IJWA

Generation of Targeted Advertisements for Online Social Networks

Pinaki Mitra[1], Kamal Baid[2]
[1]Department of Computer Science and Engineering
Indian Institute of Technology
Guwahati, India
pinaki@iitg.ernet.in

[2]Department of Computer Science and Engineering
Indian Institute of Technology
Guwahati, India
b.kamal@iitg.ernet.in

ABSTRACT: *Generating targeted advertisements for online social networks is a problem of growing interest. Monetizing activity in online social networks has been the topic of heated discussion lately. The undiscriminating tastes and spending power of a majority of its members makes this medium for self-expression and opinion sharing a very lucrative venue for advertising. The recent $240 million investment by Microsoft in Facebook clearly reaffirms the opportunity in targeted advertising for online social networks. Content-targeted advertisement programs such as Google AdSense and Yahoo Contextual Match work by automatically spotting keywords in web pages and displaying ads based on the keywords. The displayed ads are also refereed as Contextual Advertisements. These ads are generally represented by a URL along with a textual description that is also used to match the ad with the target page. This model works relatively well on the Web but its applicability to social networks is being met with some important challenges. User activity on venues such as forums, marketplaces and groups on social networking sites are excellent targets for monetization. We present an algorithm based on keyword clustering to generate targeted ads.*

Keywords: Informal text, User intents, Monetization

Received: 18 December 2009, Revised 18 February 2010, Accepted 24 February 2010

1. Introduction

Content-based Advertisement programs work by spotting keywords on a webpage or a search query and display advertisements based on the keywords. These advertisements are also called contextual advertisements. These have been very effective on web pages, largely because they are matched against content that a user is viewing. Not surprisingly, this mechanism of advertisement delivery was a good contender for social networking sites (SNSs) where advertisements need to match the content a user is viewing in order to trump the value of networking. Advertising on SNSs benefits both advertisers and the network. Advertisers get access to the desired target market and SNSs see monetization of their operations by way of advertising revenues. However, the utility of ad-models proposed to date is not yet apparent to the members of SNSs.

Besides trust and privacy issues, the content that is being exploited for ad generation is also an important reason why ads do not appeal to members of SNSs. The state-of-the-art in content- based advertising (CBA) on SNSs uses member demographics and stated interests on profiles for delivering ads. Although this information is very useful for launching product campaigns and micro targeting customers, they do not necessarily represent current or monetizable user intents; making ads based on this content inherently less relevant to the user. Over time, reduced user attention and incentive for clicking unrelated ad impressions is not a good scenario for both advertisers and social network providers. Please refer to [1] and [2] for recent developments in this area.

In this work, we argue that in addition to profile information, Ad programs should begin exploiting user activity on public venues on SNSs, such as forums and marketplaces where intents are readily expressed and representative of a user's current needs. With the growing popularity of online social networks, members are extensively using such venues to seek opinions from peers; write about things they bought; products and services they are looking for, or selling; offering advice to peers, talking about blogs they like to read etc. Today's online social networks are full of such user impressions; several of them with high monetization potentials being influenced socially to culminate in online transactions. We also think that user content authored on public forums will be less targets of privacy concerns, since posts are already on a public forum, open to everyone outside of a user's friend network.

Monetization of such user content however, is possible only when ads directly provide to a user's expressed needs. This entails understanding what the user is talking about (extracting key words and phrases) and the intents behind his post (looking for a product vs. sharing an opinion). The most straight-forward approach to these problems is to treat user posts on SNSs as any content on the Web and apply traditional keyword extraction and intent identification techniques. However, there are some key characteristics of content on SNSs that make well-known techniques less effective in many cases. Here, we discuss some of them.

Content Characteristics

1. A characteristic of communities, both online and offline, is the shared understanding and context they operate on. On two popular social networking sites, MySpace and Facebook, we observed a frequent usage of slangs and variations of entity names. Puters for computers and Skik3 for the product Sidekick3 are two examples. As a related work on identifying keywords on online broadcasting content indicates, frequency based methods like tf-idf alone are not effective in spotting such keywords. Not being able to spot keywords that advertisers might want to target implies fewer targeted ads.

2. Due to the interpersonal and interactional nature of a social networking platform, when users share information, they are typically sharing an experience or event. The main message is overloaded with information that is off-topic for the task of advertising. Consider this post from the Computers forum on MySpace. Not eliminating noisy keywords like "Merrill Lynch" and food poisoning, results in less targeted ads.

