Exploration of the Effect of Category Match Score in Search Advertising

Youngchul Cha #[1], Junghoo Cho #[2], Jian Yuan *[3], Tak Yan *[4]

# *Computer Science Department, UCLA*
Los Angeles, California 90095, USA
[1,2] {youngcha, cho}@cs.ucla.edu

* *Microsoft*
Sunnyvale, California 94089, USA
[3,4] {jianyuan, takyan}@microsoft.com

Abstract—**Categorical (topic) similarity between a web page and an advertisement (ad) text has long been used for contextual advertising. In this paper, we explore the use of the categorical similarity score, referred to as Category Match Score (CMS), in the context of *search advertising*. In particular, we explore the effect of CMS on various ad-effectiveness prediction tasks, including user-judgment prediction, ad click-through-rate prediction (CTR), and revenue-per-impression prediction. Our extensive experiments on two editorial datasets and one live traffic dataset demonstrate that CMS is one of the strongest features in the judgment prediction task and that CMS-based filtering is very effective in improving revenue per impression as well as CTR. We believe that our analyses can be extremely effective in helping web service providers serve more relevant and profitable ads to users.**

I. Introduction

Computational advertising provides money to the web ecosystem. Most web services are operated by the money generated by clicks online users make. Without computational advertising, most web services cannot be freely provided, meaning users would need to pay for the web services they currently use free of charge. It would result in fewer users and the web ecosystem would become smaller and less active.

However, users do not like ads because they have learned from their life-long experience that ads are usually not relevant to the content they are watching. If a web service provider can serve more relevant ads, users will get annoyed less and show more interest in them. They would click more ads and it would lead to higher revenue for the service providers because they are paid by advertisers when those ads are clicked.

Despite recent relevance improvements achieved by current advertising technologies, serving relevant ads is an endless problem. First, by nature, it is a very hard problem to find a few most relevant ads among millions of candidate ads in realtime under changing contexts. Second, even a slight increase in ad click-through rate (CTR) would lead to a huge increase in the web service provider's revenue. Thus, many computationally expensive machine learning techniques are now applied to improve CTR and relevance of ads, some of which use tens of thousands of features (variables) to estimate relevance of ads more correctly.

To achieve more accurate relevance estimation, machine learning approaches usually depend on three things. First, the amount of data matters. Larger amount of data normally produces better results. Second, the number of good independent features is important. Though too many features may cause over-fitting, as long as the features are carefully chosen, a larger number of good features guarantee a better result. Third, selection of good machine learning algorithms is also important. The third is one of the most active fields in the machine learning research area. However, many machine learning experts argue that selection of good algorithms is not as important as selection of good features [1], [2].

In this paper, we are attempting to apply a new feature called "Category Match Score (CMS)" to serving better search ads. CMS was initially developed for contextual ads that are displayed next to the content a user watches. It measures categorical similarity between a web page text and an ad text. To the best of our knowledge, the effect of CMS on search advertising has not been systematically analyzed. We show that CMS is very effective in serving more relevant and profitable search ads to users. We additionally show that "actual bid", the final bid price after an auctioning step, is a very useful feature.

In summary, we make the following contributions in this paper:

1 We show that CMS and actual bid are very strong features in the context of machine learning. In every query-ad relevance judgment prediction task, CMS and actual bid are picked as one of the top-5 strong features in terms of feature gain.
2 We demonstrate that prediction accuracy can be largely improved by incorporating CMS and actual bid in a prediction model, with reasonable explanations. We also show that CMS-based filtering can be effectively used to select more relevant ads.
3 We explore how much improvement we can achieve in terms of profit metrics. We demonstrate that CMS-based filtering is very effective in improving CTR and per-impression revenue in a real service environment.

978-1-4799-2555-1/14/$31.00 © 2014 IEEE

II. Background

In this section, we explain some background knowledge for our work. We first briefly explain how relevant ads are selected when a user types a query or visits a web page. Then, we introduce CMS, which was originally developed to efficiently measure similarity between a web page and an ad text. We also explain two filtering steps we apply in our experiments.

