Automatic Advertising Campaign Development
Online advertising (1/2)

Online advertising is a form of promotion that uses the Internet and World Wide Web for the expressed purpose of delivering marketing messages to attract customers.

Benefits

- More targeted than traditional means - Better ROI
- Immediate publishing of information
- Good conversion tracking
- Purchase offline but in most cases research online first (ROPO)
Textual ads - Two main channels for distributing such ads:

- **Sponsored search (or paid search advertising)** places ads on the result pages of a Web search engine, where ads are selected to be relevant to the search query.
- **Content match (or contextual advertising)** places ads on third-party Web pages.

All major Web search engines (Google, Microsoft, Yahoo!) support sponsored ads and act simultaneously as a Web search engine and an ad engine.

Pricing Models: Pay-per-click (PPC), Pay per action (PPA), Pay-per-impression (PPI)
AdWords is Google's flagship advertising product and main source of revenue.

- Offers pay-per-click (PPC) advertising, and site-targeted advertising for text, banner, and rich-media ads.
- Campaigns: Local, national, and international distribution.
The problem

Products
- Advertiser has products
- Products have landing pages

Keywords, AdText
- Keywords needed for bidding, based on intuition
- Ad-text needed for showing ad

Adwords Management
- Campaigns need to be created and configured
- Keyword bidding needs to be optimized regularly
Our approach

Keywords and Ad Creatives
- Extract features of products from landing pages
- Find keywords that best describe products
- Generate Ad text based on patterns
- Suggest more keywords related to a given keyword

Campaign Creation and Optimization
- Use keywords and AdTexts to create campaign
- Monitor and test keyword performance keeping statistics
- Select best performing keywords to optimize profit
- Repeat optimization in regular time intervals
Keyword and Ad Creative Generation

Keyword Generation Goal

• Extract and suggest keywords
  • Non-obvious, which means that these words do not have a lot of advertiser competition ensuring that their price is low. Using a lot of these cheap keywords enables an advertiser to achieve the same or even better results as compared to using very popular and thus expensive keywords
  • Highly relevant

Propose ad creative

• For large campaigns, managing the copy of sponsored-search ads, their titles and descriptions presents a substantial editorial challenge if the task is going to be implemented manually
• Template – Characteristics of the product or service (name, price, location, type of product)
### Google AdWords Keyword Tool
- Advertiser can either enter a keyword phrase to find related keywords, or he can enter the website URL and Google will then identify useful keywords.
- Provides indicators for keyword phrases which are highly competitive and thus expensive.
- Search engines use **query-log** based mining tools to generate keyword suggestions.
- **Susceptibility to topic drift**: that is, the set often expands towards senses and phrases that have nothing to do with the target.

### AlchemyAPI
- Analyze content and extracting semantic meta-data: information about people, places, companies, topics, and languages.
- Few returned keywords – **Extraction only, not suggestion**

### Wordtracker
- KEI (Keyword Effectiveness Index) analysis feature - indicator of high-profit, low competition keywords.
- **Complaints: traffic estimates are unreliable**: They don’t correlate with Google or Yahoo! Traffic.

### AdGooroo
- Determines an advertiser’s **top competitors** and then actively searches for the keywords they are targeting.
- May return keywords which are likely to be **expensive**.
Keyword Suggestion: Research studies and systems (1/3)

- Corpus Dependent – Corpus Independent approaches
- Parallel corpus of bid phrases and aligned extracted keywords pairs (Ravi, S. et al., 2010)
  - Problem: Not many synonymous phrases inside a single landing page
  - 96% of the ads had at least one associated bid phrase not present in their landing page
  - Monolingual translation model capable of generating “unseen” phrases
- Each Yahoo! Ad
  - A landing page – Keywords extracted from the landing page
  - Advertiser-specified bid phrases
  - Example: film (extracted from landing page) → movie (bid phrase)
  - Rank based on probability \( \Pr(b|l) = \frac{\Pr(l|b)\Pr(b)}{\Pr(l)} \)
Nonobviousness – TermsNet (Joshi, A. & Motwani, R., 2006)

- Bidding on a large number of low-traffic keywords, the combined traffic from them adds up to the level produced by a popular keyword, at a fraction of the cost - More focused audience
- Characteristic Document: Build context of the term using terms found in the proximity of seed term in the top 50 hits from search engine for that term

Weights = strength of directed relationship, i.e., the frequency of destination term in characteristic document of source term
Dictionary is further expanded by adding terms that are similar to the ones contained in $D_0$ (initial dictionary). A search engine is queried for each word in the dictionary. The top $l$ documents are retrieved for each query and they are added to the corpus.