Topic: I NEED HELP WITH SONY VEGAS PRO 8!!
Post : Ugh and i have a video project due tomorrow for merrill lynch :(all i need to do is simple: Extract several scenes from a clip, insert captions, transitions and thats it. really. omgg i cant figure out anything!! help!! and i got food poisoning from eggs. its not fun. Pleasssse, help?):

3. Users scribe on SNSs with different intentions. For instance, Post 1 below from a group on Facebook shows a clear transactional intent which has a high potential for monetization. Re- ply 1 and Post 2 however, share an opinion and present less potential for monetization. Identifying monetizable posts is an important problem toward generating ads that users are more likely to click.

Topic 1: iTouch
Post 1: i am looking for a 16 or 32 GB iTouch. hoping to get it cheaper then what apple sells it for. Reply 1: try ebay or marketplace.

Topic 2: The new mac air
Post 2: yh its pretty good... very convinient to use once you have the softwares put in because it doesnt come with a lot of things eg Microsoft office and all.

In this work, we present a content-analysis system that addresses two issues in utilizing content outside of a user's profile for targeted ad generation.

1. Identifying monetizable user intents and

2. Eliminating off-topic content so only the most relevant keywords are used for ad generation.

2. Proposed Solution

There are two main components to our system. The first component is a crawler that crawls, cleans and ingests user posts from SNSs. The second component spots keywords in a post, compensates for misspellings and named entity variations and eliminates off-topic content. The resulting sets of most relevant keywords in a post are provided to Ad programs for ad generation1. All components in our system are built using domain independent techniques and resources, which mean that content from any domain, can be processed with no additional effort. Techniques are time and resource efficient and easily reproducible. All training and test data are obtained from recent user activity on MySpace.

2.1 Crawling User Posts

The first component is responsible for crawling user posts from social network pages. Implemented using Java's URL and regular expression packages, a fetcher gathers pages while a parser extracts user posts, timestamps and category the post was crawled from (e.g., Electronics forum). User ids are not crawled for privacy reasons. Crawled user posts include a title and a post thread. A title for a post is mandatory, and a thread is comprised of a first post and optional response messages.

In the rest of this report, we use he term post to refer to any single post and a post thread to refer to posts and replies in a thread. Crawled posts are ingested into text files striping html tags and removing any images or advertisements found on the page.

MySpace has Groups and Forums where members share opinions and seek information, blogs, where members place buy/sell ads, artists pages where users express opinions about artists and their work etc. All data used in this work was crawled from venues with transactional posts - MySpace Electronics, Computers and Gadgets forums. For each of the three MySpace Forum the number of posts is 100.

2.2 Keywords for Advertising

The second component of our content-analysis system focuses on identifying the most relevant keywords in a user authored post. We look to addressing two prevalent characteristics of content on social networking sites.

1. misspellings, slangs and variations of entity names and

2. eliminating off-topic noise in user posts.

The goal is to supply only the most relevant keywords to Ad programs. Our experiments show that such processing of content generates ads that are more targeted than those generated by using the content as is. There are two main parts to this component: spotting keywords and then eliminating off-topic noise.

Venue on SNS	Number of Posts
MySpace Computers Forum	100
MySpace Electronics Forum	100
MySpace Gadgets Forum	100

Table 1. Crawl Statistics

2.2.1 Extracting Keywords

The task of this component is to extract key words and phrases (henceforth referred to as key- words) from an ingested user post. There has been a plethora of work on extracting keywords from text. In a recent work, [3] and [4] showed that traditional tf-idf features along with the distribution of words in a query log are most useful features in identifying keywords for advertising on Web pages. Work by [5] on identifying keywords in broadcast content for advertising used language patterns and frequency-based methods to extract keywords.

Keyword extraction however, is not a contribution of our work. One can think of this component as a placeholder that can be replaced by any keyword extraction algorithm. For this work, we use the Yahoo Term Extractor [6] (YTE), a keyword extraction tool that uses Yahoo's search API. The tool takes as input a text snippet and returns key words and phrases spotted in the text.

As the YTE uses an index built on the Web, it offers a high precision and recall in spotting keywords. To test YTE's efficacy on user posts, we hand marked keywords in 100 posts from MySpace. YTE's recall in spotting keywords was as low as 43 that it failed to spot keywords in ill-formed, fragmented sentences, or when the keywords are misspelled, or are variations that are not frequent on the Web. This is a common drawback of techniques that rely on word frequencies alone. To compensate for this, we built a simple edit-distance based keyword spotter over the YTE.