A. Selecting Relevant Ads in Computational Advertising

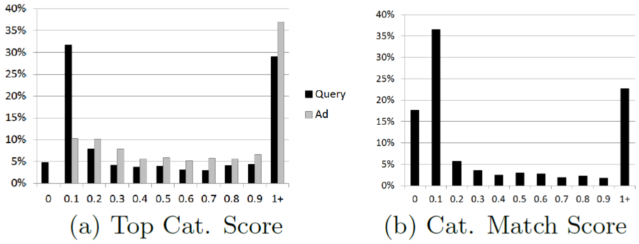
There are traditionally two major types of textual online ads: search ads and contextual ads. The process of selecting relevant search ads is very similar to the Information Retrieval (IR) process. It is basically selecting the top-k most relevant entries from a huge amount of candidates and consists of two procedures: 1. “filtering-out” less-relevant ads, and 2. “ranking (sorting)” the remaining ads based on relevance and price. In the case of selecting relevant contextual ads, an additional “keyword extraction” or “category (concept) extraction” step is required in the first filtering-out procedure.

The main purpose of the filtering-out procedure is to remove less-relevant candidate ads from the millions in an ad corpus. As the procedure needs to process a tremendous number of entries, it usually depends on simple calculation using textual information. For search ads, the similarity between an expanded query [3], [4] and an ad text (usually bid phrases) is used for this purpose. More elaborate WAND [5], [6] calculation can also be used, where each different source (e.g. a text from a bid phrase and one from ad description) may have different weight.

For the ranking procedure, more sophisticated machine learning algorithms are used to sort the remaining hundreds of candidate ads in decreasing order of predicted relevance. Differently from the IR ranking procedure, an additional auctioning step is required to achieve higher revenue. Thus, even though an ad has very high relevance to the query, if its bid price is too low, it may not have a chance to be presented to a user. Because second-price auction [7], where the winner usually pays one cent more than the bid price of the second-place winner, is normally used in this auctioning step, the price the winner pays is different from her “initial bid” price. We call this final price “actual bid”.

B. Category Match Score (CMS) and CMS Filtering

At Bing AdCenter, the category (concept) extraction mentioned in the previous section is performed by Category Hierarchy Engine (CHE) [8]. If we input a text to the CHE, it returns multiple categories with confidence scores ranging from 0 to 1. As the categories are sorted in descending order of the confidence score, the top category is considered as the most relevant category to the text. Each category corresponds to a node in the “category hierarchy tree”, where each node is allowed to have multiple parents. For example, *SnowGoggle* can have *Ski* and *Snowboard* as its parents. Thus, if *Ski* and *Snowboard* have *WinterSports* as its parent and *Sports* as its grand parent, there exist two category paths for the node *SnowGoggle*: */Sports/WinterSports/Ski/SnowGoggle* and */Sports/WinterSports/Snowboard/SnowGoggle*. When using the CHE to extract categories from a query, as the query is very short and may not contain commercially interesting words, the returned confidence scores are inevitably lower than the scores from a web page or an ad text. This low score may be one of the main reasons why CHE has not been used in search advertising. We observe these different distributions from the left panel of Fig. 1, which compares the top category scores from a query and an ad text.



(a) Top Cat. Score

(b) Cat. Match Score

Fig. 1. Distribution of TCS/CMS in an editorial dataset

To measure the similarity between the categories from a query and an ad text, we use CMS, a variant of TaxScore (taxonomy score) described in [8]. It is based on WAND [5], [6] calculation on expanded categories from the query and the ad side. More precisely, the CMS of a query Q and an ad text A is calculated as:

$$\begin{aligned} CMS(Q,A) &= \sum_{c \in CHE(Q)} Expand(c) \cdot \sum_{c \in CHE(A)} Expand(c) \\ &= \sum_{c \in CHE(Q)} \overrightarrow{V_c} \cdot \sum_{c \in CHE(A)} \overrightarrow{V_c}, \end{aligned}$$

where $CHE(T)$ denotes a returned category set for a text T, and c denotes a category in the returned category set. The function $Expand$ expands a category along with its parents in the category hierarchy tree with appropriate weight and produces the vector $\overrightarrow{V_c}$, where each element represents a score for each category in the expanded category set.

