Monetizing User Activity on Social Networks - Challenges and Experiences

Meenakshi Nagarajan*, Kamal Baid†, Amit Sheth* and Shaojun Wang*

*kno.e.sis center, Wright State University, Dayton, OH, USA; meena, amit, shaojun @ knoesis.org
†Dept. of Computer Science and Eng., Indian Institute of Technology, Guwahati, India; b.kamal@iitg.ernet.in

Abstract—This work summarizes challenges and experiences in monetizing user activity on public forums on social network sites. We present a approach that identifies the monetization potential of user posts and eliminates off-topic content to identify the most relevant and monetizable keywords for advertising. Preliminary studies using data from MySpace and Facebook show that 52% of ad impressions generated using keywords from our system were more targeted compared to the 30% relevant impressions generated without using our system.

Keywords- user activity, off-topic noise, content contextual ads

I. INTRODUCTION

Current advertising approaches to monetizing user content on social networks are profile-based contextual advertisements, demographic-based ads or a combination of the two. Demographic-based ads target an individual by age, gender and location information. Profile-based ads exploit information such as interests and activities on user profiles for delivering ads. Profile-based ads are a type of content-based ads that are generated by automatically finding relevant keywords on a page and displaying ads based on those keywords. Content-based ad delivery was made popular on the Web where ads matched content that a user was viewing on a web page. Not surprisingly, this model was a good contender for social networking sites (SNSs) where ads need to be highly targeted to the content in order to trump the value of networking. However, the utility of ad-models proposed to date on SNSs is not yet apparent to its members. Besides issues of trust, privacy and scattered user attention on SNSs, the content that is being exploited for ad generation is also an important point of concern. While profile information might be useful for launching product campaigns and micro-targeting customers, it does not necessarily contain current interests or purchase intents. Ads generated from such content are inherently less relevant to a user. Over time, this leads to a scenario where ad campaigns see several ad impressions but very few clickthroughs.

With the growing popularity of online social networks, members are extensively using public venues like forums, marketplaces and groups to seek opinions from peers, write about things they bought, offering advice and so on. Intents expressed on these venues are often times representative of a user’s current needs and in several cases, monetizable. Content from public forums is also less likely to be a target of privacy concerns, given that posted content is not personal information. In this work, we posit that in addition to using profile information, ad programs should generate profile ads (ads shown on a user profile) from user activity on public venues on SNSs. While the intuition is rather straightforward, there are challenges that need to be addressed before such content can be used for monetization.

1. User Intentions: Users scribe on SNSs with different intentions. For example, Post 1 below shows a clear transactional intent, while Post 2 shares an opinion.

Post 1: I am looking for a 32 GB iTouch, cheaper then what apple sells it for.
Post 2: MSoffices convinient to use once you have the softwares put in

The problem of identifying user intents in such free-text is not the same as that of identifying intentions behind web queries. We observed that unlike web search, the use of certain entity types does not accurately classify a post’s intents. For example, the presence of a product name does not immediately imply navigational or transactional intents.

The same product X can appear with several user intentions on SNSs - ‘i am thinking of getting X’ (transactional); ‘i like my new X’ (information sharing); and ‘what do you think about X’ (information seeking). Among the many footprints users leave on a site, it is important to identify those with high monetization potential, so as to generate profile ads that they are more likely to click.

2. Informal content and off-topic noise: A characteristic of communities, both online and offline, is the shared understanding and context they operate on. Use of slangs and variations of entity names, such as puters for computers and Sidekick3 for the product Sidekick3 are commonplace. Not being able to spot such keywords in posts will mean fewer matched ad impressions. Additionally, due to the interactional nature of social networking platforms, 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’ will potentially result in ads unrelated to the post.

I NEED HELP WITH SONY VEGAS PRO 8!! Ugh and i have a video project due tomorrow for merrill lynch ..and i got food poisoning from eggs, its not fun. help?

Contributions: Here we present our experiences with building a system that (a) identifies monetizable user activity or posts and (b) eliminates off-topic noise in these user posts, so only the most relevant keywords are used for generating
ads. Figure 1 shows the components in our content analysis system. The first component crawls, cleans and ingests user activity or posts from SNSs. The second component assesses the monetization potential of a post by scoring monetizable intentions. We approach the problem of identifying monetizable user intentions using “action patterns” - words surrounding an entity X instead of using the entity itself. Posts identified as monetizable in nature are then consumed by the third component that spots keywords, compensates for misspellings and named entity variations and eliminates off-topic content. The resulting relevant keywords are finally provided to ad programs for ad generation.