Round 1: The first round of the algorithm uses only the Yahoo Term Extractor (YTE) to spot keywords. As the algorithm processes every post in the dataset, it saves every spotted unique keyword in a global dictionary G. We build G using the 4000 posts crawled from MySpace (not overlapping with the 100 manually annotated posts).

Round 2: Using a basic window-based spotting technique backed by the global dictionary G, the second round goes through every post again and spots keywords missed in the first round. Similar window and dictionary based information extraction techniques have been used in the past with fairly reasonable success [7]. Let us call the keywords that were missed in the

first round as variants, for variations of keywords. The goal of the algorithm is to spot a variant in a post and also record its transliteration. The transliteration of a keyword is the commonly occurring form of the word which was spotted in Round 1 and is in G; e.g. computer for cmputr.

The algorithm tries to find variants of every keyword gi in G. Using a sliding window of length equal to the number of words in gi, the algorithm extracts a window of words from the post. The Levenshtein string similarity based on edit distance is computed between the window of words and gi. Given two sequences x[1..m] and y[1..n] and a set of transformation operation costs the edit distance between x to y is the cost of the least expensive operation that tranforms x to y. If the similarity score is ≥ 0.85, the algorithm treats the window of words as a variant and records the transliteration gi as the keyword spotted for that post. Using the 4000 posts for generating G, recall and precision numbers for the 100 manually annotated posts are shown in Table 3. Recall is defined as the fraction of keywords marked by both annotators that were spotted by the system.

Precision is the fraction of keywords spotted by the system that were indeed marked as keywords by both the annotators. False positives in Round 2 are rare because spotting uses high string similarity thresholds and is built on G which has high precision spots from the YTE. Recall improved by the second round is directly proportional to the size of the domain-independent dictionary G. Results are satisfactory in spite of a small worsening of precision.

Post	how much do u think a polariod electric eye costs?
Keywords after Round1	electric eye
Variant, Transliteration	polariod, polaroid
Final Keywords after Round2	electric eye, polaroid

Table 2. Spotted keywords

No. of words in G	90051
Precision, Recall of Round 1	71% 52%
Precision, Recall of Round 2	68.4% 75%

Table 3. Spotting Keywords - Performance

2.2.2 Eliminating Off-Topic Content

Once keywords in a post have been spotted, the next step is to identify and eliminate off-topic noise. This is an important contribution of our work in delivering highly targeted ads given the frequent off-topic noise found in user authored content in the social network media. Here, we present an unsupervised clustering algorithm that uses counts from the Web, instead of a domain corpus, to separate informative from non-informative keywords in a post.

The clustering algorithm is based on principles of mutual information concept. The basic idea is to use the title of a user post, which is mandatory, has a word limit and is typically representative of the post's content for finding relevant keywords in the post. The algorithm starts by placing all keywords spotted in the title in cluster C1 and all keywords spotted in a post thread (main post and replies) in cluster C2. The clustering algorithm evaluates every keyword in C2 and calculates its association strength with keywords in C1, i.e., it measures how strongly related words in the post are to words in the title. Association measured in terms of mutual information and context scopes are used to pick keywords from C2 to add to C1. The order in which keywords are added from C2 to C1 is based on the greedy heuristic that would least affect the change of characteristics of the present C1 cluster. At the end of the algorithm when all keywords in C2 have been considered, C1 houses all informative, non-noisy keywords in the post, which are used by Ad programs to generate ads. First, we cover relevant mutual information concept preliminaries, then describe the statistical clustering algorithm, show examples and discuss drawbacks of the algorithm.

Preliminaries

Mutual Information (MI) models the dependency between two random variables and is widely used in natural language processing to discover the association strength between words [8]. We measure the association between title and post

keywords using MI. The formal definition of MI between two random variables $W_i < V$ and $W_j > V$ is,

$$I(W_i,W_j) = \sum_{w_i,w_j \in V} p(w_i,w_j)\log(p(w_i,w_j)/(p(w_i)p(w_j)) \quad (1)$$

Where V is a vocabulary of words and $p(w_i,w_j)$ is the joint probability of w_i and w_j.

The MI between two particular words w_i and w_j is therefore a point-wise realization of w_i and w_j and can be computed as

$$\begin{aligned} I(w_i,w_j) &= p(w_i,w_j)\ \log(p(w_i,w_j)/(p(w_i)p(w_j)) \\ &= p(w_i)p(w_j|w_i)\ \log(p(w_i,w_j)/(p(w_i)p(w_j)) \end{aligned} \quad (2)$$

Where $p(w_j|w_i)$ is the conditional probability of the word w_j given the word w_i.

