the field of medical cardiology. This caused the ad for the dental company to be scored somewhat high and be linked to the content found in the email body about heart disease.

286	unless	6	1
288	risk	6	1
309	require	6	1
339	eat	5	1

Conclusion

While evaluating this system, it became quickly evident that the correct linkage of ads to emails was highly dependent on the number of ads stored in the system. Plainly said, more ads means higher accuracy. The ambiguities found in the failure example mentioned above would disappear as more ads for specific topics were inserted into the system.

Determining the overall success of the system in a manner that can be easily understood by a skeptic is difficult. This can be seen by the fact that the ad in the failure example scored almost as high the one mentioned in the success example. However, if implemented in a "real-world" setting, success would be easily judged based on the number of times a user clicks on a given ad. With this information, the click count could be correlated with the scoring of the ad. From there, lower scoring ads could be easily adjusted accordingly.

References

1. H. Kimoto and T. Iwadera: *Construction of a dynamic Thesaurus and its use for associated information retrieval*. Pages 227-239, Brussels, September 1990.

2. Monika Henzinger, Bay-Wei Chang, Brian Milch, and Sergey Brin: I*nformation Retrieval: Query-free news search*, Proceedings of the 12th International World Wide Web Conference, pages 3-9, May 20-24, 2003.

3. Patrick Pantel and Dekang Lin: SpamCop: *A Classification & Organization Program*, Pages 4-6, University of Manitoba, March 1998.