

An Open Source Ad Server

Submitted to:

Dr. Chris Pollett

By

Pushkar Umaranikar

Fall 2014

**Department of Computer Science
San Jose State University**

Table of Contents :

Introduction..... 4

Overview of Deliverables 6

 Deliverable 1: Study of Google AdSense and Google AdWords 6

 Deliverable 2: Implementation of configuring advertisement server in Yioop. 9

 Deliverable 3: Designing online auction system..... 11

 Deliverable 4: Prototype - show relevant ads based on search query 13

Conclusion 16

References..... 17

List of Figures:

Figure 1 : Creating your Ad Campaign.....	6
Figure 2 : Creating your Ad Campaign.....	7
Figure 3 : Auction methods.....	8
Figure 4 : Ad Server configuration in Yioop	9
Figure 5 : Ad spaces on Yioop's search results page.....	10
Figure 6 : Architecture of online auctioning system.....	11
Figure 7 : Class diagram – Online auctioning system.....	12
Figure 8 : List of documents used for indexing	13
Figure 9 : Output of cosine similarity algorithm used to find relevant documents for user's query	14
Figure 10 : Document set for query "cloud"	14
Figure 11 : relevant ads for the entered search query "cloud"	15
Figure 12 : Relevant docs for given search query "android"	15
Figure 13 : relevant ads for query "android"	15

Introduction

The aim of the project is to extend search engine capabilities with an open source advertising platform. Yioop is a GPLv3, open source, PHP search engine developed by Dr. Pollett which can be integrated with open source advertisement server to show the content based advertisements on search query's results page.

Display advertisements are a prominent revenue source for most modern search engines such as Google, Yahoo, and Bing. These advertisements constitute over 80% to 90 % of total revenue. Sophisticated algorithms and machine learning techniques are used to compute the relevance of these ads to a given search query.

Online auctioning systems are used to price advertisements. They generally use pay per click model. These auctioning systems allow advertisers to target precisely and exactly their most likely buyers. That means, if advertisers use this auctioning system correctly, they waste no money advertising to people who have no interest in what you are selling.

Search engines retrieve ads from an auction system based on user input and they charge only when somebody clicks on it.

People owning websites can display advertisements on their website with the help of an advertisement server. They can request advertisement system to show ads on their websites based on content of the website. If a user clicks out of these advertisements on the website, the advertisement system shares revenue with the website owner as visitor clicked ad displayed on his website. Hence, it allows website owner to make some revenue out of advertisements on website.

For this project, we are implementing an open source advertisement server along with auctioning system where advertisers can put their ads. These advertisements will be display on search results page or on websites based on their relevancy with the user's query or contents on the website.

This report includes details of all deliverables of the CS297 project. Before starting the project, an initial activity was to install a Yioop on the local server. Then we moved on with writing

CS297 proposal. The first deliverable was to study Google AdSense. The service is designed for website publishers who want to display targeted advertisements on website pages. We also studied Google's advertising system known as Google AdWords in which advertisers bid on certain keywords in order for their clickable ads to appear in Google's search results.

The second deliverable was to implement configuration settings for advertisement server with Yioop to enable advertisements on Yioop's search results page. We used AdSpeed, an existing open source ad server for placing advertisement on search results page. The third deliverable was to design the online auctioning system along with advertising placement service. We have come up with high level architecture diagram along with detailed class diagram. In addition, I studied Amazon review system used for rating a product. In this report, each deliverable is explained under appropriate section headers.

The forth deliverable was to work on a prototype for retrieving most relevant ads based on a user's query from advertisement database.

Overview of Deliverables

Deliverable 1: Study of Google AdSense and Google AdWords

The purpose of this deliverable was to understand the workings of Google AdSense and Google AdWords and to prepare a presentation on how Google sells Ad Spaces.

Google AdWords is an online advertising service that places advertising copy at the ad spaces available on Google's search result page. The choice and placement of the ads is based in part on a proprietary determination of the relevance of the search query to the advertising copy. AdWords offers pay-per-click (PPC), that is, cost-per-click (CPC) advertising, cost-per-thousand-impressions (cost per mille or CPM) advertising, site-targeted advertising for text, banner, and rich-media ads, and remarketing. The AdWords program includes local, national, and international distribution. Google's text advertisements are short, consisting of one headline of 25 characters, two additional text lines of 35 characters each, and a display URL of 35 characters. Image ads can be one of several different Interactive Advertising Bureau (IAB) standard sizes.

