STANFORD
UNIVERSITY

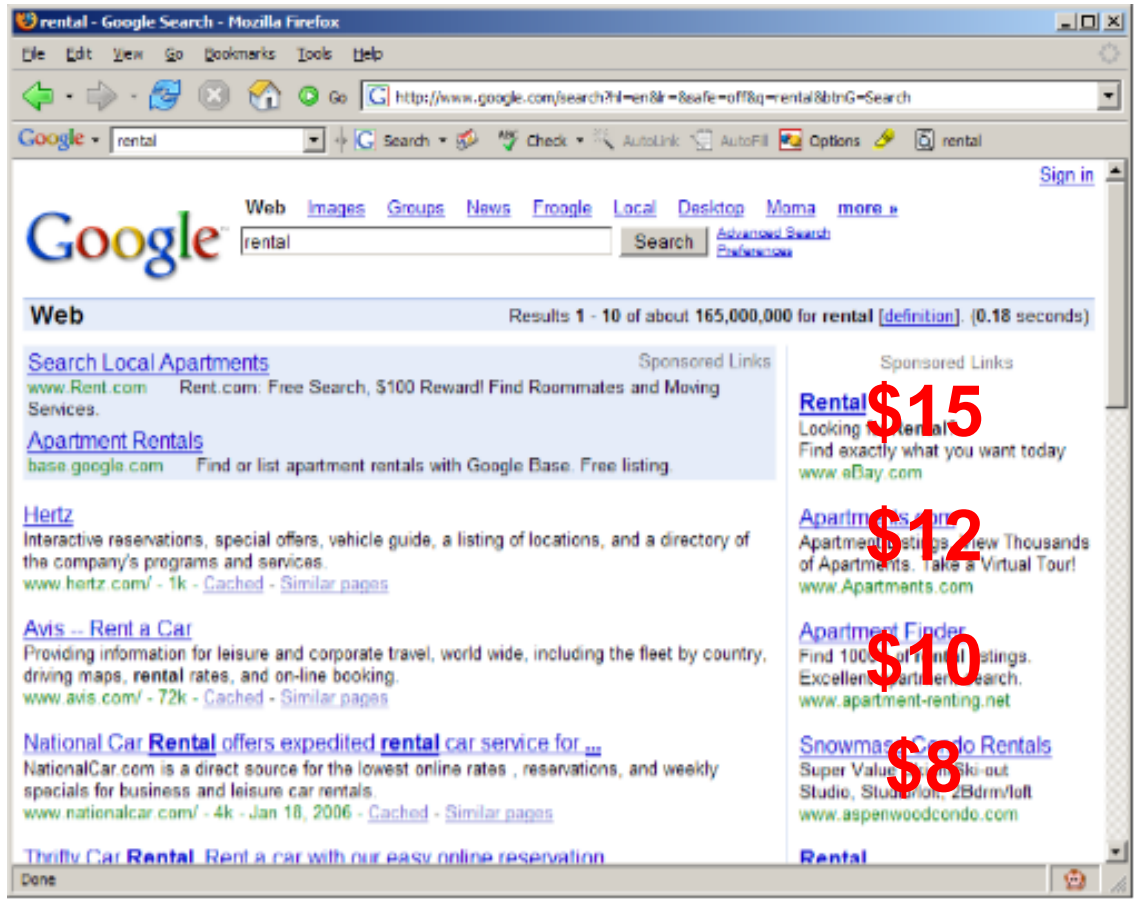
Google

Campus Talk: Targeting Ads

Sandeepan Banerjee, Google

AdWords

- Multiple ads on keywords, down right-hand-side
- Charges based on position
- Online sales



