Bidding via Clustering Ads Intentions: an Efficient Search Engine Marketing System for E-Commerce

Cheng Jie, Da Xu, Zigeng Wang, Lu Wang, Wei Shen
Walmart Labs
Sunnyvale, California, USA
{cheng.jie,da.xu,zigeng.wang0,lu.wang3,wei.shen}@walmart.com

Abstract

With the increasing scale of search engine marketing, designing an efficient bidding system is becoming paramount for the success of e-commerce companies. The critical challenges faced by a modern industrial-level bidding system include: 1. the catalog is enormous, and the relevant bidding features are of high sparsity; 2. the large volume of bidding requests induces significant computation burden to both the offline and online serving. Leveraging extraneous user-item information proves essential to mitigate the sparsity issue, for which we exploit the natural language signals from the users' query and the contextual knowledge from the products. In particular, we extract the vector representations of ads via the Transformer model and leverage their geometric relation to building collaborative bidding predictions via clustering. The two-step procedure also significantly reduces the computation stress of bid evaluation and optimization. In this paper, we introduce the end-to-end structure of the bidding system for search engine marketing for Walmart e-commerce, which successfully handles tens of millions of bids each day. We analyze the online and offline performances of our approach and discuss how we find it as a production-efficient solution.

Keywords: Clustering, Intention embedding, SEM bidding

ACM Reference Format:
Cheng Jie, Da Xu, Zigeng Wang, Lu Wang, Wei Shen. 2021. Bidding via Clustering Ads Intentions: an Efficient Search Engine Marketing System for E-Commerce. In *Proceedings of KDD (KDD International Workshop on Industrial Recommendation Systems).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD International Workshop on Industrial Recommendation Systems, Aug 14, 2021
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 Introduction

In the modern era, online advertising has become a primary channel to deliver promotional marketing messages to customers. Among the various forms of online advertising, *search engine marketing* (SEM) promotes business by showing and recommending advertisements on search-result pages. In particular, *sponsored search auctions* contribute significantly to online advertising revenue as search results often have more prominent exposure.

The business impact of SEM attracts increasing attention from both academia and industry, and the emerging challenges appeal particularly to the co-domain of economics and computer science. Over the years, a large body of literature studies the constrained bidding optimization model, which maximizes business objectives under the prefixed spending limit. For instance, [10] and [5] establish SEM bidding models for a single advertiser as constrained optimization problems in a deterministic setting where the advertisers' position, clicks, and the cost associated with a bid are known a priori. In comparison, SEM bidding as an optimization problem under the stochastic setting has been studied in [1] and [19]. Game-theoretic structures of SEM have been studied by [2] and [4], and both works aims to boost the welfare of all advertisers in search engine platforms. More recently, a new stream of work has emerged which formulate the SEM bidding optimization as a dynamic pricing problem [14, 31, 33] by incorporating the sequential behavior of SEM ads [7, 22].

While the seminal works have established rigorous mathematical properties for SEM bidding, their optimization models are often too restrictive for practical implementation. In real-world productions, the high volume of candidate ads is a crucial factor that hinders the applicability of those methods. The two challenges that stem from the industrial scale of ads are: (1). the feedback data of most SEM ads are inevitably sparse due to the limited slots at search engine platforms, preventing an accurate and effective estimation of their performances; (2). complex bidding evaluations are prohibitive because the volume of ads is enormous, and the frequency of bidding operations is very high. The previous literature addresses the sparsity issue primarily by using the ads' "keywords" in addition to the feedback data [12, 21, 27, 28]. However, using word tokens as a categorical feature can pose severe problems in building predictive models due to the high cardinality. Unlike [12], our approach constructs continuous vector representations of ads and therefore avoids the tenuous work of dealing with massive word tokens. We point out that the ideas of clustering SEM ads have also been proposed to overcome the high computation demands [6, 9, 18]. However, the clustering algorithms developed in the above work are based on the distributions of SEM ads' historical feedback data, thereby excluding those with sparse historical features, which is problematic for modern SEM applications.

SEM bidding through clustering ads intentions

In this paper, we introduce a generic bidding framework targeting the above challenges. The SEM solution proposed in our paper is currently in production for the multi-million-scale ads bidding for Walmart's e-commerce business. As we demonstrate in detail later in section 2, the solution of our system comprises of two critical components:

- a deep-learning-based multi-stage predictive algorithm for predicting the performance of the advertisement through their multi-modality signals, including the user feedback data and the contextual features of ads;
- an optimization algorithm that assigns a bidding price for each ad, based on its performance forecast for the desired business objectives.

Toward our goals, we first redesign a Transformer-based [24] deep-learning language model to extract vector representations of the ads, which captures the customer's intentions when landing on the ads page. The advantage is that we can now fully leverage the geometric characteristics of the representations to aggregate ads' information that would be sparse otherwise (detailed in Section 3). The multi-stage prediction algorithm, which then augments the grouping patterns of features via ads clustering, further alleviates the sparsity issue of the features. In the meantime, the clustering-based solution improves the scalability of the second-stage optimization algorithm by significantly reducing the number of entities in the downstream evaluation of the bids.

We thoroughly examine the performance of the proposed solutions via both offline studies and online experiments in Section 5. As expected, the clustering step is essential for trading off the sparsity, accuracy, and scalability.

2 Background for SEM Bidding

We first introduce the underlying bidding model and system that powers Walmart's SEM business.