Your first campaign

A **campaign** focuses on a theme or a group of products. To create a campaign, you'll set a budget, choose your audience, and write your ad. Keep in mind, you won't be charged for selecting options, and you can always make changes later.

1. Decide how much to spend

Your budget	Rs. 1.00 per day	←	Select maximum amount for Ad campaign	✎
-------------	------------------	---	---------------------------------------	---

2. Choose a target audience

Locations	United States and Canada	←	Region where you want to display your advertisement	✎
-----------	--------------------------	---	-----------------------------------------------------	---

Networks	Search Network, Display Network	←	Sites that shows ad along with other content	✎
----------	---------------------------------	---	----------------------------------------------	---

Figure 1 : Creating your Ad Campaign

Keywords search php open source ← Keywords associated with advertisement

3. Set your bid

Bid Automatically set my bids to get the most clicks within my budget

4. Write your ad

Text ad

YoopSearch
yooptsearch.com
PHP Search Engine
Open source Search Engine

← Actual text Ad with link to lading page and text associated with it.

Figure 2 : Creating your Ad Campaign

AdWords allows us to take the advantage of the millions of searches on Google where advertisers can create ads and choose when they want them to appear above to next to relevant search result. While creating advertisement, an advertiser enter words related to their product and then AdWords displays that ad when someone searches for that or related word. AdWords also allows advertisers to choose area to display advertisement if they advertiser want to target local area.

In addition to online business growth, AdWords allows us to manage and control advertising spends. With AdWords, one can select maximum amount that he willing to pay.

Google AdSense is a program run by Google that allows publishers to display ads from auctioning systems like AdWords. These advertisements are administered, sorted, and maintained by Google, and they can generate revenue on either a per-click or per-impression basis.

AdSense's innovative targeting options allow advertisers to more precisely reach their targeted audiences on your site which includes contextual targeting, placement targeting and internet

based targeting. AdSense has a variety of ad formats including mobile web pages, videos, online games, TV inventory, parked domains, feeds and site search results.

Ads displayed by Google AdSense contain an advertising icon at a corner. This is known as AdChoices. It is not an online advertising provider but a set of principles a consortium of online advertisers have agreed to. It is used for interest based advertising. The icon indicates information about user's interest is being gathered to improve ads displayed for that user. It helps to show ads based on individual's interest.

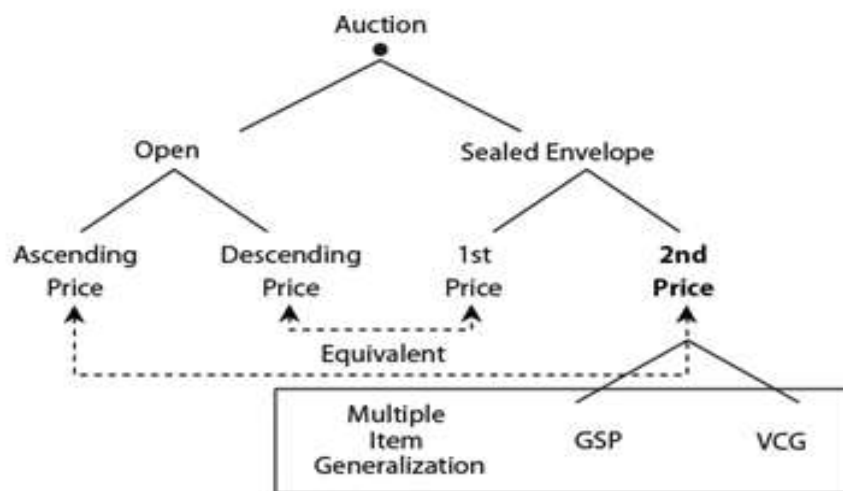


Figure 3 : Auction methods

The question may arise in a publisher mind where will my ad appear on a search result page. Google uses ad space allocation strategy for that. Allocation of ad spaces happens according to some auction rules. In most of the auction rules, highest bidder gets most valuable ad space. Auction methods are mainly divided into two, which are open and sealed envelope auctions. Open auctions are public venue auctions where sealed envelope auctions are private enclosure of bids. Vickrey-Clarke-Groves (VCG) is used to ensure truthful bidding. It maximizes total valuation of the whole system. In sealed envelope, auctioneer decides the allocation and charge for each item.

