Targeted Content Delivery for Social Media Content[*]

Meenakshi Nagarajan
Knoesis Center
Wright State University
Dayton, Ohio, USA
meena@knoesis.org

Kamal Baid
Department of Computer Science and Engineering
Indian Institute of Technology
Guwahati, India
b.kamal@iitg.ernet.in

Amit Sheth, Shaojun Wang
Knoesis Center, Wright State University, Dayton, Ohio, USA
amit.sheth@wright.edu
shaojun.wang@wright.edu

ABSTRACT

Spotting contextually relevant keywords is fundamental to effective content suggestions on the Web. In this regard, misspellings, entity variations and off-topic discussions in content from Social Media pose unique challenges. Here, we present an algorithm that assists content delivery systems by identifying contextually relevant keywords and eliminating off-topic keywords. A preliminary user study over data from MySpace and Facebook clearly suggests the usefulness of our work in delivering more targeted content suggestions.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Contextual Content Delivery, Social Media Content

Keywords

Mutual Information, Contextual keywords

1. INTRODUCTION

Content Delivery is the task of complementing content that a user is viewing (Web search results or a Web page) with related content such as advertisements, similar articles, RSS feeds, images, tags and so on. Suggested content is pushed to a user because it is deemed relevant to the content the user is viewing and with the goal of minimizing his information seeking efforts. Typically, content delivery involves spotting keywords in the content being consumed and matching those with keywords in the content being delivered. More sophisticated techniques append spotted keywords with synonymns or category level metadata to deliver additional content. Zemanta[1] is one such content delivery application that offers suggestions on user blogs by matching what a user writes with a database of pre-indexed multimodal content to deliver related text links, images and tags.

Compared to traditional online media, content on Social Media poses unique challenges for content delivery. User-generated content on blogs, discussion forums etc. tends to be more informal compared to content found in scientific or news articles. Given the interactional purpose to communication in Social Media, fragmented sentences, misspellings and entity variations are commonplace. Typically, users are also sharing an experience which results in the main message being overloaded with off-topic content. These characteristics, more prevalant in Social Media than elsewhere on the Web, affect the accuracy in identifying contextual keywords, i.e., keywords that are relevant to the main discussion. This in turn affects content suggestions that are matched against identified keywords. Poor suggestions impair user experience, are intrusive and over time, reduce user attention. Consider these examples shown at [2] where the presence and elimination of off-topic keywords significantly affects the relevance of content suggestions.

The contribution of this work is a simple yet effective algorithm to **accurately identify contextual keywords**, i.e, keywords that are relevant to the main discussion in the content a user is viewing, and eliminate off-topic keywords. The goal is to assist content delivery systems in generating more relevant or targeted content suggestions. The algorithm is based on well-founded principles of information theory and is applied after keywords have been identified in content and before suggestions are made.

As a test case, we evaluate the algorithm on posts from discussion forums on social networking sites. Data on these sites are good representatives of off-topic chatter given the majority teen and tween user demographic. Using Google AdSense for content delivery, we evaluate the targeted nature of content suggestions with and without using our algorithm. According to user evaluations over 57 posts, our algorithm results in 22% more targeted content suggestions.

2. REACHING CONTEXTUAL KEYWORDS

The task before us is to identify keywords in content that are relevant to the main discussion. As a first step, we spot keywords and phrases (henceforth referred to as keywords) and then identify contextually relevant keywords while eliminating off-topic ones. Data for this work was crawled from three MySpace forums and a Facebook 'To Buy' Marketplace forum (see Table 1).

