Learning From Weights: A Cost-Sensitive Approach For Ad Retrieval

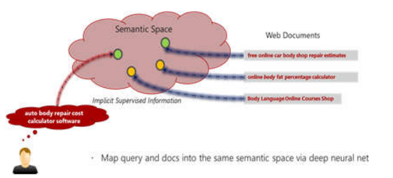
Nikit Begwani*
Microsoft AI and R
nibegwan@microsoft.com

Shrutendra Harsola*†
Intuit AI
shrutendraj@gmail.com

Rahul Agrawal
Microsoft AI and R
rahulagr@microsoft.com

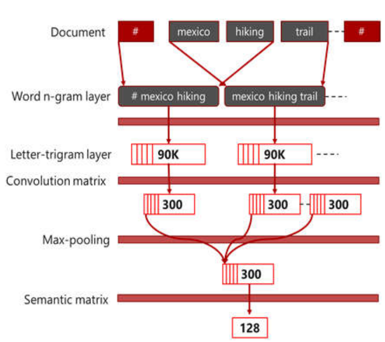
Semantic Match For Ads

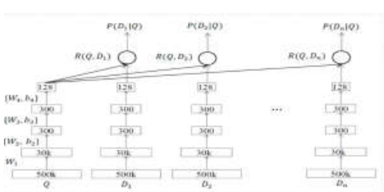
Vector Based Approach



- Vector Based Approach
  - Create Semantic vector representation of Text
    - Query
    - Listing (Keyword, Ad-document)
  - Find Matches based on Vector similarity Using Offline Graph Search
    - Cosine Distance

Sample Non-Interaction based NN for IR





*Y Shen, X He, J Gao, L Deng, G Mesnil. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. ACM, 2014

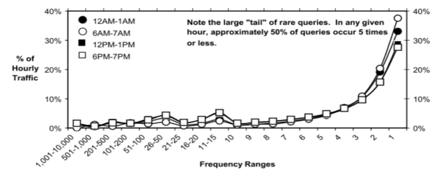
Cost-Sensitive Approach

Why we need Cost Sensitive Learning ?

- Not all Query Doc pairs are equal in nature

Query	Doc	Clicks
jet2holidays	jet2holidays book your dream holiday today	43516
lassi kefalonia holidays [Tail]	holidays lassi book today jet2holidays	1

- Long tail nature of queries



How to apply Cost Sensitive Approach ?

CLSM Loss Function

- D′ contains D⁺ and J randomly selected unclicked documents.

$$P(D^+/Q) = \frac{exp(R(Q,D^+))}{\sum_{D'\in D} exp(R(Q,D'))}$$

$$L(\Lambda) = \sum_{(Q,D^+)}(-log(P(D^+/Q)))$$

Interpretation

- For J=1 i.e. one negatively sampled doc, this loss is same as pair-wise loss of RankNet*

$$C = -log\big(P(D^+|Q)\big) = log(1 + exp(s^- - s^+))$$

Where $s^+ = R(Q,D^+)$ and $s^- = R(Q,D^-)$

$$\frac{\partial C}{\partial \Lambda} = \frac{exp(s^- - s^+)}{1 + exp(s^- - s^+)}\left(\frac{\partial s^-}{\partial \Lambda} - \frac{\partial s^+}{\partial \Lambda}\right)$$

- Cons: Assigns same label (+1) to all clicked docs i.e. $y(Q,D^+) = 1$

Solution: Optimize list-wise loss (DCG) with real-valued labels

- Assign true label based on probability of click :

$$y(Q,D) = \frac{Clicks(Q,D)}{Impr\ (Q,D)}$$

- As shown in LambdaRank⁺, we can optimize DCG by multiplying gradients with $|\Delta_{DCG}|$
  - Δ_{DCG} = change in DCG on swapping ranks of D^+ and D^-

$$\frac{\partial C}{\partial \Lambda} = \frac{|\Delta_{DCG}| * \exp(s^- - s^+)}{1 + \exp(s^- - s^+)}\left(\frac{\partial s^-}{\partial \Lambda} - \frac{\partial s^+}{\partial \Lambda}\right)$$

- On swapping D_j with a negative doc: $|\Delta_{DCG}| = y_j / \log(1 + j)$
- For j=1 i.e. swapping with top most doc: $|\Delta_{DCG}| = y_j$

Interpretation

- Same as weighing each train point $\{Q,D^+\}$ by weight $y(Q,D^+)$
- For CLSM, train data weighing is same as optimizing DCG

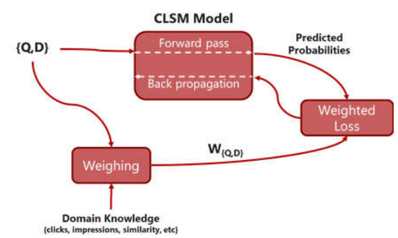
$$L(\Lambda) = -\sum_{(Q,D^+)} y(D+,Q) * log\big(P(D^+/Q)\big)$$

*Burges et al. Learning to rank using gradient descent. In ICML 2005.