Deliverable 2: Implementation of configuring advertisement server in Yioop.

The purpose of this deliverable was to add and implement configuration options for ad space allocation and ad script in Yioop to enable content base advertisement placing service in Yioop.

The configuration settings of my deliverable allow Yioop administrator to select ad space. An administrator can also add adscripts which will execute on the search results page and render most relevant ads based on user's query retrieved from ads database. In Yioop, these options are available to administrator under server settings.

Ad Server Configuration

Ad Alignment
 Top Side Both None ← Option to select ad location on search results page.

Global Element Ad Script

Top Element Ad Script

```
<!-- AdSpeed.com Serving Code 7.9.6 for [Ad] Sample Image Ad 468x60 -->
<script type="text/javascript">var asdate=new Date();var q='&tz='+asdate.getTimezoneOffset()/60 +'&ck='+
(navigator.cookieEnabled?'Y':'N') +'&jv'+(navigator.javaEnabled()?'Y':'N')
+'&scr='+screen.width+'x'+screen.height+'x'+screen.colorDepth +'&z'+Math.random()
+'&ref='+escape(document.referrer.substr(0,255))
+'&uri='+escape(document.URL.substr(0,255));document.write('<iframe width="468" height="60"
```

↑ Ad script to retrieve content based ads from ads database

Figure 4 : Ad Server configuration in Yioop

Based on the location selected in configuration, ads relevant to a user's query are displayed on search results page. If "Top" alignment is selected then ads would be available at top banner. If "Side" alignment is selected then ads would be available at side banner. To display ads at top as well as at side banner, administrator must select "Both" as ad alignment option. To disable ads from Yioop, "None" option is provided.

To test the implementation, I used AdSpeed which is an open source high performance ad server for advertisers and publishers. User can login as an advertiser to adSpeed and create your advertisement. Then we have to associate newly created ad with available zones. Then ad speed would generate serving code for the ad which can be added to under Yioop's ad server configuration to test its working. The size of top banner ad on adSpeed is 468 x 60.