[*]Supported by a Microsoft External Research award 'Beyond Search, Semantic Computing and Internet Economics'

[1]http://www.zemanta.com/about/

Table 1: Crawl Statistics

Venue on SNS	No. of Posts
Training Data - MySpace Computers, Electronics, Gadgets	8000, 2000, 2000 resp.
Test Data - MySpace Electronics	100
Test Data - Facebook Electronics	120

Spotting Keywords and Phrases

Spotting keywords in text is a well-studied problem. Keyword extraction [9], named entity identification [7], information extraction [5] etc. accomplish this goal using different strategies. Spotting keywords however, is not our focus. In this work, we used the Yahoo Term Extractor [3] (YTE), an off-the-shelf keyword extraction service built over Yahoo's search API. YTE uses an index built off the Web, takes as input a text snippet and returns key words and phrases in text. We chose YTE because we did not want to be limited by frequencies from the 12000 post corpus for tf.idf calculations. Also, a recent work comparing YTE, tf.idf and mutual information techniques for keyword identification concluded that YTE did better than tf.idf in identifying *top* $k < 4$ keywords in a document and all three were similar in characterizing document content for larger values of k [10].

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 mispelled or were variations not frequent on the Web. To compensate for this, we built a simple edit-distance based spotter over YTE spotted keywords, similar to dictionary based window spotting techniques used in the past [8].

Round 1. The first round processes all 12000 training posts from MySpace using YTE and saves unique spotted keywords (lowercased) in a global dictionary $\mathcal{G}$.

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 $\mathcal{G}$, the algorithm extracts a *window of words* from the post. The Levenshtein string similarity[1] is computed between the lowercased *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 main contribution of our work is to identify keywords that are related to the main discussion and 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 [2]). 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 seperating strongly related and unrelated keywords. One way to measure semantic associations is to use word co-occurence frequencies in language. Creating word clusters using co-occurence based association strengths have been used in the past for assigning words to syntactic and semantic categories, learning language models and so on.

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 $C1$ and content keywords in cluster $C2$. The idea is to gradually expand $C1$ by adding keywords from $C2$ that are strongly associated with $C1$. At every iteration, the algorithm measures the change in Information Content (IC) of $C1$, $IC(C1, k_i)_\delta$, before and after adding a keyword k_i from $C2$ to $C1$. The keyword that results in a positive and minimum $IC(C1, k_i)_\delta$ score is added to $C1$ and removed from $C2$. Additionally, keywords resulting in negative $IC(C1, k_i)_\delta$ scores are discarded as off-topic. The algorithm terminates when all keywords in $C2$ have been evaluated or when no more keywords in $C2$ have positive $IC(C1, k_i)_\delta$ scores (no strong associations with $C1$).

Word association strengths are measured using the information theoretic notion of **mutual information**. **Word co-occurence counts** are obtained **from the Web** using AltaVista. First, we describe preliminaries and then detail the clustering algorithm using an example shown in Table 2.

The algorithm starts by adding every keyword from $C2$ to $C1$ and measuring the change in Information Content (IC) of $C1$. $IC(C1)$ 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.

$$IC(C1) = MI(C1) / \left(\begin{array}{c} |C1| \\ 2 \end{array} \right) \qquad (1)$$

where $|C1|$ denotes the cardinality of the cluster $C1$ and $\left(\begin{array}{c} |C1| \\ 2 \end{array} \right)$ is the number of word pairs in the cluster $C1$, normalizing for clusters of different sizes. $MI(C1)$ is the Mutual Information of cluster $C1$, defined as the sum of pairwise Mutual Information of words within the cluster.

$$MI(C1) = \sum_{w_i, w_j \in C1, i \neq j} MI(w_i, w_j) \qquad (2)$$

Recall that w_i or w_j can be a single word or a phrase. The MI of words $w_i, w_j \in C1$ measures their association strength in terms of their co-occurence 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[4].

$$\begin{aligned} MI(w_i, w_j) &= p(w_i, w_j) \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \qquad (3) \\ &= p(w_i)p(w_j|w_i) \log \frac{p(w_j|w_i)}{p(w_j)} \end{aligned}$$

Standard definition for point-wise mutual information ignores the joint probabilty 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 (preceeding or following) within a window. Unlike standard bi-gram models in language modeling that require words to occur in

Table 2: Eliminating Off-topic Noise and Reaching Contextual Keywords

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’, ‘camera’, ‘off topic’]