The intuition behind this work is that generating ads from monetizable keywords in user activity will generate more attention or clickthroughs compared to ads based on profile information. Ideally, large scale clickthrough experiments would be needed to evaluate if user attention to ads will indeed convert to clickthroughs. Since we do not have access to the private profiles of users on SNS (activity information is currently public) or to the systems that generate profile based ads, comparing state-of-the-art in advertising on SNSs to the activity-based ads that we propose is non-trivial. Instead, we concentrate on evaluating our system on a subset of the problem - how well are we able identify monetizable user posts and how do ads generated using the keywords identified by our system compare to ads generated using the user post as is? In other words, if an Ad delivery program were to use the proposed approach over the content found on SNSs, what can they expect to find?

For all our experiments, we used Google AdSense to generate content-based ads. To the best of our knowledge, AdSense is the only ad program that allows members to place a script against content they want to display ads for. From our experiments, we found that 52% of ads generated after using our system were deemed more targeted to the content in a user post compared to the 30% of ads generated without using our system that were considered relevant by a set of 30 users evaluating 60 random user posts crawled from MySpace and Facebook. While results from the experiment are based on small sample sizes, they are clearly indicative of the potential and feasibility in exploiting user activity for profile ad generation.

II. CRAWLING, INGESTING USER POSTS

The component is responsible for crawling user posts from social networking sites. Some of the public venues for self-expression on MySpace include Groups and Forums where members share opinions and seek information, Blogs, Classifieds where members place buy/sell ads etc. Facebook has a Marketplace similar to MySpace Classifieds and Groups similar to MySpace Groups. All data used in this work was crawled from venues with transactional intents - three MySpace forums and two Facebook Marketplace forums. Table I shows statistics of data collected. Text portions of 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 replies. In the rest of this paper, we use the term post to refer to any single post and a post thread to refer to posts and replies in a thread.

III. IDENTIFYING MONETIZABLE INTENTS

In this section, we describe the algorithm used by the second component for identifying monetizable user posts. We find the monetization potential of a user post by identifying intents that are monetizable. Perhaps, the most closely aligned work to ours is the identification of intentions in web search queries [1] where search intents were classified as being navigational (locate a webpage), transactional (obtain a product) or informational (locate a website covering a topic) in nature. The authors implemented a classifier that used a dictionary of entities - company, product, people names etc., and manually derived characteristics of intent types to achieve 74% accuracy in classifying search intents.

Our approach to identify intents behind user posts is different from this entity centric approach and relies on patterns that surround the entity. We observed that unlike web search, the use of certain entity types does not accurately classify a post’s intents. For example, the presence of a product name X does not immediately imply navigational or transactional intents. It can occur with three different intentions - ‘i am thinking of getting X’ (transactional); ‘i like my new X’ (information sharing); and ‘what do you think about X’ (information seeking). We believe that “action patterns” - those surrounding entity X, are better predictors of intent types for this data than the entity itself.

Examining posts from MySpace and Facebook, we found that user intents behind posts classify under information sharing, information seeking and transactional intents or a combination of the three and can be defined as follows: A. Information Seeking: The intent of information seeking is to solicit responses concerning a question that addresses the information needs of the user. The query can be one

<table>
<thead>
<tr>
<th>Venue on SNS</th>
<th>No. Posts</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySpace Training data Crawled July 18, 2008</td>
<td>8000</td>
</tr>
<tr>
<td>MySpace Computers Forum</td>
<td>2000</td>
</tr>
<tr>
<td>MySpace Gadgets Forum</td>
<td>2000</td>
</tr>
<tr>
<td>MySpace Test data Crawled May 23, 2008</td>
<td>100</td>
</tr>
<tr>
<td>MySpace Electronics Forum</td>
<td></td>
</tr>
<tr>
<td>Facebook Test data Crawled Oct 8, 2008</td>
<td>120</td>
</tr>
<tr>
<td>Electronics ‘To Buy’ Marketplace</td>
<td></td>
</tr>
<tr>
<td>Electronics ‘To Sell’ Marketplace</td>
<td>540</td>
</tr>
</tbody>
</table>
asking for information toward the end goal of comparisons, transactions, locating a webpage, etc.

B. Information Sharing: The intent of information sharing is to inform. Users are typically sharing information or opinion about a product, an experience, promotions etc.

C. Transactional: The intent of transactional posts is to express an explicit buy, sell or trade intents. The goal is to seek responses that will provide cues leading to an offline (outside-network) transaction.

