

# Automatic Advertising Campaign Development

Matina Thomaidou, Kyriakos Liakopoulos, Michalis  
Vazirgiannis

Athens University of Economics and Business

April, 2011

# Outline

- 1 Online Advertising
- 2 Keyword and Ad Creative Generation
- 3 Campaign Creation and Budget Optimization
- 4 Experiments
- 5 Conclusions and Future Work

# Online Advertising

- *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)

# Online Advertising

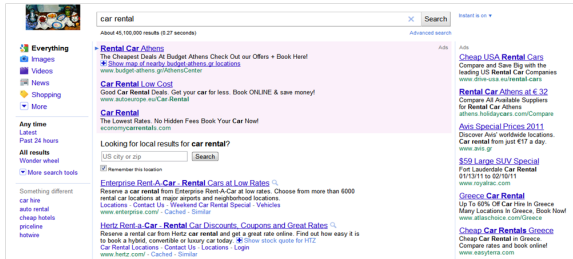
- Textual ads - Two main channels for distributing such ads:
  - ① Sponsored search (or paid search advertising) places ads on the result pages of a Web search engine, where ads are selected to be relevant to the search query
  - ② Content match (or contextual advertising) places ads on third-party Web 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)

# Terminology

- **Keyword:** A word or phrase that matches a web-users search query and at the same time describes the content advertised
- **Ad-Creative:** The text that a web-user reads on an advertisement
- **Impression:** The appearance of an advertisement in a SERP after a web-users query
- **Click:** The action of a web-user clicking on an advertisement
- **Conversion:** Action (e.g. purchase, registration) after arriving to a website
- **Campaign:** Set of components and preferences for the advertising purpose
- **Ad Group:** Set of related ads, keywords, and placements within a campaign

# Google AdWords

- Google 2010 revenues: \$30 billion, 97% come from ads
- Adwords is Google's flagship advertising product



- Offers pay-per-click (PPC) advertising for text, banner, and rich-media ads

# The problem

## Products

- Advertiser has products to be advertised
- Products have landing pages

## Keywords, Ad Text

- 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 keywords to optimize profit or traffic
- Automatically repeat optimization in regular time intervals



# Automatic Advertising Campaign Development Architecture

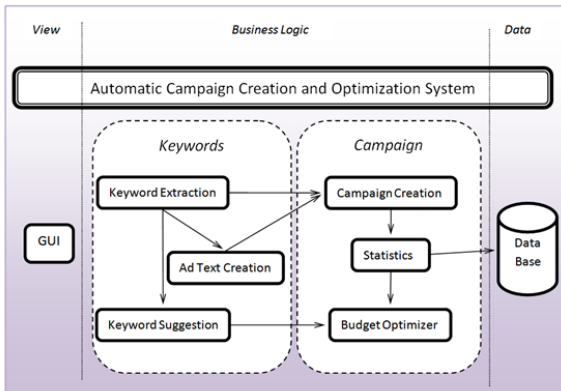


Figure: Automatic Campaign Creation and Optimization System - Parts

# Keyword Extraction Module

Table: Tag Weights

Element	Assigned Weight
<title>	50
meta keywords	40
meta description	40
anchor text	30
<h1>	30
<b>	10
other	1

- Java using JDK SE 6 on the Eclipse IDE
- Jericho HTML Parser: java library allowing analysis and manipulation of parts of an HTML document. Eclipse Public License (EPL) and GNU Lesser General Public License (LGPL).
- Stopwords:
  - KEA data file for english stopwords
  - Lucene GreekAnalyzer.java for greek stowords

# Keyword Suggestion Module

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 30 results are downloaded and loaded in **Apache Lucene Library**

[Hertz Rent-a-Car - Rental Car Discounts, Coupons and Great Rates](#)  
Reserve a rental car from Hertz car rental and get a great rate online. Find out how easy it is to book a hybrid, convertible or luxury car today. [Show stock quote for HTZ](#)  
[Car Rental Locations - Contact Us](#) - [Locations](#) - [Login](#)  
[www.hertz.com/](#) - [Cached](#) - [Similar](#)

Figure: Snippet

# Suggestions

## Data

- Response data in Atom format
- Retrieve feed/entry/summary/text() string type property indicating the snippet of the search result and feed/entry/title/text() string type property indicating the title of the search result
- Each set of title and snippet results that were retrieved after a seed query represents a document for Lucene
- Find new unique distinct words
- Co-occurrence find phrases
- Lucene scoring: Similarity class

## Apache Lucene

- Cross-Platform Solution
- Available as Open Source software under the Apache License which lets you use Lucene in both commercial and Open Source programs
- 100% pure Java
- Implementations in other programming languages available that are index-compatible