The right panel of Fig. 1 shows CMS distribution in an editorial dataset we use in this paper. We can observe that about 50% of entries have CMS below 0.1. Given that CMS is an additive sum of confidence score multiplications, this distribution could be problematic because CMS of 0.1 indicates that it may not be credible. From the left panel of Fig. 1, we can infer that one of the main reasons for a low CMS value is a low query category score originated from a short query or a non-commercially interesting query.

In this paper, we attempt to explore the effect of CMS on prediction accuracy, relevance, CTR, and per-impression revenue. Thus, we try various CMS cut-offs to closely look at the changes in these metrics according to the different cut-offs. Since query-ad pairs with lower CMS produce poorer results in terms of those metrics, as we show in the next section, we prefer the pairs with higher CMS. In this sense, we call this step CMS-based filtering (simply, CMS filtering). We also try different cut-offs for a query’s Top Category Score (TCS) because it is closely related to CMS as we observe in Fig. 1.

TCS filtering is done right after getting query categories from the CHE.

III. Experiments and Analyses

The primary goal of this paper is to investigate whether the use of CMS in the context of search ad selection is beneficial. To address this issue, we explore the following questions in the rest of this paper.

Q1 *Are CMS values correlated to the user's judgments on whether a particular ad is relevant to a query?*

Q2 *Are CMS values likely to be picked up by our machine-learning-based classifier?*

Q3 *How much improvement can we achieve by using CMS values?*

To explore these questions we perform a number of experiments on two editoral datasets and one live-traffic dataset collected from a commercial search engine.

A. Datasets

Before we proceed to our experimental setup, we first describe our datasets in more detail.

The two editorial datasets for our experiments were gathered from Microsoft in the United Kingdom (UK). They are referred to as *Ad Copy judgments (ACP)* and *Ad Landing Page judgments (ALP)*. The first dataset, ACP, contains human judgments on the relationship between a query and an ad text and the second dataset, ALP, contains human judgments on the relationship between a query and an ad landing page (the page that the user lands on when she clicks on the ad). Both datasets are very useful for our experiments because of their different nature: the ACP dataset records a *relevance level* between a query and an ad text (i.e., how much are they relevant to each other?) and the ALP dataset records different *types of relevance* between a query and an ad landing page (e.g., Is the query more specific than the ad? Are the query and the ad completely disjointed?).

More precisely, the ACP dataset contains 46K entries, where each entry corresponds to a query-ad pair and other associated features including judgment, CMS, actual bid, etc. Here, judgment is made by human participants who recorded the relevance between the query and the ad at five different scales: *Bad*, *Fair*, *Good*, *Excellent*, and *Perfect*. On the other hand, the judgment in the ALP dataset has a value among *Spam*, *BadLink*, *Aggregator*, *Disjoint*, *Overlap*, *Subset*, *Superset*, and *Same*. Note that most ALP judgments are based on set relationships such as *Disjoint*, *Overlap*, *Subset*, *Superset*, and *Same*. For example, a *Subset* judgment means that a human editor judged that the query is more specific than the matched ad (e.g. *digital camera* and *camera*) and a *Superset* judgment implies the opposite case (e.g. *camera* and *digital camera*). Both datasets have *Unsure* judgments, for which the human participants cannot clearly determine the relationship between a query and an ad. The ACP dataset is useful for addressing our first and third question (Q1, Q3) and the ALP dataset can be used for addressing our second and third question (Q2, Q3). The ALP dataset has 103K entries equipped with more features than those of the ACP dataset. However, we are mainly interested in judgment, CMS, and actual bid [1], all of which are commonly available in both datasets.