We define a user activity to be monetizable only if it contains explicit monetizable user intents, i.e., transactional or information seeking intents where users are looking for information that advertisers can exploit. The approach we take for identifying monetizable intents relies on finding transactional and information seeking “action patterns” in user posts that suggest the presence of the intent. The first task is to learn a set of information seeking “action patterns” from a corpus that can be used to assess the information seeking intent behind any user post. We propose a bootstrapping algorithm that starts with a set of seed information seeking patterns and learns new patterns from an unlabelled training corpus of 8000 MySpace user posts. New patterns are classified as information seeking in nature depending on their functional and distributional similarity with the seed patterns. This technique is similar in spirit to pattern induction algorithms that have been used successfully for information extraction tasks. In the second step, the learnt patterns along with cues indicating the presence of transactional intents are used to identify the monetization potential of any test post.

A pattern: A pattern in our work is a 4-gram word pattern that is indicative of an information seeking intent. One can think of these patterns as asking questions of something, to someone. We care more about how the questions are asked (“action patterns”) as opposed to topic words that indicate what the question is about. In the post, “where can I find a chotto psp cam”, we are interested in the ‘where can I find’ pattern rather than the ‘find a chotto psp cam’ pattern. We experimented with several n-grams, reporting results of which is outside the scope of this paper. We found that 3-grams were ambiguous while 5-grams were redundant in the amount of information we wished to capture.

Extracting similar patterns: Here, we outline the bootstrapping process for learning information seeking intent patterns.

1. Initially, all 4-gram patterns are extracted from the training set of 8000 MySpace user posts using an implementation of the Bayardo pruning method [2] and ordered by frequency. Patterns with frequency \( \geq 3 \) are placed in a universal pattern pool \( P_u \). It is important to note that this corpus is not labelled for presence of any type of intention.

2. A list of 5 seed wh-question words (why, when, where, how and what) are placed in a word pool \( W \). Using these seed words, all 4-gram patterns in \( P_u \) that extract these seed words (i.e. contain seed words) are selected, sorted by frequency and placed in a candidate pattern pool \( P_c \). The user picks the top 10 most frequent seed patterns in \( P_u \) that are added to a information seeking pattern pool \( P_{is} \). The only condition we place is that every seed word get a representation in the pool \( P_{is} \). Table II shows examples of seed patterns in \( P_{is} \). The goal of the algorithm is to expand the pool \( P_{is} \) by finding new information seeking intent patterns from \( P_c \).

3. A function \( f \) is used to assign an information seeking intent score between 0 and 1 to every pattern considered for inclusion into \( P_{is} \). Patterns scoring higher than a tuneable threshold are included in \( P_{is} \) and removed from \( P_u \); those scoring lower than a threshold are discarded.

4. Alongside patterns, the algorithm also learns keywords that are likely to induce other information seeking patterns. These words are added to \( W \) and the algorithm continues from step 2, populating 4-gram patterns from \( P_u \) to \( P_c \).

At a certain point, the algorithm has considered all patterns in \( P_{is} \) and has found no new patterns or words. To prevent the algorithm from stagnating, we infuse new patterns from the candidate pattern pool \( P_c \) to \( P_{is} \). \( P_c \) also has patterns like ‘what I meant is’, that contain seed words but no information seeking intents. It is important that we pick patterns that have a high likelihood of being information seeking in nature to avoid extracting erroneous patterns in subsequent steps. Candidate patterns are chosen using the same function \( f \) that decides the information seeking score for a pattern. The algorithm terminates when the three following conditions hold: No new words are added to \( W \), no new information seeking patterns are added to \( P_{is} \) and there are no more patterns to infuse from \( P_c \).

Pattern Similarity: The algorithm uses a subset of the Linguistic Inquiry Word Count dictionary (LIWC)[3] to selectively extract and score new patterns. The LIWC is a dictionary and a text analysis program initially designed to identify words that tapped emotional and cognitive dimensions. The recent, considerably expanded version of LIWC captures over 86 percent of the words used in writing and speech with 4,500 words and word stems, and approximately 80 categories including linguistic dimensions, words tapping psychological constructs (affect, cognition), personal concern categories (work, home), paralinguistic and punctuation dimensions (assents, fillers, commas).

We use words from three LIWC categories - cognitive mechanical (e.g., if, whether, wondering, find), adverbs (e.g., how, somehow, where) and impersonal pronouns (e.g., someone, anybody, whichever). We find that these three word types occur frequently in information seeking patterns when people are discussing a thought process, as in, ‘I am thinking about getting X’ or asking others for information, as in, ‘Someone tell me where can I find X’. The bootstrapping algorithm uses rules over these word types to define a function that gradually expands the pattern pool \( P_{is} \).