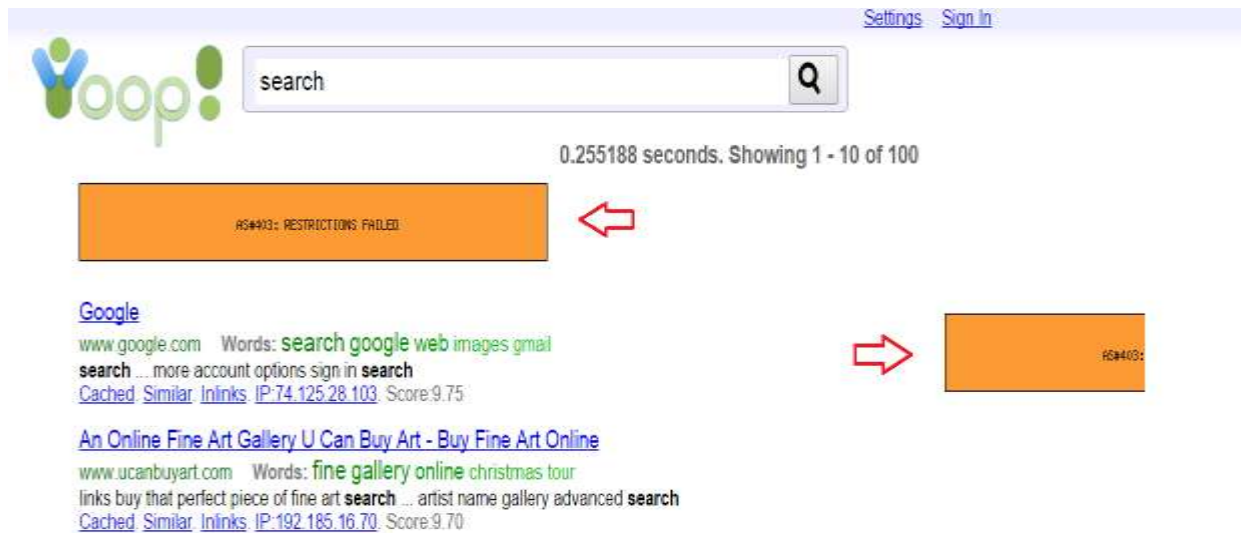


Figure 5 : Ad spaces on Yioop's search results page

Figure 5 shows sample ads displayed on Yioop's search results page. In this case "Both" option selected from ad alignment. Hence, we can see sample ad displayed at top as well as side banner. Ad retrieved from the ad script added in the configuration settings under top adscript and side adscript. This is a test adscript generated from advertisement created on AdSpeed open source server.

Deliverable 3: Designing online auction system

The aim of this deliverable was to design online auction system where advertisers can create campaigns for their products. Another part of the deliverable is to present on a topic “When Can I trust an average rating on Amazon.” Advertisers will provide information about keywords association, bid amount, and description of the advertisement along with actual campaign. This information would help in retrieving content based advertisement based on user’s query.

The auctioning system is nothing but an ad server where user create ad with targeting criteria. The server will handle ad serving, ad tracking, and real-time reporting on ad impressions, clicks, and revenue. An ad server will manage ad inventory with user-friendly interface.

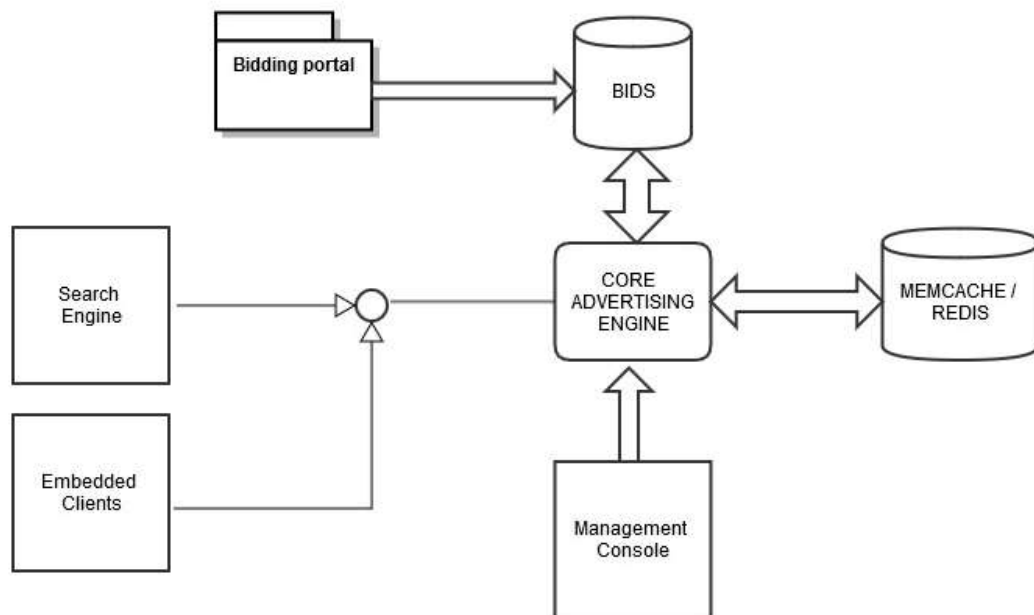


Figure 6 : Architecture of online auctioning system

Figure 6 shows high level architecture of online auctioning system. A user can create campaign using web based bidding portal. Ads created by user will be added to BIDS database. When a user searches for a particular query, the core advertising engine processes ads from database based on certain parameters like keywords, ad description. An advertising engine then would respond with list of most relevant ads from existing database. Sophisticated algorithms are used

to find most relevant ads. Traditional database we can have in memory database like Redis to minimize response time. The system will also have management console which helps to change the rules of bidding, and relevancy at run time.