The live traffic dataset was collected from real UK search queries. We sampled entries from five-days of traffic between June 21st and 25th of 2012. There are 2,072,407 ad impressions and generated by 1,863,915 queries. Among these matching ads, we filtered out "dynamic ads" because their ad text contains $\{$query$\}$ instead of the query word itself and may produce incorrect categories when inputted to the CHE. (The $\{$query$\}$ variable is replaced with the input query when displayed to the user.) All the remaining "static ads", which do not contain variable phrases such as $\{$query$\}$ in their ad text, have 1,038,806 impressions and 18,794 of them were clicked. We grouped the remaining impressions by a unique query-ad pair and selected top-1000 query-ad pairs based on their number of impressions, because bottom query-ad pairs normally do not have enough number of impressions and clicks to produce a reliable CTR value. Thus, each entry in the processed live traffic dataset consists of a query-ad pair, number of impressions, number of clicks, and summed revenue for that pair. Note that the live traffic dataset does not contain initial bid or actual bid for all entries. It contains revenue (actual bid) only for a clicked impression.

With these datasets, we conducted five experiments to explore three questions described above: 1. correlation analysis (ACP, Q1), 2. feature selection analysis (ALP, Q2), 3. prediction accuracy analysis (ALP, Q3), 4. relevance level analysis (ACP, Q3), and 5. CTR and revenue analysis (Live, Q3).

B. Correlation Analysis

We first explore whether CMS is correlated to a user's judgment on relevance between a query and an ad (Q1). For this purpose, we measured correlation between CMS and judgment in the ACP dataset, where each judgment is about a relevance level and can be converted into a value. We simply defined RelevanceLevel by assigning 0, 2, 5, 7, and 10 to Bad, $Fair$, $Good$, $Excellent$, and $Perfect$ judgment, respectively (We ignored $Unsure$ judgment.). After grouping entries by RelevanceLevel, we calculated average CMS per group and measured correlation between RelevanceLevel values and average CMS values. We also calculated correlations between RelevanceLevel and other available features. CMS showed the second largest correlation of 0.42 with RelevanceLevel, only preceded by actual bid's 0.88 [2]. Since actual bid shows the highest correlation with RelevanceLevel, we also explore the effect of actual bid in the following experiments.

C. Feature Selection Analysis

In the second experiment, we explore whether CMS and actual bid are practically useful in the machine learning context. For this purpose, we incorporated these two features with top-2K features selected from tens of thousands of

[1] Note that all the ads in the datasets have actual bid value because it is calculated for all the candidate ads to be displayed

[2] Kendall's τ and Speanman's ρ showed similar results.

TABLE I
TOP-10 FEATURES IN THREE PREDICTION TASKS

$Aggregator$		$Disjoint$		$Overlap$	
CMS	1.0000	**ActualBid**	1.0000	**CMS**	1.0000
AgeInMinutes	0.7206	**CMS**	0.9698	AgeInMinutes	0.9116
ActualBid	0.7096	BM25F	0.9652	**ActualBid**	0.8934
ProximityBM25FNorm	0.6783	AgeInMInutes	0.9524	ProximityBM25FNorm	0.6157
BM25FNorm	0.6289	WordFoundBM25F	0.7829	ProximityBM25F	0.5922
ProximityBM25F	0.5300	ProximityBM25FNorm	0.7308	BM25F	0.5346
StreamLength_AdDescription	0.3836	BM25FNorm	0.6791	BM25FNorm	0.5324
WordFoundBM25F	0.3672	StreamLength_AdDescription	0.5592	WordFoundBM25FNorm	0.4724
BM25F	0.3514	ProximityBM25F	0.5511	StreamLength_AdDescription	0.4681
TAU_AdDescription	0.3413	StreamLength_AdTitle	0.4943	StreamLength_AdTitle	0.3974

currently used machine learning features and see whether these two features are picked as one of the top-10 features in terms of feature gain. The top-2K “base” features are mostly related to ad title, ad description, bi-gram, link structure, and BM25F [9]. We applied TLC Boosted Tree Classifier, a GBDT implementation by Microsoft Research, to the prediction task of the nine ALP judgments. One benefit of the TLC Boosted Tree Classifier is that it provides a ranked list of features based on their feature gains. Table I shows top-10 features for three judgment prediction tasks: $Aggregator$, $Disjoint$, and $Overlap$. We observe that the newly added CMS and actual bid are selected as one of the top-5 features among the $2K$ strong features. Though we only report the three tasks in this section due to lack of space, they are listed in top-5 positions in every prediction task except the *Unsure* judgment prediction task, where we could not apply a machine learning classifier due to the small number of the *Unsure* judgments in our dataset. From this observation, we see great potential in using CMS and actual bid over other currently used machine learning features. The other consistent top-10 features include AgeInMinutes (lifetime of an ad), BM25F, and BM25F’s variants. Note that AgeInMinutes is stronger than the famous BM25F and its variants. We give our explanation to this interesting observation in the next section.