⁺Chris Burges. From RankNet to LambdaRank to LamdaMart: An Overview. 2010.

WCLSM

Proposed Model



Weighing Strategy

- Guiding Principles
  - **Global** in nature i.e. weight for a (query, doc) should be comparable across documents and queries
  - Weight should **not be biased** towards **head/torso/tail** queries or documents
  - Weight should be **proportional to the clicks**

Given click logs over N queries and M documents,
I_{ij} represents number of **impressions**
c_{ij} number of **clicks**
and w_{ij} **weight** for $\{Q_i, D_j\}$ pair.

$$w_{ij} = \frac{c_{ij}}{I_{ij}}$$

CTR: Click through rate and is computed as number of clicks divided by number of impressions.

Experimental Setting

Unweighted [Curated Training Dataset]

- Take all those query ad pairs as training data which are above predefined thresholds
- Thresholding may be on clicks, impressions, or CTR
- Cons:-
  - Miss out on data exploration
  - End up training a head/torso heavy model

Unweighted [Complete Training Dataset]

- Take all those query ad pairs which have at least one click
- Pros :-
  - Regain lost training data - which takes care of exploration
  - Learn the tail better
- Cons:-
  - May spoil representations for head query and documents

Weighted [Complete Training Dataset]

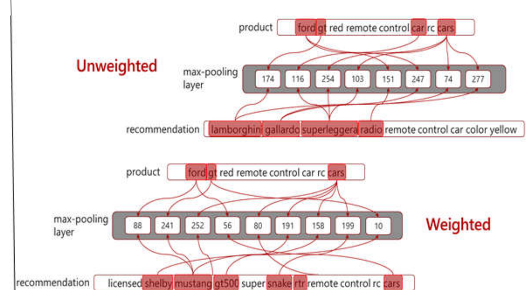
- Take all those query ad pairs which have at least one click
- Weigh them based on historical performance
- Pros :-
  - Complete Training Data
  - Learn the tail better w/o losing out on head representation

Results

#	Model Type	AUC-ROC	AUC-PR
1	CLSM Curated	72.03%	71.62%
2	CLSM Unweighted	73.74%	73.24%
3	**CLSM Weighted-CTR**	**74.25%**	**73.91%**
4	Bi-LSTM Unweighted	74.17%	73.56%
5	**Bi-LSTM Weighted-CTR**	**74.84%**	**74.29%**

Purchased product	Title of top - 1 returned product
comfort control harness **large**	*Weighted:* comfort control harness **xxl** blue *Unweighted:* comfort control harness
ford gt red remote control car rc cars	*Weighted:* licensed **shelby mustang** gt500 super **snake** rtr remote control rc cars *Unweighted:* lamborghini gallardo superleggera radio remote control car color yellow
nasa mars curiosity rover **spacecraft** poster 13x19	*Weighted:* space shuttle **blastoff** poster 24x36 *Unweighted:* imagination nebula motivational photography poster print 24x36 inch
wonderful **wonder world clockmaker**	*Weighted:* **alice country clover** ace hearts *Unweighted:* olympus

What does the model learn?



Bing Sponsored Search Deployment

A	B	Clicks	Click-through rate (CTR)	Bounce rate
		% change of metrics in A as compared to B		
Unweighted	Curated	+10.11%	+10.08%	+2.85%
Weighted	Curated	+22.78%	+23.13%	-5.63%
Weighted	Unweighted	+11.51%	+11.85%	-8.25%

- **Clicks** : Total clicks generated on displayed ads.
- **Click Through Rate (CTR)** : Number of ads clicked out of number of ads shown.
- **Bounce Rate (BR)** : Percentage of times user returned back to the search engine immediately after clicking an ad.

*These authors contributed equally to this work.
†This work was done while the author was affiliated with Microsoft AI and R.