# Scoring

$$w_{i,j} = tf * idf = \frac{freq_{i,j}}{Max_l freq_{l,j}} * \log \left( \frac{N}{n_i} \right) \quad (1)$$

$$w_{jtag} = weight_{tag} * f_{jtag} \quad (2)$$

$$special\_weight_j = \sum w_{jtag} \quad (3)$$

$$relevance\_score_j = \frac{special\_weight_j}{MAX\_WEIGHT} \quad (4)$$

$$boosted\_score_j = relevance\_score_j * k^{noOfWords} \quad (5)$$

# Ad Creative Module

- Extract all the text from the HTML document
- Summarize to keep the most important meaning for the description of our advertising page
- Classifier4J: Java library. Text summary facility
- Divide the sentence into the two description lines
- Cut exceeded words

```
graph TD; API([Adwords API]) <--> Adapter[Adwords Adapter]; Adapter <--> CC[Campaign Creation]; Adapter <--> TS[Task Scheduler]; TS <--> BO[Budget Optimization]; TS <--> Stats[Statistics]; Stats <--> DB[(Database)]; Stats --> PP[Performance Prediction]; PP --> BO;
```

The diagram illustrates the Adwords API architecture. It features a central column of components: Adwords Adapter, Task Scheduler, and Statistics. The Adwords API (grey oval) connects to the Adwords Adapter. Campaign Creation connects to the Adwords Adapter. The Adwords Adapter connects to the Task Scheduler. The Task Scheduler connects to Budget Optimization and Statistics. Statistics connects to the Database and Performance Prediction. Performance Prediction connects to Budget Optimization. A dashed orange line separates the Adwords API from the rest of the system.






European Union  
European Social Fund

EFMD  
EQUIS  
ACCREDITED

ASSOCIATION TO  
IMPROVE QUALITY IN  
HIGHER EDUCATION  
ACCREDITED

NSRF  
2007-2013  
OPERATIONAL PROGRAM  
EDUCATION AND  
SOCIAL POLICY

Co-Financed by Greece and the European Union

# Configuration

- Java using JDK SE 6 on the Eclipse IDE
- MySQL 5.5 Community Server
- Adwords API Java Library (adwords-api-8.2.0.jar)
- MySQL JDBC Connector  
(mysql-connector-java-5.1.10-bin.jar)
- Simple Adwords account (client)
- Google Analytics account (monitor conversions)
- My Client Center (MCC account) for Adwords API



# Task Scheduler

- The system needs "memory"
- Tasks
  - 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
- System "sleeps" until the next task

# Keyword Statistics

Uses database to get statistics

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

Result: Find keyword options (k,b) and their statistics

# Database Tables

Table: Database tables and attributes - I

AdGroup	Relevance
adgroupId-INTEGER	adgroupId-INTEGER
landingPageUrl-VARCHAR(2000)	keywordText-VARCHAR(60)
profitPerSale-DOUBLE	relevance-DOUBLE
	Tested-BOOLEAN

Table: Database tables and attributes - II

Keyword	Statistics
adgroupId-INTEGER	statisticsId-INTEGER
keywordId-INTEGER	adgroupId-INTEGER
text-VARCHAR(60)	keywordId-INTEGER
relevance-DOUBLE	maxCpc-DOUBLE
matchType-VARCHAR(7)	startDate-DATE
	daysDuration-MEDIUMINT
	impressions-INTEGER
	conversions-INTEGER
	averageCpc-DOUBLE
	averageCpm-DOUBLE
	averagePosition-DOUBLE

# Budget Optimization

Limited Budget  $B$ . Available keywords  $K$ .

Goal: maximize profit or traffic.

$$\sum_{k \in K} w_k(k, b_k) \leq B$$

where  $w_k$  is the cost of bidding price  $b$  on keyword  $k$  (weight) that produce:

$$\max \sum_{k \in K} v_k(k, b)$$

where  $v_k$  is the expected profit of bidding price  $b$  on keyword  $k$  (value).

# Knapsack and Genetic algorithm

- Multiple-choice Knapsack problem (NP-hard)
- Genetic algorithm solves it
- Finds an approximately optimum solution
- A possible solution is modeled as a chromosome

k1	k2	k3	...	kN
\$0.60	\$0.00	\$0.45	...	\$0.50

Fitness Function:

$$\sum_{k \in K} v_k(k, b)$$

# Genetic Algorithm 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.

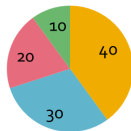


Figure: Weighted Roulette Wheel Selection

- The process of combining two chromosomes is called **crossover**.
- **Mutation** is used to escape local optimums
- With **elitism** the fittest chromosomes pass to the next generation directly

# Configuration

- JGAP Library (jgap.jar)
  - The Java Genetic Algorithms Package library is an open source framework for implementing and configuring genetic algorithms and genetic programming. It is published under the GNU Lesser Public License for non commercial use.
- Michael Thomas Flanagan's Java Scientific Library (flanagan.jar)
  - This library implements many mathematical areas, one of them is regression. It also provides methods for plotting the data. Permission to use for non-commercial purposes is granted.