- Contextual similarity is established between different keywords in the dictionary.
- Traditional document measures cannot be applied to terms as they are too short.
- Semantic Similarity $K(x, y) = QE(x). QE(y)$
- One-word term

Semantic Similarity between Terms – Wordy
(Abhishek, V. & Hosanagar, K., 2007)
Keyword Extraction Module (1/2)

- Pre-processing step
  - HTML content of each landing page is parsed, stop words are removed and the content is tokenized
  - Jericho HTML Parser: java library allowing analysis and manipulation of parts of an HTML document

- For each word $lj$ in the tokenized output, we compute a weight associated with the word for each occurrence inside a specific tag
  - $w_{j\text{tag}} = weight_{\text{tag}} \times f_{j\text{tag}}$
    - weight_{\text{tag}} is a special weight assigned to each different kind of HTML tags and $f_{j\text{tag}}$ is the frequency of the word inside the specified tag
    - Important tags: <title>, meta keywords, meta description, anchor text, <h1>, <b>
  - Compute the special weight of each word as the sum of all $w_{j\text{tag}}$ weights for this word
Keyword Extraction Module (2/2)

- Relevance score of each word
  - \( relevance\_score_j = \frac{special\_weight_j}{MAX\_WEIGHT} \)
    - MAX_WEIGHT represents the maximum special weight that a word could have inside the HTML document
- Unimportant words occurring on the page can be filtered out using a threshold on the relevance score
- Construct word co-occurrence matrix
  - top N relevant words
  - Co-occurrence: if word_i and word_j appear in the same unit, then they co-occur once, and freq_{i,j} should be added one
For each given seed keyword (extracted from previous step)

- keyword is entered as a query into a search engine API (Google JSON/Atom Custom Search API) – example: “car rental”
- API returns a set of short text snippets relevant to the query
- The top 50 results are downloaded and loaded in Apache Lucene Library

With Google Search API we can retrieve from each result: Title, snippet, pagemap, abridged version of this search result’s URL, e.g. www.example.com

For each word find the relevance score – same approach as before
  - Importance: Title
Keyword Suggestion Module (2/2)

- Faster than retrieving and crawling actual documents
- Trends
- Keep the same importance for last results – we don’t want to miss “unseen” words
- APIs: one key - limited queries per day
- Web Scraping: Legal issues for systems
- Optional: Propose through the Google AdWords API and TargetingIdeaService for popular keywords
- Length of phrases: 2-3 words in most cases
How can we evaluate at a first glance the results?

Ratings

- Relevance
  - Indicates Relevance of suggested keyword to seed word
  - Given by human evaluators
  - e.g.: For query ‘flights’
    - Relevance (‘flights’, ‘cathay pacific’) = 1
    - Relevance (‘flights’, ‘cheap flight’) = 1
    - Relevance (‘flights’, ‘magazines’) = 0

- Nonobviousness
  - Indicates nonobviousness of suggested keyword relative to seed word
  - Calculated as: If no base query word/stem present in suggested keyword, Nonobviousness = 1, else = 0
  - e.g.: For query ‘flights’
    - Relevance (‘flights’, ‘cathay pacific’) = 1
    - Relevance (‘flights’, ‘cheap flight’) = 0
    - Relevance (‘flights’, ‘magazines’) = 1
  - Porter stemmer for automating this rating
Evaluation Measures

Average Precision:
- Ratio of number of relevant keywords retrieved to number of keywords retrieved.
- Indicates quality of results

Average Recall
- The proportion of relevant keywords that are retrieved, out of all relevant keywords available.
- Total number of relevant keywords unknown. Approximate:
  \[ \text{Recall} (T_i) = \frac{\# \text{ retrieved by } T_i}{\# \text{ retrieved by } (T_1 \cup T_2 \cup \ldots \cup T_n)} \]

Average Nonobviousness
- The proportion of nonobvious words, out of retrieved relevant words
Experiment with 3 approaches for generating the sentences that will describe our advertisement

1. Advertiser gives as input an indicative phrase or sentence for the promoted product. Paraphrase the sentence to generate candidate ad texts

2. Use automatic text summarization techniques for summarizing the content of the given landing page and then generate paraphrases from the resulted sentences