Extracting and scoring patterns: The first step in the
Table II

EXTRACTING AND SCORING NEW PATTERNS

1. \( P_{is} = \{\text{"does anyone know how", \"where do i find", \"someone tell me where"} \}
2. \( \rho_{is} = \{\text{"does anyone know how"} \}
3. Wildcard patterns = \{\"* anyone know how", \"does * know how", \"does anyone * how" and \"does anyone know *\} 
4. Pattern under focus = \{\"* anyone know how"\}; no matching patterns found
5. Pattern under focus = \{\"* anyone know how"\}; substitution word = \"someone\"

<table>
<thead>
<tr>
<th>Matched pattern</th>
<th>replacement pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>{&quot;* anyone know how}</td>
<td>{&quot;someone}</td>
</tr>
</tbody>
</table>

Scoring replacement words: If \( \text{support}_{r} \geq 0.25 \) (\( r \) extracts at least 25% of the patterns in \( P_{is} \)), \( \rho_{is} \) is considered information seeking in nature and added to \( P_{is} \).

Over iterations, single-word substitutions and functional usage considerations conservatively expand the pattern pool \( P_{is} \). Table II shows part of the extraction and scoring process using a sample pool \( P_{is} \) of three patterns, two new extracted and scored patterns, one of which is added to \( P_{is} \) and the other rejected during the first pass.

Scoring replacement words: At the end of the first iteration, the algorithm has considered all 10 seed patterns in \( P_{is} \), extracted, scored and added new patterns to \( P_{is} \), discarded some patterns and found candidate (replacement) words to add in the word pool. At this stage, we consider if any of the replacement words have a tendency to extract more information seeking patterns in subsequent steps, i.e., can they be added to \( W \)? All replacement words are ordered by their \( \text{support}_{r} \) score (percentage of information seeking patterns they extract) and the top one word that extracts at least 50% of the patterns is added to the word pool \( W \). We use a higher threshold here as words are weaker indicators of intents than patterns and adding an incorrect word to \( W \) can add potentially false information seeking patterns to \( P_{is} \).

The bootstrapping process continues with new words in \( \bar{W} \) extracting new patterns in \( P_{is} \). We re-iterate through all patterns in \( P_{is} \) even if they had been considered in previous iterations. We do this so patterns that were missed in previous steps due to low \( \text{support}_{r} \) scores can be reconsidered given the new patterns in the current iteration.

Infusing new patterns: At a certain point, the algorithm has considered all patterns in \( P_{is} \) and has found no new patterns or replacement words. To prevent the algorithm from stagnating, we INFUSE new patterns from the candidate pattern pool \( P_{c} \) to the information seeking pool \( P_{is} \). Let us call the pattern that is being infused into \( P_{is} \) \( \rho \). To assess the information seeking strength of \( \rho \), we use rules similar to those defined for replacement words and calculate two scores \( \text{funct}_{w} \) and \( \text{support}_{w} \) for every word \( w \) in pattern \( \rho \). If \( w \) is in one of the LIWC categories, \( \text{funct}_{w} = 1 \); else \( \text{funct}_{w} = -1 \). If \( \text{funct}_{w} = 1 \), we calculate \( \text{support}_{w} \) which is the percentage of information seeking patterns in \( P_{is} \) extracted by \( w \). If there are at least two words in \( \rho \) (other than the seed word) whose \( \text{funct}_{w} = 1 \) and \( \text{support}_{w} \geq .25 \), \( \rho \) is considered highly likely to be an information seeking pattern. For all such patterns, we calculate \( \text{score}_{\rho} \), reflective of the pattern's information seeking intents in terms of its words: where, \( \frac{1}{3} \) normalizes the score

\[
\text{score}_{\rho} = \frac{1}{3} \sum_{w \in \rho} \text{support}_{w}, \text{ such that } \text{funct}_{w} = 1 \text{ and } w \notin W
\] across the three other words in the 4 gram pattern \( \rho \).