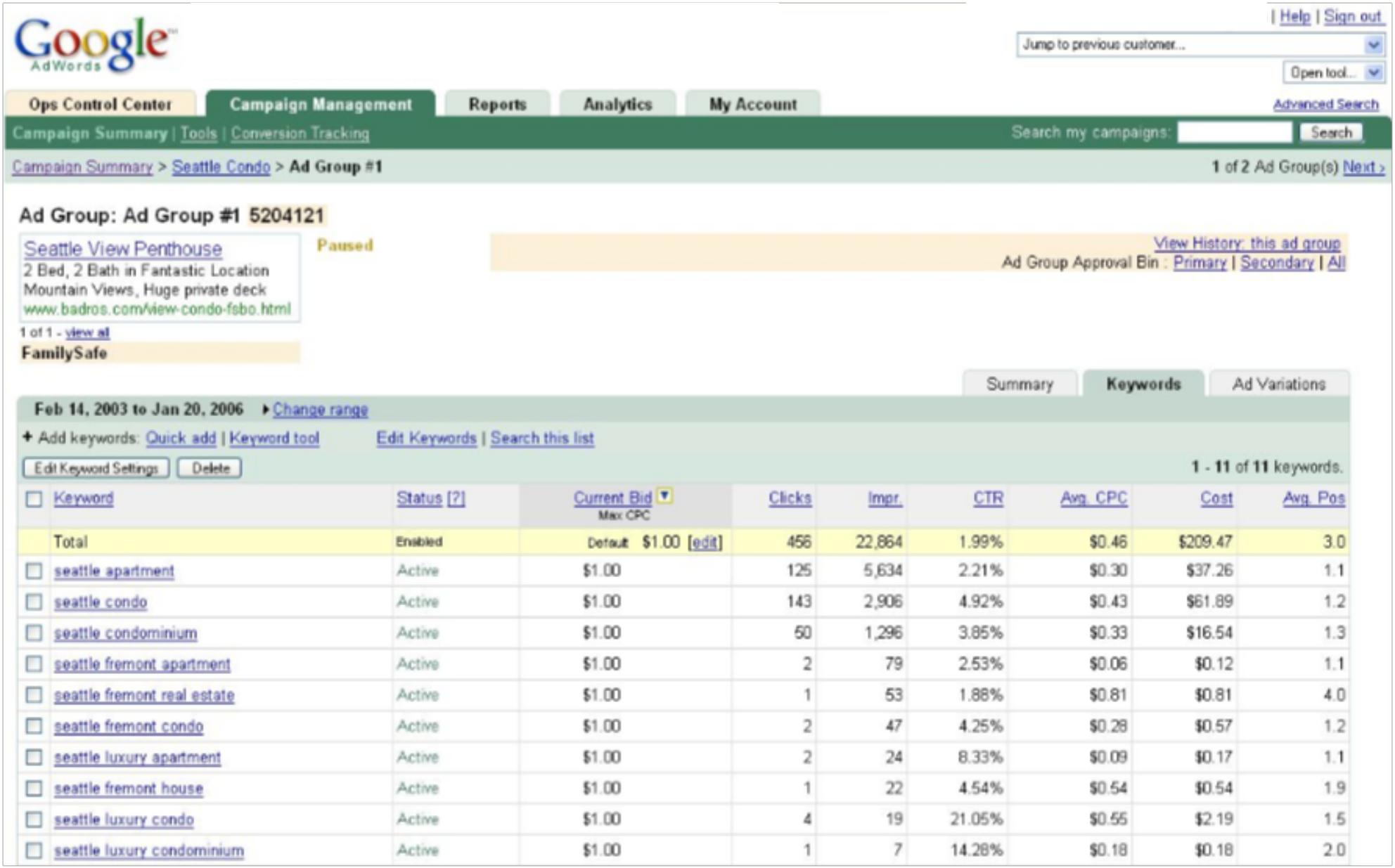
Google

Definitions Soup

- **Page inventory** available page slots for ads
- **Keywords** terms entered in a search, bought by adv.
- **Impression** showing ad to a user
- **Creative** – the text/image/video that is shown
- **CPM** Cost Per Mille (1,000 impressions)
- **CPC** Cost Per Click
- **CTR** Click Through Rate (= Clicks/Impressions)

Google

Creating an AdWords Ad



Google

Specialized Search

- Given a query, find the best ads from over 100,000 advertisers
- How do you model utility to users?
  - Want high-quality, targeted ads, that generate revenue
  - Balance importance of high click-through-rate (CTR) with advertiser's willingness to pay
- Auction theory helps!

Google

Ranking Ads

Keyword: **skydive**

Skydive with Us
Only one accident last year.
Have fun and play the odds!
www.**skydive**withus.com

CPC=$0.40, CTR=2%

Effective CPM = $0.40*20=$8

Need **Skydiving** Insurance?
We've got your back.
Even if you lose, you win!
www.**skydive**insurance.com

CPC=$0.20, CTR=5%

Effective CPM = $0.20*50=$10

Google

Ad Auction Ranking

- $0.40 and $0.20 are “bids” per click reflecting the *maximum* CPC the advertiser is willing to pay
- Insurance company could have bid $0.16001 CPC
  had eCPM = $0.16001*50 = $8.005
  and still gotten ranked #1
- So… we act as if they did:
  they pay only $0.16/click, not $0.20

Google

Let Advertisers Bid True Value

- The system acts in their best interest
- No need to increase their bid when someone else gets ranked ahead of them
- When there's no competition, you pay the minimum
- The minimum based on quality of the ad, based on a user-driven assessment

Auction Basics

- English – “going going gone!”
- Dutch – price dropped until someone bites
- 1st price sealed – winner pays their bid

“Winners curse” • “Bid Shading”

➔ Complicates selecting a bid

Google

Vickrey Auction

- 2nd price sealed – pay 2nd highest bid
- All 4 auctions have the same expected revenue for seller
- Vickrey has simplest bidding strategy:
  Just bid your true value
  (no bid shading, no winner's curse)
- Won Nobel Prize in Economics (1996)

Google

Engineering challenge: Predicting CTR

- Dizzying set of factors could affect clickthrough
  - Country, time of day, targeted text vs query, …
- How does one automatically figure out which factor is more relevant?
  - How to update model quickly in face of change
  - How do you estimate CTR for not-yet-shown ads?