3. Filter generated sentences with the keywords that we are bidding on. Keep only the most relevant

Sentence compression - AdWords limitations

Perform sentiment analysis - Ad text must be attractive and positive for a user to click
Feature Extraction: Further Challenges

- Entity recognition and webpage classification
- Other features that can be extracted for helping the automated creation of the campaign
  - Topic Categorization - Assign the most likely topic category (gaming, mobile phones, movies, etc.)
  - Language Detection – Useful for identifying regions for the campaign
  - Structured Content Scraping - Product attributes, descriptions, pricing, etc.
Campaign Creation and Budget Optimization

- Use keywords and AdTexts to create campaign
- Monitor and test keyword performance keeping statistics
- Select best performing keywords to optimize profit
- Repeat optimization in regular time intervals
Components

- Adwords Adapter
- Campaign Creation
- Database and Statistics
- Task Scheduler
- Budget Optimization
- Performance Prediction
Component diagram

Adwords API

Adwords Adapter

Campaign Creation

Task Scheduler

Budget Optimization

Statistics

Performance Prediction

Database
Task Scheduler

- The system needs “memory”
- When to collect statistics
- When to optimize budget
- When to test for new keywords

- Tasks are executed in date and time priority
- Tasks are stored in file
- Sleeps when idle
Keyword Statistics

- Number of Impressions
- Number of Clicks
- Number of Conversions
- Click-through rate = Clicks / Impressions
- Conversion rate = Conversions / Clicks
Budget Optimization

- Goal: maximize profit
- which $k \in K$ keywords and what bid $b \in \mathbb{R}$ to select?

Goal: $\text{Max } ( B + \sum v(k,b) )$

Limitation: $\sum w(k,b) \leq B$

- $w(k,b) = \text{AvgCPC}(k,b) \times \text{Clicks}(k,b)$
- $v(k,b) = \text{SaleProfit}(k) \times \text{Conversions}(k,b) - w(k,b)$
Knapsack and Genetic algorithm

- Multi-choice Knapsack problem (NP-hard)
- Genetic algorithm solves it
- Finds an approximately optimum solution

- A possible solution is modeled as a chromosome:

<table>
<thead>
<tr>
<th>k1</th>
<th>k2</th>
<th>k3</th>
<th>...</th>
<th>kN</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.60</td>
<td>$0.00</td>
<td>$0.45</td>
<td>...</td>
<td>$0.50</td>
</tr>
</tbody>
</table>

- Fitness Function \( = B + \Sigma v(k,b) \)
Genetic algorithms basics

- **Population** is a set of chromosomes
- The population evolves in each **generation**
- **Selection** is the process of finding the fittest chromosomes to become the parents of the next generation.

[Diagram showing weighted Roulette Wheel Selection with labels 10, 20, 30, 40]
The process of combining two chromosomes is called **crossover**.

| Parent 1 | [1 0 4 | 0 2 1 0] | produce | Offspring 1 | [1 0 4 3 0 0 1] |
|----------|----------------|---------|-------------|----------------|
| Parent 2 | [0 2 2 | 3 0 0 1] |         | Offspring 2  | [0 2 2 0 2 1 0] |

**Mutation** is used to escape local optimums.

**With elitism** the fittest chromosomes pass to the next generation directly.
We want to predict **Impressions**

Impressions affect Clicks and Conversions

<table>
<thead>
<tr>
<th></th>
<th>MaxCPC</th>
<th>Competition</th>
<th>ELMS</th>
<th>Impressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>X1,1</td>
<td>X2,1</td>
<td>X3,1</td>
<td>Y1</td>
</tr>
<tr>
<td>2.</td>
<td>X1,2</td>
<td>X2,2</td>
<td>X3,2</td>
<td>Y2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>n.</td>
<td>X1,n</td>
<td>X2,n</td>
<td>X3,n</td>
<td>Yn</td>
</tr>
<tr>
<td>n+1.</td>
<td>X1,n+1</td>
<td>X2,n+1</td>
<td>X3,n+1</td>
<td>Yn+1 = ?</td>
</tr>
</tbody>
</table>

Multivariate regression

\[ Y = \beta_0 + \beta_1 \times X_1 + \beta_2 \times X_2 + \beta_3 \times X_3 \]


Yunhong Zhou, Victor Naroditskiy. Algorithm for the Stochastic Multiple-Choice Knapsack Problem and Application to Keywords Bidding, WWW 2008, April 21-25, 2008, Beijing, China.