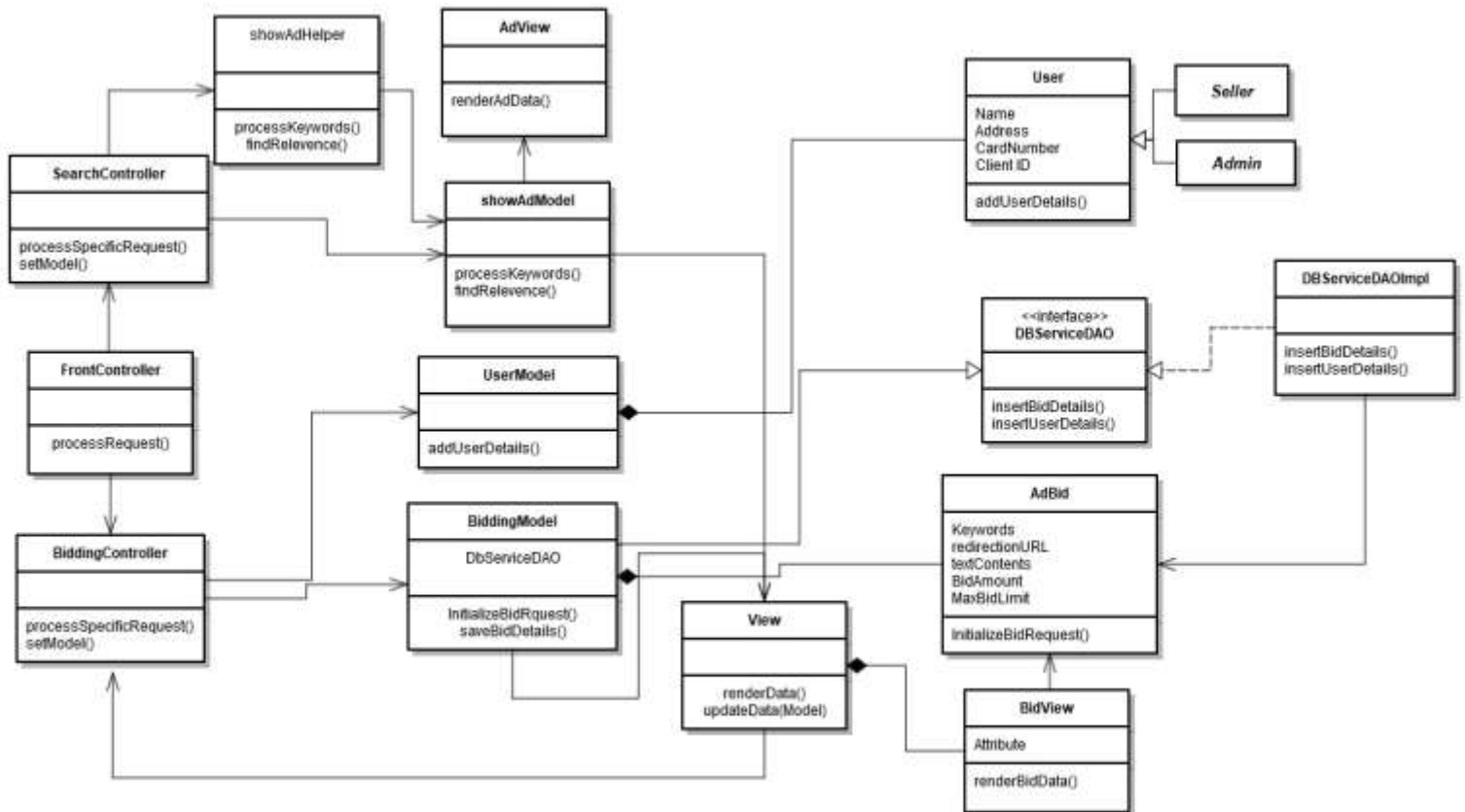


Figure 7 : Class diagram – Online auctioning system

Figure 7 shows class diagram of an online auctioning system. An auctioning system is designed to work with Yioop. The front controller accepts user request and if it's an auction request, it forwards request to BiddingController. A controller invokes model based on request. It acts as a listener both to model and view to receive events. It is responsible for keeping model and view in sync. Suppose, BiddingController receives change in event from BidView, then it routes appropriate changes to BiddingModel. The advantage of it is that the view doesn't need to know about the model. DBServiceDAO responsible for handling database request and respond with appropriate list of results.