If there is a strong association between w_i and w_j, $I(w_i,w_j) < 0$; if there is no significant relationship between the words, $I(w_i,w_j) < 0$ and if the words are in complementary distributions or unrelated, $I(w_i,w_j) > 0$. In this work, $p(w_j,w_i)$ is the probability of w_j co-located (either proceeding or following) with word w_i within a window. Unlike the standard bi-gram used in language modeling that requires that words occur in a sequence, we are interested only in association strengths between words and therefore, can ignore word order. Maximum likelihood estimates of the parameters are calculated as

$$p(w_i) = n(w_i)/N;\ \ p(w_j|w_i) = n(w_i,w_j)/n(w_i) \quad (3)$$

where $n(w_i)$ is the count of word w_i on the Web,

$n(w_i,w_j)$ is the co-occurrence count of words,

w_i and w_j, N is the number of tokens in the Web.

We query AltaVista [9] to obtain word counts, approximating number of pages a word occurs in for the number of times it occurs. Co-occurrence counts were obtained using AltaVista's NEAR operator that returns the number of pages where two words appear within ten words of each other. The entire process of obtaining counts was automated using a script that generates search terms for all words and word pairs in C1 < C2 for all posts, issues an AltaVista query for each search term and saves results in a map Counts to be used by the cluster algorithm. Plugging (3) and (4) into (2), we have the mutual information between two words as,

$$I(w_i,w_j) = (n(w_i,w_j)/N)\log(n(w_i,w_j)N/(n(w_i)n(w_j))) \quad (4)$$

From the formula above, one can see that this measure is symmetric, i.e., $I(w_i,w_j) = I(w_j,w_i)$ for two co-occurring words w_i and w_j.

Here and elsewhere, when n(wi,wj) = 0, their mutual information then is zero. The title cluster C1 is expanded gradually to include relevant keywords from the post cluster C2. To decide which keyword to pick from C2 to add to C1, we use the concept of mutual information (MI) and information content (IC) of a cluster. The MI of cluster C1 is defined as the sum of pairwise mutual information of words within the cluster,

$$I(C1) = \sum I(w_i,w_j) \qquad w_i,w_j \in C1 \quad (5)$$

The information content (IC) of a cluster C1 is the average of the pairwise mutual information of words within the cluster or the cohesiveness of a word cluster.

$$IC(C1) = I(C1)/^{|C1|}C2 \quad (6)$$

Where |C1| denotes the cardinality of the cluster C1 and $^{|C1|}C2$ is the number of word pairs in the cluster C1. This normalizes for clusters of different sizes. As the algorithm expands C1, IC(C1) is computed before and after adding a keyword from C2 to C1. The change in IC(C1) while adding a keyword k, is called Information Gain. It is measured as,

$$IG(C1,k) = IC(C1,k) - IC(C1)$$

Where IC(C1, k) is the information content of C1 after adding keyword k from C2.

IG(C1, k) is positive when adding the new k increases the cohesiveness between keywords within the cluster, i.e., k is strongly associated with words in C1 and negative when k is unrelated to words in C1. The IC score of the new cluster (after adding

a keyword) is also indicative of the relative strength of the co-occurrence relationships between the words in the cluster. A keyword k that has a high association strength with words in C1 and therefore higher IC(C1, k) scores tends to occur in minimally constrained or generic contexts with the other words. Otherwise stated, a keyword occurring in very general contexts with words in C1 will increase IC(C1) relatively more than a keyword that occurs in narrower, specific contexts. For example, if C1 has the keyword [’speakers’], the keyword ’beep’ that occurs in maximally constrained or specific contexts of mal- functioning ’speakers’ has relatively lower association strengths with C1 compared to a keyword ’logitech’ that occurs in minimally constrained or wider contexts with ’speakers’. The clustering algorithm uses this relationship between change in information content of C1 and context to pick keywords from C2 to add to C1.

Eliminating Off-topic Content

1. Post Title: camcorder C1: [’camcorder’]
2. Main Post: yeah i know this a bit off topic but the other electronics forum is dead right now. im looking for a good camcorder, somethin not to large that can record in full HD only ones so far that ive seen are sonys

Reply: Canon HV20. Great little camera under $1000.

C2: [’electronics forum’, ’hd’, ’camcorder’, ’somethin’, ’canon’, ’little camera’, ’canon hv20’, ’cameras’, ’off topic’].