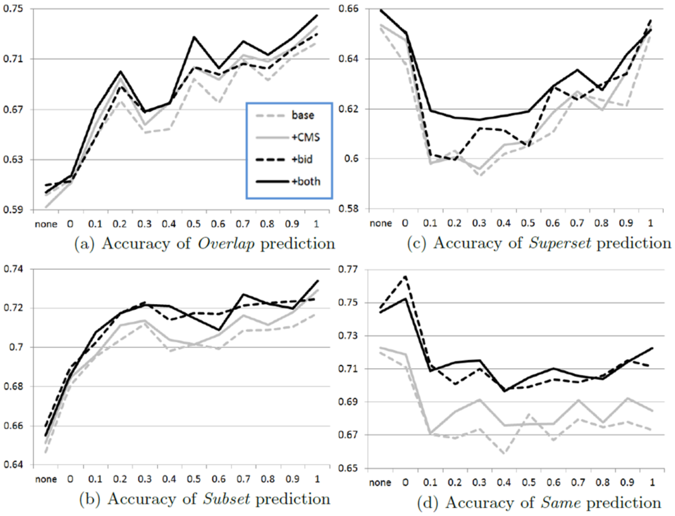
D. Prediction Accuracy Analysis

In the previous experiments, we verified that CMS and actual bid are highly correlated to judgement and significantly useful in machine learning tasks. From the third to the last experiment, we explore how much improvement we can achieve by using CMS and actual bid (Q3). Firstly, we explore how much prediction accuracy can be improved with these new features using the ALP dataset. Since one of our main goals is to explore the effect of CMS filtering described in Section II-B, we filtered the ALP dataset with 12 different CMS cut-offs: $none$, 0, 0.1, 0.2, ..., 1, where $none$ stands for the whole dataset without any filtering. We denote each of them as D, D_0, $D_{0.1}$, $D_{0.2}$, ..., D_1, where D denotes a dataset and each subscript denotes a cut-off. We also use the superscript, *base*, $+CMS$, $+bid$, and $+both$, to denote which features are in the dataset. For example, $D_{0.2}^{+both}$ denotes a dataset filtered by the CMS cut-off of 0.2 having the 2K base features, CMS, and actual bid. We made the size of each sub-dataset the same by randomly sampling $|D_1|$ entries from each sub-dataset because the dataset size is very important in a machine learning task. The TLC Boosted Tree Classifier with 10-folds validation was applied to the 48 sub-datasets.

Fig. 2 shows some noticeable judgment prediction results. In each sub-figure, the horizontal axis denotes CMS cut-offs in increasing order (*none* denotes no-filtering) and the vertical axis denotes AUC values. AUC is preferred to simple accuracy due to its stronger discriminant power [10] and a higher AUC value means better accuracy. We plot the AUC values for the four feature sets. The difference between dotted and solid lines shows the effect of adding CMS to the prediction model and the difference between gray and black lines shows the effect of adding actual bid to the model. The AUC value changes in the y-axis according to the CMS cut-offs in the x-axis shows the effect of CMS filtering.

Fig. 2(a), 2(b), 2(c), and 2(d) depict AUC values from *Overlap*, *Subset*, *Superset*, and *Same* prediction task respectively. We observe clear increasing patterns in the *Overlap* and the *Subset* prediction tasks, which imply that CMS filtering is very effective in achieving more accurate predictions for these prediction tasks. The improvements achieved in those predictions are 21.79% and 11.19% each. However, these increasing patterns are not observed in all prediction tasks. We observe a bouncing pattern in Fig. 2(c) and a decreasing pattern in Fig. 2(d). Also, the prominent gaps between gray and black lines in Fig. 2(d) indicates that adding actual bid to the model is very effective in the *Same* prediction task. Differently from CMS filtering, adding CMS to the model does not show noticeable improvements.