3. $\mathbf{IC(C1},k)_\delta$ scores of$C1$and C2 keywords:

[‘camcorder’, ‘canon’]	:0.00015	**[‘camcorder’, ‘canon hv20’]**	:0.000011	[‘camcorder’, ‘camera’]	:0.00009
[‘camcorder’, ‘hd’]	:0.000079	[‘camcorder’, ‘little camera’]	:0.000029	[‘camcorder’, ‘electronics forum’]	:-0.00000006
[‘camcorder’, ‘somethin’]	:-0.0000000012	[‘camcorder’, ‘off topic’]	:-0.000000019		

4. Eliminated Keywords: [‘somethin’, ‘off topic’, ‘electronics forum’]

5. Final C1 using maximally constrained contexts: [‘camcorder’, ‘canon hv20’, ‘little camera’, ‘hd’, ‘camera’, ‘canon’]

6. Final C1 using minimally constrained contexts: [‘camcorder’, ‘canon’, ‘camera’]

a sequence, we do not care about word order. Maximum likelihood estimates of the parameters are calculated as

$$p(w_i) \quad = \quad \frac{n(w_i)}{\mathcal{N}}\,; p(w_j|w_i) = \frac{n(w_i, w_j)}{n(w_i)} \qquad (4)$$

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; $\mathcal{N}$ is the number of tokens available on the Web[2].

Word and word pair frequency estimates are obtained by querying AltaVista. We chose AltaVista mainly for its NEAR functionality for obtaining counts for co-occuring 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 $C1 \cup C2$ and issues Altavista queries.

Plugging (4) into (3), we have MI of two words as shown in (5). This measure is symmetric, i.e., $MI(w_i, w_j) = MI(w_j, w_i)$. When $n(w_i, w_j) = 0$, we define $MI(w_i, w_j) = 0$.

$$MI(w_i, w_j) \quad = \quad \frac{n(w_i, w_j)}{\mathcal{N}} \log\Big(\frac{n(w_i, w_j)\mathcal{N}}{n(w_i)n(w_j)}\Big) \qquad (5)$$

As every keyword k_i is added from $C2$ to $C1$, the change in Information Content of $C1$ is measured as

$$IC(C1, k_i)_\delta = IC(C1, k_i) - IC(C1) \qquad (6)$$

where $IC(C1, k_i)$ is the information content of $C1$ after adding keyword k_i from $C2$. $IC(C1, k_i)_\delta$ is **positive** when k_i is **strongly associated** with words in $C1$ and **negative** when k_i is **unrelated** to words in $C1$. Bullet 3, Table 2 shows the computed $IC(C1, k_i)_\delta$ scores for words in $C2$ at the end of the first iteration.

At this time, the algorithm eliminates keywords that result in negative $IC(C1, k_i)_\delta$ scores (Bullet 4). This is done only at the first iteration when $C1$ has only title keywords. The intuition is that if content 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 $IC(C1, k_i)_\delta$ score, ‘canon hv20’ in this example, is greedily added to $C1$. The reasoning behind the pick is as follows. A keyword k_i occuring 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. For ex., if $C1$ has the keyword ‘speakers’, the 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 occuring 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 $C1$ (see 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 $C1$ (see Bullet 6). In our experiments we use the **first strategy** to have as many related, specific keywords for targeted content delivery.

Drawbacks of the Algorithm: 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. One way to tell if title keywords are relevant is to measure their association strengths with all content keywords. If all title-content clusters have low association strengths, 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 these 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. Best case scenario occurs when all keywords in $C2$ are off-topic or only one $C2$ keyword is contextually relevant. One iteration of the algorithm after computing MN association strengths suffices to partition relevant and noisy keywords. Worst case complexity is $O(\mathrm{MN}^2)$ 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. It is possible that multiple words resulting in similar Information Content change scores in the same iteration can be added to $C1$ to reduce the time complexity of the algorithm. This is an important focus of future investigations, especially given the wordier nature of blogs.