3. IG(C1, k) of C1 and C2 keywords.
4. Eliminated Keywords: [’somethin’, ’off topic’, ’electronics forum’].
5. Final C1 using maximally constrained contexts: [’camcorder’, ’canon hv20’, ’little camera’, ’hd’, ’cameras’, ’canon’].
6. Final C1 using minimally constrained contexts: [’camcorder’, ’canon’, ’cameras’].

[’camcorder’, ’canon’] : 0.0001515
[’camcorder’, ’canon hv20’] : 0.00001135
[’camcorder’, ’cameras’] : 0.00009694
[’camcorder’, ’somethin’] : -0.000000001298
[’camcorder’, ’hd’] : 0.000079658
[’camcorder’, ’off topic’] : -0.0000000197
[’camcorder’, ’little camera’]: 0.0000290
[’camcorder’, ’electronics forum’]: -0.00000006713

Table 4. IG(C1, ki)

Algorithm for Generating Informative Keyword Clusters

We use a running example shown in Table 4 to explain the clustering algorithm. Inputs to the algorithm are cluster C1, which initially has all keywords spotted in the title, cluster C2 that has all keywords from the post thread and the map of word and word pair counts, Counts, obtained by querying AltaVista. The only assumption we make is that all keywords in C1 are informative to the topic at hand. The first iteration of the algorithm measures the change in IC(C1) when keywords k_i from C2 are added to it. Using Eqn. (7).

$$IG(C1, k_i) = IC(C1, k_i) - IC(C1) \quad \forall\ i \in C2 \tag{8}$$

Bullet 3, Table 4 shows the computed IG(C1, k_i) scores for all words in C2. At this time, the algorithm also eliminates keywords k_i that resulted in a negative IG(C1, k_i) score. This step is performed only at the first iteration when C1 has only title keywords. The intuition is that if post keywords are unrelated to the informative title keywords, they will not contribute to the subsequent steps, given that the algorithm gradually builds the title keyword cluster(see Bullet 4). Next, the algorithm greedily adds the keyword k_i that occurs in maximally constrained or specific contexts with words in C1, i.e., has the lowest, but positive IG(C1, k_i) score. This keyword is added to C1 and removed from C2. The intutive reason to add keywords from C2 to C1 in this ascending order of IG values is to add those keywords first from C2 that would affect

the least change in the characteristics of C1. The reasoning would be more clear if we conceive IC values as the average edit distance between two elements in a cluster. The algorithm proceeds to consider the remaining words in C2, greedily adding keywords that have the lowest IC(C1, k_i) scores at every step. The algorithm terminates when it has either evaluated all keywords in C2 or when no more keywords result in positive IG(C1, k_i) scores, i.e., no more keywords in C2 are strongly associated with those in C1.

Bullet 3 and 4 in Table 4 show the first iteration of the algorithm, where some keywords are eliminated and 'canon hv 20' with the lowest, positive IG(C1, k_i) score is first added to C1. Bullet 5 shows the result informative cluster C1 obtained using this greedy strategy.

An alternate strategy was to greedily add the keyword k_i that occurred in minimally constrained or generic contexts with words in C1. Bullet 6 shows the resulting cluster C1 using this strategy. As one can see, the former strategy has the tendency of adding specific to general keywords while the later picks the most general keyword first and runs out of keywords that add to the information content of C1. Work in studying user search and buying behavior online informs that people use generic keywords when they are in the exploratory phase and specific phrases in the buying stage . Our system uses the first strategy in expanding the informative cluster so we have as many related, specific keywords in the pool for the task of targeted advertising.

Drawbacks of the Algorithm

An important drawback of this algorithm is the fundamental assumption that title keywords in C1 are informative in nature. When this is not the case, results of our algorithm are poor. Also, when no keywords are spotted in the title, we fall back on category level information for a post; for example, the name of the forum it was crawled from. However, these tend to be as broad topic words, such as 'Electronics' or 'Computers' and do not selectively pick informative keywords from the post. A possible solution to deal with an empty title cluster is to use frequency based techniques to identify and use a possible informative word from the post.

Algorithm Complexity

Using title keywords as starting points reduces the context space from all keywords to a few title keywords. The worst-case complexity of our algorithm is $O(N^2)$ where N is the number of unique spotted keywords, a fairly small number in a document. Social Media platforms generate content of varied lengths with blogs being the wordiest, followed by discussion forums, message boards and chats. A recent article found an average of 526 words (not keywords) per post across five blog sources. The 220 test posts from MySpace and Facebook had an average of 9 spotted keywords per post and an algorithm execution time of 4.3ms per post. Measuring performance upper bounds on blogs and reducing time complexity is a focus of near term research.