The intuition behind using the functional properties of words is to gradually expand the pattern pool with words that are used in similar semantic contexts. The rules that operate over these scores above to assess the information seeking nature of the pattern \( \rho_{is} \) are the following:

1. If \( \text{funct}_{r} == -1 \), the pattern \( \rho_{is} \) is discarded to trash pool \( \bar{T} \) and removed from the pool it was extracted from. An example of this is \{\"does anyone know you\} that matched \{\"does anyone know \*\} with \{\"how\} as the substitution word
2. If \( \text{funct}_{r} == 1 \), the pattern is considered for the next rule over \( \text{support}_{r} \).
top one pattern with the highest score $p_i$ is infused into $P_{is}$ and the algorithm continues. The algorithm terminates when all the three following conditions hold: No new words are added to $W$, no new information seeking patterns are added to $P_{is}$, and there are no more patterns to infuse from $P_c$.

Using our corpus of 8000 MySpace user posts, 3608 4-gram patterns that occurred at least three times in the corpus, five seed words and ten seed patterns that are extracted by the seed words, the algorithm extracted a total of 309 unique new patterns. Of these 263 were evaluated by human annotators as accurate information seeking patterns, for an accuracy of 85%.

**Identifying Monetizable Posts**

The bootstrap algorithm generates information seeking patterns $P_{is}$ offline. During the run-time processing of a crawled post, our system uses $P_{is}$ to identify information seeking intents and a LIWC ‘Money’ dictionary to measure the transactional intent behind a post. The ‘Money’ category in LIWC has 173 words and word forms indicative of transactions, e.g., trade, deal, buy, sell, worth, price etc.

**Calculating information seeking intent score:** The algorithm uses a sliding window technique over a post to extract a window of 4 words and compares it with every extracted pattern $p_{is}$ in $P_{is}$. Using regular expression patterns and preserving word order, the algorithm counts the number of words in the post window that match a pattern $p_{is}$. If all 4 words in the post window and the pattern match in order, the information seeking intent score of the pattern is incremented by 1. If not, word order is forcibly ignored - if at least two words match, the score is incremented by \( \text{number of word matches} / 4 \). The final score is normalized over the total number of 4-gram patterns in the post to normalize for posts of varying lengths.

**Calculating Transactional intent score:** Transactional intents are those where users express explicit buy, sell or trade intents. The transactional intent score of a post is measured by simply using the number of words in a post that match entries in the ‘Money’ dictionary using regular expression patterns. For every word that matches, the score is incremented by 1. The final score is normalized over the total number of words in the post. The total monetizable potential of a post is calculated as a sum of the information seeking and transactional intent scores.

**Evaluating with Facebook Marketplace:** We measure the efficacy of our intent identification component using the 120 posts from ‘To Buy’ Electronics Marketplace on Facebook where intents behind user posts are pre-classified. All posts in the ‘To Buy’ set had information seeking and transactional intents. Our algorithm obtained 83% recall in identifying ‘To Buy’ posts, i.e. it was able to identify about 99 of the 120 posts as containing monetizable intents (intent scores $> 0$). The reader should note that the patterns were learnt from user posts on MySpace but tested on those from Facebook, confirming the generality of effect of the learnt patterns. The performance of the intent identification algorithm could improve if it learnt more candidate patterns from a larger and varied training corpus. Evaluating how the size of a training set affects the algorithm is outside of the scope of this work, but a part of our future investigations.

**IV. Keywords for Advertising**

Once monetizable posts are identified, the next step is to identify keywords from these posts for advertising and further discard keywords that are off-topic. Spotting keywords in text is a well-studied problem. Keyword extraction, named entity identification, information extraction etc. accomplish this goal using different strategies (see survey at [4]). Spotting keywords however, is not our focus. In this work, we used the Yahoo Term Extractor [5] (YTE), an off-the-shelf keyword extraction service built over Yahoo’s search API. YTE takes as input a text snippet and returns key words and phrases in text. We chose YTE over frequency based techniques since we did not want to be limited by counts from a 12000 post corpus for tf.idf calculations. Also, a recent work comparing YTE, tf.idf and mutual information based techniques for word and phrase identification concluded that YTE did better than tf.idf when identifying top $k < 4$ keywords in a document and all three were similar in characterizing document content for larger values of $k$ [6].

To test YTE’s efficacy on crawled posts, we marked keywords in 100 test posts from MySpace using two human annotators who were instructed to mark names of products, services and category names such as books, car, camera etc. Recall and precision were calculated against annotations that both users agreed upon. With an inter-annotator agreement of 0.59, YTE’s recall and precision were 52% and 71% respectively. YTE failed to spot keywords that were misspelled or were variations not frequent on the Web. To compensate for this, we built a simple edit-distance based spotter over the YTE spotted keywords, similar to dictionary based window spotting techniques used in the past [7].