Table II summarizes the effect of CMS, actual bid, and CMS filtering on prediction accuracy improvements in all prediction tasks. As all our new features seem effective, we may selectively incorporate them into the second ranking procedure of the ad-selection process to achieve better prediction accuracy. The reason why CMS is helpful in these prediction tasks is probably because it measures how a query and an ad are conceptually similar even though they do not share common words (e.g., query *ski* and ad *snow goggle*). (With traditional cosine similarity, their similarity is 0 because they do not share common words.) One interesting thing in this table is that actual bid shows consistent improvements in all prediction tasks. We argue that the possible reasons are: 1. an advertiser paying more money pays more attention to the quality of her ad, or 2. a proven ad lets the advertiser pay more. These



(a) Accuracy of *Overlap* prediction

(c) Accuracy of *Superset* prediction

(b) Accuracy of *Subset* prediction

(d) Accuracy of *Same* prediction

Fig. 2. Prediction accuracy of judgment prediction tasks

reasons might be similar to the reasons why AgeInMinutes is picked up as one of the top-10 features in every prediction task as exemplified in Table I: 1. a long-lasting ad has good quality, or 2. a proven ad is long-lasting. Note that the actual bid value we used in our experiments is the ad price after the auctioning step and is not available before that step. Though we wanted to incorporate the initial bid price to the prediction model, our editorial datasets do not have it. However, we believe that the initial bid price (or historical average of actual bids of an ad) is also helpful in improving prediction accuracy from the above reasoning. Though bid price has been mainly used for selecting a winning ad at the auctioning step, our results imply that the bid price can be incorporated before the auctioning step to achieve higher prediction accuracy.

TABLE II
SUMMARY ON PREDICTION ACCURACY IMPROVEMENTS

Task	+CMS	+bid	+both	CMS Filtering
Aggregator	0.15%	7.44%	9.53%	16.86%
BadLink	-0.11%	1.35%	1.74%	10.27%
Disjoint	0.12%	0.96%	1.33%	-4.20%
Overlap	1.34%	1.31%	2.73%	21.79%
Same	1.29%	5.07%	5.33%	-4.85%
Spam	0.50%	3.12%	3.32%	0.46%
Subset	0.72%	1.79%	1.83%	11.19%
Superset	-0.54%	0.51%	1.20%	2.15%

E. Relevance Level Analysis

In the fourth experiment, we use the ACP dataset to explore how much improvement we can achieve in terms of RelevanceLevel prediction accuracy. When the RelevanceLevel of a query-ad pair is high, it means that people think the ad is highly related to the query and are more likely to click it. If we can correctly predict RelevanceLevel based on given features, we are able to provide more relevant ads to the users by filtering out the ads with low predicted RelevanceLevel values. Thus, we trained TLC Boosted Tree Regression to minimize RMSE (Root Mean Squared Error) on predicted RelevanceLevel. While adding CMS and actual bid to the model achieves marginal improvements, CMS filtering shows clear improvement with a 22.33% decreased RMSE value as the cut-off value increases.