Deliverable 4: Prototype - show relevant ads based on search query

The aim of this deliverable was to design and develop a model which would retrieve relevant ads from advertisement database based on user's query. To implement the system, we first created bag of words model based on documents available for indexing. The bag of words model used to retrieve documents relevant to the search query using cosine similarity ranking algorithm.

Once we have relevant documents, get all keywords from retrieved documents and build a document vector based on keywords obtained. Relevant ads from advertisement database are obtained from document vector and ad vector considering each ad as a different document by cosine similarity ranking algorithm. We can measure closeness between the two vectors by taking their dot product.

When $\Theta = 0^0$, $\cos \Theta = 1$ and two vectors are collinear and if $\Theta = 90^0$, $\cos \Theta = 0$ and two vectors are orthogonal. Consider we have a query vector \vec{q} and document vector \vec{d} , and similarity is defined as the cosine of the angle between them.

$$\text{sim}(\vec{d}, \vec{q}) = \frac{\vec{d} \cdot \vec{q}}{|\vec{d}| \cdot |\vec{q}|}$$

Consider list of 10 documents. First, we will find relevant documents based on user's query.

```

https://pushkarumaranikar.wordpress.com/
http://en.wikipedia.org/wiki/Android_(operating_system)
http://en.wikipedia.org/wiki/Online_and_offline
http://software.canon-europe.com/
http://www.canonrumors.com/
http://en.wikipedia.org/wiki/Database
http://en.wikipedia.org/wiki/Cloud_computing
http://www.microsoft.com/enterprise/it-trends/cloud-computing/default.aspx#fbid=DGpI5SyUMGd
http://www.interoute.com/what-are-cloud-servers
http://www.rackspace.com/cloud/servers

```

Figure 8 : List of documents used for indexing

For a search query “cloud”, the relevant documents from lists are 7, 8,9,10 which are relevant documents for a given user's query using cosine similarity ranking algorithm.

```

Enter search query:
cloud
Document vector created for document :0
Document vector created for document :1
Document vector created for document :2
Document vector created for document :3
Document vector created for document :4
Document vector created for document :5
Document vector created for document :6
Document vector created for document :7
Document vector created for document :8
Document vector created for document :9
cosine similarity for given query with document 0 :0.0
cosine similarity for given query with document 1 :0.004937291752307908
cosine similarity for given query with document 2 :0.0
cosine similarity for given query with document 3 :0.0
cosine similarity for given query with document 4 :0.0
cosine similarity for given query with document 5 :0.003579073207718753
cosine similarity for given query with document 6 :0.5771441342195869
cosine similarity for given query with document 7 :0.19466739886057416
cosine similarity for given query with document 8 :0.43019880494981483
cosine similarity for given query with document 9 :0.0

http://en.wikipedia.org/wiki/Cloud_computing => 0.5771441342195869
http://www.interoute.com/what-are-cloud-servers => 0.43019880494981483
http://www.microsoft.com/enterprise/it-trends/cloud-computing/default.aspx#fbid=DGpI5SyUMGd => 0.19466739886057416
http://en.wikipedia.org/wiki/Android_(operating_system) => 0.004937291752307908
http://en.wikipedia.org/wiki/Database => 0.003579073207718753
https://pushkarumaranikar.wordpress.com/ => 0.0
http://en.wikipedia.org/wiki/Online_and_offline => 0.0
http://software.canon-europe.com/ => 0.0
http://www.canonrumors.com/ => 0.0
http://www.rackspace.com/cloud/servers => 0.0

```

Figure 9 : Output of cosine similarity algorithm used to find relevant documents for user's query

A set of documents is build using retrieved documents. Each of these documents is formed with entered keywords and immediate three words associated with it on either sides. In above case, I parsed through retrieved documents to locate “cloud” and formed documents containing “< 3 words > cloud <3 words>” as shown in figure 10.

```

systems allocated needed cloud hosting multiple cloud
place key benefits cloud servers flexibility scalability
example longer required cloud servers also offer
information different version cloud servers please visit

```

Figure 10 : Document set for query "cloud"

Then, calculate cosine similarity of each of these documents with ads in the inventory by forming vectors. Cosine similarity ranking algorithm will give the most relevant ads for given search query.


```

Relevant Ads:
cloud computing for dummies => 0.9045340337332909
applications in the cloud => 0.6666666666666666
servers for businesses => 0.35355339059327373
web design and marketing solutions for business websites => 0.2773500981126146