**Round 1.** The first round processes all 12000 training posts from MySpace using YTE and saves unique spotted keywords in a global dictionary $G$.

**Round 2.** The second round examines every post again and spots keywords missed in the first round. Using a sliding window of length equal to the number of words in every keyword $g_i$ in $G$, the algorithm extracts a window of words from the post. The Levenshtein string similarity [8] is computed between the window of words and $g_i$. If this score is $\geq 0.85$, $g_i$ is recorded as a spotted keyword. An advantage of the second phase is that non-common forms of keywords are transliterated to the common version spotted by YTE in Round 1. Results after Round 2 are satisfactory considering that recall increased by 23% and precision reduced only by 2.6% for the 100 annotated posts.
IDENTIFYING CONTEXTUAL KEYWORDS

The next step is to identify keywords that are related to the main discussion and eliminate those that are off-topic. One solution to this problem is to use tf.idf to rank discriminatory terms in a document higher. However, not all discriminatory terms are necessarily relevant to the discussion (see sample at [9]). A more promising approach is to cluster words that have strong semantic associations with one another, namely words that are called to mind in response to a given stimulus, thereby separating strongly related and unrelated keywords. One way to measure semantic associations is to use word co-occurrence frequencies in language. Creating word clusters using co-occurrence based association strengths have been used in the past for assigning words to syntactic and semantic categories, learning language models etc.

However, generating semantically cohesive keyword clusters still does not indicate which clusters are relevant to the discussion. To overcome this, we use a simple heuristic of assuming title keywords, as in blog titles, to be good indicators of context. Using these keywords as stimulus, our algorithm expands the context by including content keywords that are strongly associated with the title keywords.

Our clustering algorithm starts by placing all title keywords in cluster \( C_1 \) and content keywords in cluster \( C_2 \). The idea is to gradually expand \( C_1 \) by adding keywords from \( C_2 \) that are strongly associated with \( C_1 \). At every iteration, the algorithm measures the change in Information Content (IC) of \( C_1 \), \( IC(C_1, k_i) \delta \), before and after adding a keyword \( k_i \) from \( C_2 \) to \( C_1 \). The keyword that results in a positive and minimum \( IC(C_1, k_i) \delta \) score is added to \( C_1 \) and removed from \( C_2 \). Additionally, keywords resulting in negative \( IC(C_1, k_i) \delta \) scores are discarded as off-topic. The algorithm terminates when all keywords in \( C_2 \) have been evaluated or when no more keywords in \( C_2 \) have positive \( IC(C_1, k_i) \delta \) scores (no strong associations with \( C_1 \)).

Word association strengths are measured using the information theoretic notion of mutual information. Word co-occurrence counts are obtained from the Web using AltaVista. First, we describe preliminaries and then detail the clustering algorithm using an example shown in Table III.

The algorithm starts by adding every keyword from \( C_2 \) to \( C_1 \) and measuring the change in Information Content (IC) of \( C_1 \). \( IC(C_1) \) is the strength of the semantic associations between words in the cluster and is defined as the average pairwise Mutual Information (MI) of the words.

\[
IC(C_1) = \frac{MI(C_1)}{|C_1|/2}
\]

(1)

where \(|C_1|\) denotes the cardinality of the cluster \( C_1 \) and \(|C_1|/2\) is the number of word pairs in the cluster \( C_1 \), normalizing for clusters of different sizes. \( MI(C_1) \) is the Mutual Information of cluster \( C_1 \), defined as the sum of pairwise Mutual Information of words within the cluster.

\[
MI(C_1) = \sum_{w_i, w_j \in C_1, i \neq j} MI(w_i, w_j)
\]

(2)

Recall that \( w_i \) or \( w_j \) can be a single word or a phrase. The MI of words \( w_i, w_j \in C_1 \) measures their association strength in terms of their co-occurrence statistics. It is defined as the point-wise realization of the MI between two random variables \( W_i \) and \( W_j \in V \), a vocabulary of words[10].

\[
MI(w_i, w_j) = p(w_i, w_j) \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}
\]

(3)

\[
= p(w_i) p(w_j | w_i) \log \frac{p(w_j | w_i)}{p(w_j)}
\]

Standard definition for point-wise mutual information ignores the joint probability term, \( p(w_i, w_j) \) in (3). We keep this term to ensure the consistency of (2). Here, \( p(w_j | w_i) \) is the probability of \( w_j \) co-located with word \( w_i \) (preceding or following) within a window. Unlike standard bi-gram models in language modeling that require words to occur in a sequence, we do not care about word order. Maximum likelihood estimates of the parameters are calculated as

\[
p(w_i) = \frac{n(w_i)}{N}; p(w_j | w_i) = \frac{n(w_j | w_i)}{n(w_i)}
\]

(4)

where \( n(w_i) \) is the frequency 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 available on the Web. Due to lack of recent statistics, we use a conservative estimate of \( N \approx 70 \) billion calculated for AltaVista in 2003[11].