F. CTR and Revenue Analysis

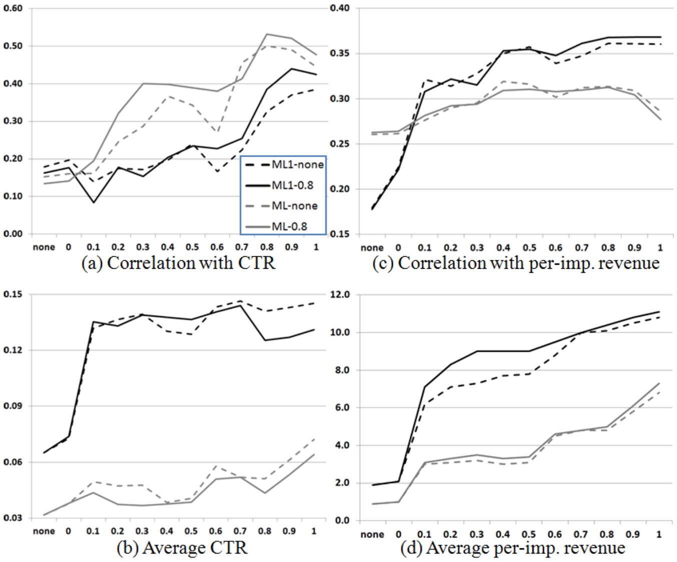
In the last experiment, we explore how much improvement we can achieve with more practical metrics, CTR and per-impression revenue, in a real service environment. As they are directly related to profit, they are considered as the two most important metrics in computational advertising. For this purpose, we use the live traffic dataset. Since we do not have machine learning features for the live traffic dataset described in Section III-A, we considered ad positions and TCS filtering instead of different feature sets. We prepared sub-datasets for the first mainline ad ($ML1$), all the mainline ads (ML), and all sidebar ads (SB) with two different TCS cut-offs: $none$, and 0.8. We also populated sub-datasets according to different CMS cut-offs but did not perform the resizing because our evaluation metrics are average values. Thus, $D_{\tau_1} \subseteq D_{\tau_2}$ holds for $\tau_2 < \tau_1$. We analyze how correlation and average value change according to these different cut-offs.

1) CTR Analysis: Fig. 3(a) shows correlation values between CMS and CTR values in various sub-datasets. The horizontal axis denotes different CMS cut-offs as in the previous figures. and vertical axis denotes Pearson's ρ values. The black and gray lines are for $ML1$ and ML, respectively. Though we also performed experiments for SB, since there are too few clicks [3] and its CTR is almost 0 (0.000326: 37 clicks out of $113,413$ impressions), we do not plot SB's result in Fig. 3. The dotted, loosely-dotted, and solid lines are for the different TCS cut-offs: $none$, 0.2, and 0.8.

We observe that as CMS cut-off increases, the correlation also increases. Though there are some outliers, this trend is quite clear for $ML1$ and ML. We also observe that the effect of TCS filtering is marginal compared to that of CMS filtering. In Fig. 3(b), we report overall average CTR value for each sub-dataset. The increasing trend holds after a big surge around the CMS cut-off of 0.1 for $ML1$. For ML, though it looks marginal due to scale, CTR increases up to two times as CMS cut-off increases. Though we cannot disclose details due to confidentiality reasons, we recently observed quite a significant CTR lift through CMS filtering in Taiwan market.

2) Revenue Analysis: Though CTR is one of the most widely used performance metrics in most online services, revenue could be more important in computational advertising because it considers ad price as well as CTR, and is more directly related to service provider's profits. Fig. 3(c) shows the correlation between CMS and per-impression revenue. The increasing trend is much clearer than that in Fig. 3(a) and TCS filtering again does not make much difference. In Fig. 3(d), we report overall per-impression revenue changes according

[3] Usually, CTR of sidebar ads is extremely low compared to that of mainline ads.



(a) Correlation with CTR

(c) Correlation with per-imp. revenue

(b) Average CTR

(d) Average per-imp. revenue

Fig. 3. CMS filtering effect on CTR and per-impression revenue

to different cut-offs. The increasing trend is extremely clear and the value increases up to 5.5 times for $ML1$. For ML, it increases up to 5 times though the value seems to increase marginally, due to its scale. As we observed so far, CMS filtering can be effectively used to select more relevant (higher CTR) and more profitable (higher per-impression revenue) query-ad pairs. We also observe that the effect of TCS filtering is very limited.