```

Figure 11 : relevant ads for the entered search query “cloud”

Another Example: Search query is “android”.

```

Enter search query:
android
Document vector created for document :0
Document vector created for document :1
Document vector created for document :2
Document vector created for document :3
Document vector created for document :4
Document vector created for document :5
Document vector created for document :6
Document vector created for document :7
Document vector created for document :8
Document vector created for document :9
cosine similarity for given query with document 0 :0.0
cosine similarity for given query with document 1 :0.4684008152181554
cosine similarity for given query with document 2 :0.0
cosine similarity for given query with document 3 :0.0
cosine similarity for given query with document 4 :0.0
cosine similarity for given query with document 5 :0.0
cosine similarity for given query with document 6 :0.0
cosine similarity for given query with document 7 :0.0
cosine similarity for given query with document 8 :0.0
cosine similarity for given query with document 9 :0.0

http://en.wikipedia.org/wiki/Android_(operating_system) => 0.4684008152181554
https://pushkarumaranikar.wordpress.com/ => 0.0
http://en.wikipedia.org/wiki/Online_and_offline => 0.0
http://software.canon-europe.com/ => 0.0
http://www.canonrumors.com/ => 0.0
http://en.wikipedia.org/wiki/Database => 0.0
http://en.wikipedia.org/wiki/Cloud_computing => 0.0
http://www.microsoft.com/enterprise/it-trends/cloud-computing/default.aspx#fbid=DGpI5SyUMGd => 0.0
http://www.interoute.com/what-are-cloud-servers => 0.0
http://www.rackspace.com/cloud/servers => 0.0

```

Figure 12 : Relevant docs for given search query "android"

```

Relevant Ads:
android app development software => 1.3416407864998738
google android apps development => 0.7559289460184544
android application developers => 0.7071067811865475
android application development uk => 0.6666666666666666
google android application development => 0.6666666666666666
google android app development => 0.6666666666666666
notebook samsung => 0.3779644730092272
google adsense => 0.3779644730092272
android app developers => 0.35355339059327373
android application developer => 0.35355339059327373
app developer android => 0.35355339059327373

```

Figure 13 : relevant ads for query "android"

Conclusion

My CS297 deliverables have helped me to understand online advertisement systems. In Deliverable 1, I learned about existing advertising placement service, Google AdSense and Google's advertisement server Google AdWords which helped me to understand interest based advertising with AdChoices.

In Deliverable 2, I actually worked on Yioop. This helped me to learn about the internals of Yioop. I learned PHP to integrate configuring ad server module with Yioop. In Deliverable 3, my main focus was on designing an online auction system for Yioop. I experienced designing individual components from the proposed requirements. Deliverable 4 gave me the opportunity to learn the basics of an information retrieval system, building an inverted index, the cosine similarity ranking algorithm, and the BM25 relevance model.

Next semester, I will work on building online auction system which can be integrated with Yioop. It will be an ad server where user create ads with targeting criteria. The server will handle ad serving, ad tracking, and real-time reporting on ad impressions, clicks, and revenue.

References

- [1] AdSpeed, open source high performance ad server,
<http://www.google.com/adsense/start/>
- [2] A. Z. Broder, P. Ciccolo, M. Fontoura, E. Gabrilovich, V. Josifovski and L. Riedel,
Search advertising using web relevance feedback, Yahoo Research,
<http://gabrilovich.com/publications/papers/Broder2008SAW.pdf>
- [3] S. Buettcher, C. L. A. Clarke and G. V. Cormack, Information retrieval: implementing
and evaluating search engines, The MIT press, 2010.
- [4] M. Chiang, Networked life: 20 questions and Answers, Section 2 and 5, Cambridge
University Press, New York.
- [5] Cosine Similarity Ranking, Wikipedia,
http://en.wikipedia.org/wiki/Cosine_similarity
- [6] Google AdSense, an advertisement placing service,
<http://www.google.com/adsense/start/>
- [7] Google AdWords, An online advertising service,
<https://www.google.com/adwords/>
- [8] Yioop Documentation v 1.02, SeekQuery,
<http://www.seekquarry.com/?c=static&p=Documentation>