# Performance Prediction

- We want to predict Impressions
- Impressions affect Clicks and Conversions

	Clicks(X <sub>1</sub> )	GMS (X <sub>2</sub> )	Competition (X <sub>3</sub> )	Impressions (Y)
1.	X <sub>1,1</sub>	X <sub>2,1</sub>	X <sub>3,1</sub>	Y <sub>1</sub>
2.	X <sub>1,2</sub>	X <sub>2,2</sub>	X <sub>3,2</sub>	Y <sub>2</sub>
...	...	...	...	...
n.	X <sub>1,n</sub>	X <sub>2,n</sub>	X <sub>3,n</sub>	Y <sub>n</sub>
i: 1..n	X <sub>1,i</sub>	X <sub>2,i</sub>	X <sub>3,i</sub>	Y <sub>i</sub> = ?

- Multivariate regression

$$Y = \theta_0 + \theta_1 * X_1 + \theta_2 * X_2 + \theta_3 * X_3.$$



# Experiments

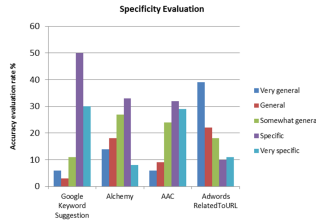
## AdWords API Sandbox

- Testing and development environment that replicates the functionality of the AdWords API web services.
- Develop and test applications without spending API units (calls made in the sandbox are free)

## Budget Optimization Testing - 4 optimization scenarios

- Budget Optimization for Profit with No Prediction (NoPredProfit)
- Budget Optimization for Traffic with No Prediction (NoPredTraffic)
- Budget Optimization for Profit With Prediction (PredProfit)
- Budget Optimization for Traffic With Prediction (PredTraffic)

During consecutive weeks of March 2011, we run an automatic campaign...



Budget = 50	<u>Clicks</u>	<u>Cost</u>	<u>Profit</u>	<u>Counter</u>	<u>AverageBid</u>
NoPredProfit:	60.451	49.944	219.51	24	1.495
NoPredTraffic:	61.5	49.938	206.22	23	1.436
PredProfit:	82.363	49.901	317.18	16	1.379
PredTraffic:	86.51	49.886	274.82	18	1.429

Budget = 200	<u>Clicks</u>	<u>Cost</u>	<u>Profit</u>	<u>Counter</u>	<u>AverageBid</u>
NoPredProfit:	197.83	199.88	621.32	56	1.552
NoPredTraffic:	200.84	199.91	582.22	54	1.506
PredProfit:	236.94	199.87	787.63	31	1.423
PredTraffic:	248.6	199.86	638.13	32	1.438

Budget = 600	<u>Clicks</u>	<u>Cost</u>	<u>Profit</u>	<u>Counter</u>	<u>AverageBid</u>
NoPredProfit:	333.55	389.6	798.9	97	1.618
NoPredTraffic:	343.99	405.17	795.29	107	1.631
PredProfit:	607.74	599.85	1645.6	70	1.569
PredTraffic:	622.69	599.82	1569.2	68	1.523

# Contributions

- Fully implemented and functional system with a comprehensive experimental evaluation in real world data
- Showcase that the proposed architecture and methodology is plausible and feasible and provide proof of concept
- Good basis for a larger system that has more features and can provide more functionality useful for the field of web-search advertising
- Find and implement tangible solutions to the issues arising at the very core of a Campaign creation and Budget optimization software

## Selected Bibliography



Abhishek, V., Hosanagar, K., 2007.

Keyword generation for search engine advertising using semantic similarity between terms.  
*In Proceedings of the ninth international conference on Electronic commerce.*



Joshi, A., Motwani, R., 2006.

Keyword Generation for Search Engine Advertising.  
*Sixth IEEE International Conference on Data Mining - Workshops (ICDMW06).*



Ravi, S. et al., 2010.

Automatic generation of bid phrases for online advertising.  
*Proceedings of the third ACM international conference on Web search and data mining.*



Yunhong Zhou, Victor Naroditskiy., 2008.

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



Chu P. and Beasley JE, 1998.

A genetic algorithm for the multidimensional knapsack Problem.  
*J Heuristics 4.*



Farhad Djannaty, Saber Doostdar, 2008.

A Hybrid Genetic Algorithm for the Multidimensional Knapsack Problem  
*Int. J. Contemp. Math. Sciences.*

# Thank you!

email: [thomaidous@aueb.gr](mailto:thomaidous@aueb.gr)

webpage: <http://www.db-net.aueb.gr/matina>