IV. Related Work

The literature in contextual advertising is mostly about how to accurately match a web page and an ad text. Compared to a succinct bid phrase in an ad, the words in a web page are more redundant and noisy. In many cases, the word in a bid phrase is different from the word in a web page even though they refer to the same thing or product. In [11], the authors introduced this *vocabulary impedance problem* and evaluated various matching strategies. Among them, incorporating words from similar pages achieved the best performance. To reuse the search advertising system for contextual advertising as well, Yih et al. [12] proposed a method to effectively extract advertising keywords from a web page. In [8], Broader et al. suggested a holistic approach of combining a syntactic match and a semantic match. They tried different combinations of the two matches and reported that the semantic match, which measures conceptual closeness, has much greater importance than the syntactic match in contextual advertising. In [4], the authors interpreted search ads as contextual ads in a SERP. Through this interpretation, they expanded the query and achieved better recall.

V. Conclusion and Future Work

In this paper, we explored the effect of CMS and actual bid in search advertising with five experiments. We first verified that CMS and actual bid are highly correlated with a human judgment on relationship between a query and an ad. We also showed that they are significantly useful in the context of machine learning. Then, we demonstrated how much improvement we can achieve with these new features in terms of judgment prediction accuracy and relevance level prediction accuracy. Lastly, we showed that CMS-based filtering is extremely effective in achieving higher CTR and per-impression revenue in a real service environment. It resulted in a significant CTR lift in Taiwan market recently.

While the improvements made by the two new features are very impressive, there remain a couple of future work items. Firstly, we did not report click prediction task results from the third experiment due to a slightly biased click distribution in our ALP dataset, even though we achieved quite significant improvement in terms of click prediction accuracy. It would be interesting to perform click prediction task with a more correctly sampled (in terms of clicks) editorial dataset and verify the relationship between judgments and clicks. Secondly, we could not apply any machine learning technique to the live traffic dataset due to the unavailability of relevance features. The result would be more interesting if we could extract relevance features for the top-1000 query-ad pairs in the live traffic dataset and perform more extensive experiments. Finally, even though CMS filtering is very effective, its coverage is limited. As this limited coverage is mainly due to short or non-commercial queries, it would be very interesting to see how much coverage can be increased with query expansion techniques described in [3], [4].

References

[1] A. Halevy, P. Norvig, and F. Pereira, “The unreasonable effectiveness of data,” *IEEE Intelligent Systems*, vol. 24, pp. 8–12, 2009.

[2] “More data usually beats better algorithms,” http://anand.typepad.com/datawocky/2008/03/more-data-usual.html.

[3] M. Sahami and T. D. Heilman, “A web-based kernel function for measuring the similarity of short text snippets,” in *World Wide Web*. ACM, 2006, pp. 377–386.

[4] A. Z. Broder, P. Ciccolo, M. Fontoura, E. Gabrilovich, V. Josifovski, and L. Riedel, “Search advertising using web relevance feedback,” in *CIKM*. ACM, 2008, pp. 1013–1022.

[5] A. Z. Broder, D. Carmel, M. Herscovici, A. Soffer, and J. Zien, “Efficient query evaluation using a two-level retrieval process,” in *CIKM*. ACM, 2003, pp. 426–434.

[6] Y. Chen, M. Gupta, and T. W. Yan, “Fast query evaluation for ad retrieval,” in *World Wide Web*. ACM, 2012, pp. 479–480.

[7] B. Edelman, M. Ostrovsky, M. Schwarz, T. D. Fudenberg, L. Kaplow, R. Lee, P. Milgrom, M. Niederle, and A. Pakes, “Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords,” *American Economic Review*, vol. 97, 2005.

[8] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel, “A semantic approach to contextual advertising,” in *SIGIR*. ACM, 2007, pp. 559–566.

[9] S. Robertson, “The Probabilistic Relevance Framework: BM25 and Beyond,” *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[10] J. Huang and C. X. Ling, “Using auc and accuracy in evaluating learning algorithms,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 299–310, 2005.

[11] B. Ribeiro-Neto, M. Cristo, P. B. Golgher, and E. Silva de Moura, “Impedance coupling in content-targeted advertising,” in *SIGIR*. ACM, 2005, pp. 496–503.

[12] W.-t. Yih, J. Goodman, and V. R. Carvalho, “Finding advertising keywords on web pages,” in *World Wide Web*. ACM, 2006, pp. 213–222.