3. Results and Discussions

The goal of our experiments is to highlight the importance of using only contextually relevant keywords for content delivery. Using Google AdSense that matches contents of web pages with advertisements, we show that contextual keywords (returned by our algorithm) help AdSense deliver more relevant ad suggestions. We used 57 posts (42 from MySpace and 15 from Facebook's test dataset) for this experiment. These posts had atleast one spotted keyword in the title, less that ten keywords in the post for ease of user evaluation and atleast three keywords, so there was chance of off-topic content. First, all 57 posts were processed by our keyword spotting and cluster algorithm to extract contextual keywords. Next, two sets of ads were generated for each post using Google AdSense. The first set, Adsc, contained ads generated from the content as it is. The second set, Adsk, contained ads generated using keywords returned by our algorithm. Snapshots of ads for all posts were captured on a single day and stored offine. Each post had a maximum of 8 ads, 4 in each set. Users responded by picking ads that they thought were relevant to the post. For 54 of the 57 posts, ads generated using contextual keywords were just as or more relevant than ads generated using the content as it is. Our algorithm did poorly only on three posts, where title clusters did not have contextually relevant keywords. Contextual keywords generated just as many relevant ads as content for 23 posts; one additional relevant ad for 12 posts; twice as many relevant ads for 10 posts; three times as many relevant ads for six posts and four times as many relevant ads for three posts. As we can see, for 54 generated ads using our approach instead of using the content is - a clearly indicates the importance and effectiveness of our algorithm.

3.1 Discovering Frequent Sets of Keywords

Another component of our system finds frequent set of keywords. To do this we are using a recursive elimination algorithm which is similar to apriori algorithm. Initially we calculate support for all keywords. Keywords having support less than some threshold are removed. Thus the support for all sets of keywords is calculated in the increasing of the cardinality and if it is greater than a user specified minimum threshold we call the set frequent otherwise we eliminate the set and all its supersets.

4. Conclusion and Future Works

There are some drawbacks of the clustering algorithm as explained earlier. One is the fundamental assumption that title keywords in C1 are informative in nature, when this is not the case, results of our algorithm are poor. Other, when no keywords are spotted in the title, we fall back on category level information for a post. Improving the cluster algorithm is an important part of our future work.

As explained in Algorithm Complexity section that the worst-case complexity of our algorithm is $O(N^2)$ where N is the number of unique spotted keywords. So reducing time complexity can be a focus of near term research.

One more problem in content-targeted advertising is vocabulary impedance. The problem occurs because an ad generator can't contain all the keywords in its databases. When a keyword is purchased by a company, that keyword and corresponding ad is put into database. A company can purchase limited number of keywords. So in this case, even if we provide informative keywords to ad-generator, we get unrelated ads. Solving this problem is also a part of future work.

References

[1] Boyd, D. M., Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship, Journal of Computer-Mediated Communication, 13 (1) 210-230, November.

[2] Yan, J., Liu, N., Wang, G., Zhang, W., Jiang, Y., Chen, Z. (2009). How much can behavioral targeting help online advertising?, *In:* WWW 2009, p. 261-270.

[3] Freitag, D. (1998). Information extraction from html: Application of a general machine learning approach, *In*: Proceedings of the Fifteenth National Conference on Artificial Intelligence. AAAI Press, p. 517-523.

[4] Turney. P., Canada, C. (1997). Extraction of keyphrases from text: Evaluation of four algorithms, National Research Council, Institute for Information Technology, Tech. Rep.

[5] Li, H., Zhang, D Hu, J., Zeng, H.-J., Chen, Z (2007). Finding keyword from online broadcasting content for targeted advertising, *In*: ADKDD '07: Proceedings of the 1st international work-shop on Data mining and audience intelligence for advertising. New York, NY, USA: ACM, p. 55-62.

[6] Yahoo term extraction service. http://developer.yahoo.com.

[7] Craven, M., Dipasquo, D., Freitag, D., Mccallum, A.,Mitchell, T., and. Nigam, K (1998). Learning to extract symbolic knowledge from the world wide web. AAAI Press. p. 509-516.

[8] Church, K. W., Hanks, P. (1989). Word association norms, mutual information, and lexicography, *In*: Proceedings of the 27th annual meeting on Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, p. 76-83.

[9] Altavista: Altavista advanced search cheat sheet, Altavista company, Palo Alto, California. http://www.av.com.