Word and word pair frequency estimates are obtained by querying AltaVista. We use AltaVista mainly for its NEAR functionality for obtaining counts for co-occurring words. This operator constrains Web search to documents containing two words within ten words of one another, in either order. When obtaining counts for phrases we use “double quotes” around it. The process of obtaining frequency estimates is conducted offline and automated using a script that generates search terms for all words and word pairs in \( C_1 \cup C_2 \) and issues Altavista queries. Plugging (4) into (3), we have the MI between two words as shown in (5).

\[
MI(w_i, w_j) = \frac{n(w_i, w_j)}{N} \log \frac{n(w_i, w_j)}{n(w_i)n(w_j)}
\]

(5)

As every keyword \( k_i \) is added from \( C_2 \) to \( C_1 \), the change in Information Content of \( C_1 \) is measured as

\[
IC(C_1, k_i) = IC(C_1) - IC(C_1)_{k_i}
\]

(6)

where \( IC(C_1, k_i) \) is the information content of \( C_1 \) after adding keyword \( k_i \) from \( C_2 \). \( IC(C_1, k_i) \) is positive when \( k_i \) is strongly associated with words in \( C_1 \) and negative when \( k_i \) is unrelated to words in \( C_1 \). Bullet 3, Table III shows the computed \( IC(C_1, k_i) \) scores for words in \( C_2 \) at the end of the first iteration.

At this time, the algorithm eliminates keywords that result in negative \( IC(C_1, k_i) \) scores (Bullet 4). This is done only at the first iteration when \( C_1 \) has only title keywords. The
intuition is that if context keywords are unrelated to the context-indicating title keywords, they will not contribute to subsequent steps that build the title keyword cluster.

Next, the keyword that results in a positive and minimum \( IC(C1, k_i) \) score, ‘canon hv20’ in this example, is greedily added to \( C1 \). The reasoning behind the pick is as follows. A keyword \( k_i \) occurring in specific contexts with words in \( C1 \) will increase the Information Content of the \( C1 \) relatively less than a keyword that occurs in generic contexts. If \( C1 \) has the keyword ‘speakers’, keyword ‘beep’ that occurs in maximally constrained or specific contexts of malfunctioning speakers will have lower association strengths with \( C1 \) compared to a keyword ‘logitech’ that occurs in minimally constrained or broader contexts with ‘speakers’. As the algorithm continues, the keyword occurring in a maximally constrained context with \( C1 \) is removed from \( C2 \) and added to \( C1 \) at every iteration. This strategy has the tendency of adding specific to general keywords from \( C2 \) to \( C1 \) (Bullet 5). The alternate strategy is to greedily add the keyword that occurs in minimally constrained or generic contexts with \( C1 \). This tends to pick generic keywords first and runs out of keywords that add to the Information Content of \( C1 \) (Bullet 6). In our experiments we use the first strategy to have as many related, specific keywords for generating ads.

The algorithm does poorly when the assumption that title keywords are always contextual in nature does not hold or when no keywords are spotted in the title. If title keywords have low association strengths with all content keywords, it is an indication of non-contextual title keywords. When no keywords are spotted in the title, we use all title words (minus stopwords) to seed \( C1 \). If the words are too generic, they do not selectively pick contextual keywords from the content. In both cases, a viable option is to ignore our algorithm and use the content as is.

Algorithm Complexity: Using title keywords as starting points reduces the context space from all keywords to a few title keywords. The best case running time of our algorithm is \( O(MN) \) where \( M = |C1| \), size of the title cluster and \( N = |C2| \), size of the content cluster. This occurs when all keywords in \( C2 \) are off-topic or only one \( C2 \) keyword is contextually relevant and one iteration after computing \( MN \) association strengths suffices to partition the keywords. Worst case complexity is \( O(MN^2) \) when there are no off-topic keywords and the algorithm has to evaluate all \( N \) keywords in \( C2 \) one after another, computing \( MN \) association strengths at every step, for \( N \) iterations. Multiple words resulting in similar Information Content change scores in the same iteration could potentially be added to \( C1 \) to reduce the time complexity, an important focus of our future investigations.