In the 220 crawled test posts from MySpace and Facebook, average size of $C1$ was 3 and that of $C2$ was 9. Average execution time of the cluster algorithm was 4.3ms per post.

[2] Due to lack of recent statistics, we use a conservative estmate of $\mathcal{N}$=70 billion calculated for AltaVista in 2003 [6]

3. EXPERIMENTS AND EVALUATION

The goal of our experiments 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 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. We recruited 36 graduate students and briefed them on the problem and experiment.

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, Ads_c, contained ads generated from the content as is. The second set, Ads_k, contained ads generated using keywords returned by our algorithm. Snapshots of ads for all posts were captured on a single day and stored offline (see sample at [2]). Each post had a maximum of 8 ads, 4 in each set. The 57 user posts were divided into ten sets, nine sets with six posts and one with three posts. Every set was evaluated by three randomly chosen users for a total of 30 evaluators used for the study.

Each user was shown a set of six posts one after another. Three users evaluated only three posts in the last set. For each post, users were also shown ads from the two sets, Ads_c and Ads_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 the ads they thought were **relevant to the post**. Instructions provided to the evaluators and a sample user response can be found at [2].

Results: Users responded by picking ads that they thought were **relevant** to the post. We aggregated reponses for the 57 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 atleast a 50% inter-evaluator agreement. Table 3 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, Ads_c and Ads_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 3 also shows these statistics. According to evaluator picks, processing content using our algorithm led to 22% more targeted unique ads.

For 54 of the 57 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 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. To summarize, for 54% 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.

Table 3: Targeted Content Delivery

Using content as is	
Number of ad impressions	144
Number and % of ads picked as relevant	43, 29.8%
Number and % of Unique ads picked as relevant	25, 17.36%
Using keywords returned by our algorithm	
Number of ad impressions	162
Number and % of ads picked as relevant	85, 52.47%
Number and % of Unique ads picked as relevant	64, 39.5%

4. DISCUSSION AND CONCLUSION

It is fairly well understood that user-generated content on Social Media has characteristics different from content we find elsewhere on the Web. What has not been extensively studied is how these characteristics affect content-analysis applications that work well on traditional media content. Here, we focussed on one particular characteristic of Social Media content - the prevelance of off-topic noise, and how it affects content delivery. The outcome of this work is useful for any application that needs to identify highly contextual keywords in content.

Using a simple heuristic of title keywords indicating the right context and the relationship between constrained contexts and word association strengths, we presented an intuitive way of partitioning a set of keywords into contextually relevant and off-topic ones. The algorithm is efficient, domain independent and easily adoptable. Preliminary user studies with posts from MySpace and Facebook and using Google AdSense clearly suggest the importance of eliminating off-topic noise and the efficacy of the algorithm in assisting targeted content delivery. A similar but large scale experiment using blogs and Zemanta is in the pipeline.

5. REFERENCES

[1] Simmetrics. sourceforge.net/projects/simmetrics/.

[2] Supplemental information. `http://knoesis.wright.edu/students/meena/WSDM2009/`.

[3] Yahoo term extraction service. `http://developer.yahoo.com/search/content/V1/termExtraction.html`.

[4] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th annual meeting on ACL*, 1989.

[5] D. Freitag. Information extraction from html: application of a general machine learning approach. In *AAAI '98/IAAI '98*, 1998.

[6] F. Keller and M. Lapata. Using the web to obtain frequencies for unseen bigrams. *Comput. Linguist. 2003.*

[7] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007.

[8] S. Soderland. Learning to extract text-based information from the world wide web. *Proceedings of Third International Conference on KDD*, 1997.

[9] P. Turney and C. Canada. Extraction of keyphrases from text: Evaluation of four algorithms. Technical report, National Research Council, Institute for Information Technology, 1997.

[10] Y. Yang, N. Bansal, W. Dakka, P. Ipeirotis, N. Koudas, and D. Papadias. Query by document. *Second ACM International Conference on Web Search and Data Mining*, 2009.