The 10 billion-dollar JavaScript snippet…

```
<script type="text/javascript"><!--
google_ad_width = 728; google_ad_height = 90;
google_ad_format = "728x90_as"; google_ad_type = "text_image";
//--></script>
<script type="text/javascript"
src="http://pagead2.googlesyndication.com/pagead/show_ads.js">
</script>
```

Google

The Power of Data applied to Contextual Targeting

- Conventional wisdom:
  - Given an order of magnitude increase in computational power…
  - … you can solve previously impractical problems
- Unconventional wisdom
  - Given an order of magnitude increase in data…
  - … you can solve previously unsolvable problems!
- Consider how to determine similarity between text:
  - How similar is “Kofi Annan” to “UN Secretary-General”?

Google

Traditional Information Retrieval Similarity

- Traditionally: Similarity is function of *term frequency within a document* and *across all documents*
- TF(w) = frequency of term w in a document/query
  - Intuition: a word appearing more frequently in a text is more likely to be related to its "meaning"
- IDF(w) = log [N/n_w] + 1

  where N = # documents, n_w is # documents containing w
  - Intuition: words appearing in many documents are generally not very informative (e.g., "the")
- TFIDF: contribution of a term is product of quantities:

  TFIDF(w) = TF(w) x IDF(w)

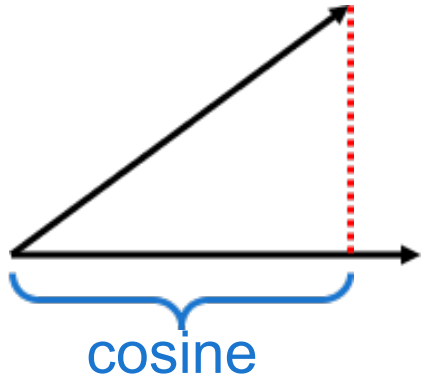
Google

Using TFIDF to Measure Similarity

- Consider each document as a vector:

```
dog compute window ...
```

Doc. 1 = < 3.2, 0, 1.2, ... >
Doc. 2 = < 0, 2.1, 5.4, ... >
Doc. 3 = < 0, 1.7, 0, ... >



- Vectors are constructed such that
  - Each dimension of vector represents a term w_i
  - Each entry of vector has value: TFIDF(w_i)
  - Normalize the vectors to unit length (Euclidean norm)
- Similarity of two texts is measured by the cosine between the TFIDF vectors of the documents/queries
  - Cosine = vector dot product!

Google

Determining Similarity of Short Text Snippets

- Many queries on the web are short (~2.5 words)
- For short text snippets, cosine is insufficient
- Cosine of term vectors for all following text pairs is 0:
  - “AI” “artificial intelligence”
  - “Kofi Annan” “UN Secretary-General”
  - “Eric Schmidt” “Google CEO”
  - “NASA” “space exploration”
  - “Larry Page” “Google founder”
- Should also identify unrelated concepts, even if high term overlap
  - “Larry Page” “web page”

Google

Determining Contextual Similarity of Short Text

"… the meaning of a word is its use in the language"

~ Ludwig Wittgenstein

- For short text snippets, need to determine greater contextual meaning
- Insight: leverage huge quantity of web information!
- Approach: Expand short text snippet into vector with additional context terms
  - Find terms that co-occur on web with terms in text snippet to determine contextual vector
  - Similar to "query expansion" in Information Retrieval

Google

Leverage the Web to Determine Similarity

- Let *x* and *y* be two short text snippets
- Want to define a function *f*(*x*, *y*) that measures "semantic" similarity between *x* and *y*
- Define "query expansion" of text *x*, **QE**(*x*), as follows:
  - Issue *x* as query to search engine (oh, say, Google…)
  - Let R be retrieved set of N documents: $\{D_1, \ldots, D_N\}$
  - Compute TFIDF vector $\mathbf{V}_i$ for each document $D_i \in R$
  - Compute **QE**(*x*) as average (centroid) of all vectors $\mathbf{V}_i$
- Define *f*(*x*, *y*) = **QE**(*x*) • **QE**(*y*)

Google

How Well Does This Work?

- Recall previous text pairs: $f(x,y)$ Cosine

  ("AI", "artificial intelligence") 0.831 0.000

  ("Kofi Annan", "UN Secretary-General") 0.825 0.000

  ("Eric Schmidt", "Google CEO") 0.845 0.000

  ("NASA", "space exploration") 0.691 0.000

  ("Larry Page", "Google founder") 0.770 0.000

  ("Larry Page", "web page") 0.123 0.500

- Consider multi-faceted term "Java":

  ("Java island", "Indonesia") 0.454 0.000

  ("Java programming", "Indonesia") 0.020 0.000

  ("Java programming", "applet development") 0.563 0.000

Google

More Info

- Sandeepan Banerjee works on Google's Infrastrucutre (Crawling, Indexing, Storage, Map/Reduce, Video, International projects, …)
- Send follow-up questions to sandeepan@google.com

Google