### V. Experiments and Evaluation

The two hypotheses guiding our work were that (a) user activity outside his/her profile is more representative of a user’s current interests and monetizable and (b) eliminating off-topic noise in user activity will generate more targeted ads than using the content as is. Ideally, large scale click-through experiments would be needed to evaluate if user attention to ads generated from profile activity will indeed convert to clickthroughs. Since we do not have access to the private profiles of users on SNS or to the systems that generate profile based ads, comparing state-of-the-art in advertising on SNSs to the activity-based ads that we propose is non-trivial. Instead, we concentrate on evaluating our system on a subset of the problem - how well are we able to identify monetizable user posts and how do ads generated using the keywords identified by our system compare to ads generated using the user post as is? In other words, if an Ad delivery program were to use the proposed approach over the content found on SNSs, what can they expect to find?

The goal is to highlight the importance of using only contextually relevant keywords for content delivery. Using Google AdSense that matches content on web pages with advertisements, we show that contextual keywords (returned by our algorithm) help AdSense deliver more relevant ad suggestions. We randomly picked 60 crawled user posts (45 from MySpace and 15 from Facebook’s test dataset) for this experiment. These posts had positive monetizable scores, atleast one spotted keyword in the title, less that ten keywords, so there was chance of off-topic content. We recruited 30 graduate students arbitrarily and briefed them on the problem and experiment.

First, all 60 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, Ads\(_{k\_cont} \), contained ads generated from the content as is. The second set, Ads\(_{k\_ct} \), contained ads generated using keywords returned by our algorithm. Snapshots
of ads for all posts were captured on a single day and stored offline. Each post had a maximum of 8 ads, 4 in each set. See [9] for sample ad snapshots. The 60 user posts were divided into ten sets, each with six posts, with each set being evaluated by three randomly chosen users. We also generated a third set of ads using only the title keywords. The number of ad impressions in this set for each post was sparse, owing to the vocabulary impedance problem [12], and was therefore excluded from the experiments.

Each user was shown a set of six posts one after another. For each post, users were also shown ads from the two sets, Ads\textsubscript{c} and Ads\textsubscript{k}, randomly arranged with checkboxes to indicate preferences. Users were instructed to read every post and accompanying ads (url and text) and click the checkbox against only those ads they thought were relevant to the post. They were also instructed to comment on whether the posts were monetizable in nature. Sample instructions and a user response can be found at [9].

**Results:** All users agreed unanimously that the posts had monetizable content. Users also responded by picking ads that they thought were relevant to the post. We aggregated responses for the 60 posts by counting the number of ads that users picked from each set. We counted only ads that two or more evaluators picked to ensure at least a 50% inter-valuator agreement. Table IV shows statistics for the total number of ads displayed for all posts and their keywords and the number of ads users picked as relevant from the two sets. Users thought that 52% of the ads shown using keywords returned by our algorithm were relevant, compared to the 30% of relevant ads generated using the content as is.

For several posts however, Ads\textsubscript{c} and Ads\textsubscript{k} had ads in common. A more accurate measure of user feedback is the number of ads that were deemed relevant and were unique to each set. Table IV also shows these statistics. According to evaluator picks, processing content using our algorithm led to 22% more targeted unique ads. Additionally, for 54 of the 60 posts, ads generated using contextual keywords were just as or more relevant than ads generated using the content as is. Our algorithm did worse only on three posts, where title clusters did not have contextually relevant keywords. Contextual keywords generated just as many relevant ads as did the content for 23 posts; one additional relevant ad for 15 posts; twice as many relevant ads for 10 posts; three times as many relevant ads for 6 posts and four times as many relevant ads for 3 posts. As we can see, for 56% of the posts, our algorithm enabled more relevant ad generation than using the content as is - a clear indication of the importance and effectiveness of our algorithm.

**VI. Discussion and Conclusion**

In this work, we experimented with user activity on public forums in SNSs as possible targets for monetization. We presented a system that enables generation of profile ads from user activity by assessing the monetization potential of the activity and then identifying relevant keywords for advertising. While our experiments are preliminary, they strongly suggest the potential and feasibility in utilizing user activity for profile ad generation. While generating targeted ads is essential for converting ad impressions to clickthroughs, it is not a sufficient condition. Placement of ads on already crowded profile pages, discarding objectionable user activity etc. will gain more focus with the use of such user-generated content. We are currently deploying our system as an application on the Facebook platform to measure the potential of this work on a larger scale.

**References**