2.1 SEM bidding model

We address the SEM bidding problem via *ad groups*, where the detailed process of generating the ad groups are deferred to Section 3. Suppose an advertiser aims to maximize its revenue as its business objective under budget B. Denote $\mathcal{G}$ as the set of ad groups for the advertiser. The SEM bidding model on an ad group $g \in \mathcal{G}$ can be given by:

$$\max_{\{b_g\}} E\Big[\sum_{g\in\mathcal{G}} R_g(b_g)\Big] \text{ s.t. } E\Big[\sum_{g\in\mathcal{G}} S_g(b_g)\Big] \leq B, \qquad (1)$$

where b_g is the bidding value assigned to the ads at group g, $R_g(\cdot)$ and $S_g(\cdot)$ are the corresponding *revenue* and *spend* function.

Directly solving (1) is impractical since the expected revenue $E(R_g(b_g))$ and expected spending $E(S_g(b_g))$ can be very complicated [10, 17]. However, by adding certain practical assumptions on $E(R_g(b_g))$ and $E(S_g(b_g))$, the optimum of (1) can be found quite efficiently. We denote the *expected click* for the bid value of b as $E[C_g(b)]$, and introduce the notions of RPS (*revenue per spend*) and RPC (*revenue per click*) below.

Definition 1. *The RPS, i.e, revenue per spend (revenue of an ad per unit of spend), equals:* $RPS_g = \frac{E[R_g(b_g)]}{E[S_g(b_g)]}$ *given an ad group g. The RPC, i.e, revenue per click, equals* $RPC_g = \frac{E[R_g(b_g)]}{E[C_g(b_g)]}$ *for a given ad g.*

We now state the critical assumption.

Assumption 1. *For a given ad group g, its revenue per click* RPC_g *is invariant to the change of bid value* b_g*. Furthermore, we suppose* $E[C_g(b_g)] = c_g \cdot b_g$ *for a given constant of* c_g*. When the search engine uses the first-price auction [23], we have* $E[C_g(b_g)] = c_g \cdot b_g^2$ *as a result.*

Under Assumption 1, we have the following key result:

Theorem 1. *The optimal solution of the optimization problem in (1) is achieved when the RPS (revenue per investment) is the same for all* $g \in \mathcal{G}$*.*

Proof. The Lagrangian of (1) is given by:

$$\mathcal{L} = E\Big[\sum_{g\in\mathcal{G}} R_g(b_g)\Big] - \lambda\{B - E\Big[\sum_{g\in\mathcal{G}} S_g(b_g)\Big]\}, \qquad (2)$$

The KKT condition for gradient of 1 is:

$$\forall g: \frac{d\mathcal{L}}{db_g} = \frac{d}{db_g}E\Big[R_g(b_g)\Big] - \lambda\frac{d}{db_g}E\Big[S_g(b_g)\Big] = 0, \lambda >= 0$$

since $R_g(b_g)$ and $S_g(b_g)$ are independent from other ad groups. KKT condition of 2 implies that an optimal solution exists when:

$$\frac{d}{db_g}E\Big[R_g(b_g)\Big] / \frac{d}{db_g}E\Big[S_g(b_g)\Big]$$

takes the same value across $g \in \mathcal{G}$. Under assumption 1, we immediately have:

$$E\Big[R_g(b_g)\Big] = c_g b_g RPC_g \text{ and } E\Big[S_g(b_g)\Big] = c_g b_g^2$$

$$\frac{d}{db_g}E\Big[R_g(b_g)\Big] / \frac{d}{db_g}E\Big[S_g(b_g)\Big] = \frac{RPC_g}{2b_g} \qquad (3)$$

and the revenue per spend is: $RPS_g = \frac{E[R_g(b_g)]}{E[C_g(b_g)]} = \frac{RPC_g}{b_g}$. Hence, the KKT condition of 2 is equivalent to saying that RPS_g are equal across $g \in \mathcal{G}$. □

Recall that Assumption 1 claims that RPC_g is steady against b_g. Therefore, Theorem 1 implies that as long as we have an accurate prediction of RPC_gs for each group, the optimal condition in (1) can be easily achieved by setting the bids b_g such that RPC_g/b_g are equal across all ad groups g.

Remark 1. *Note that the classical singular-ad bidding algorithm can be easily recovered by replacing the ad group g with the single ad.*

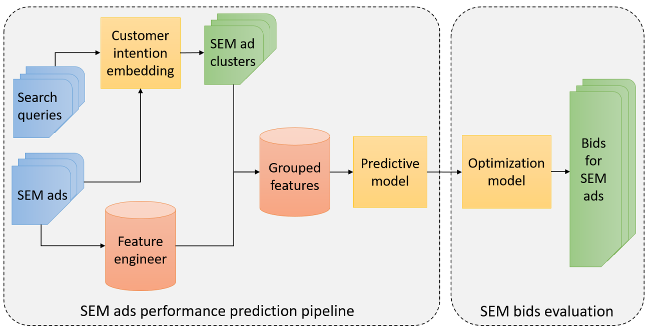

Figure 1. Overview of the infrastructure for providing SEM ads' bids by our approach.

2.2 SEM bidding system

The results in the previous section suggest that the critical tasks for determining the bids of SEM ads is to accurately predict the revenue per click for each ad group g. In the sequel, we propose a design of the SEM ads bidding system illustrated in Fig 1. In Figure 1, the first task for obtaining the *RPC* predictions is clustering the pool of SEM ads into ad groups. It consists of two steps: 1. building a representation learning model that encodes SEM ads into embeddings; 2. clustering SEM ads into ad groups. After creating the SEM ad groups, the system will aggregate the features for ads within each ad group, and then train a predictive model to accurately forecast the RPC_g for each ad group. We plug the RPC_g back to the optimization problem and obtain the final bidding b_g for each $g \in \mathcal{G}$ as $b_g = RPC_g/RPS_g$, where RPS_g is known in advance.

3 Embedding and Clustering of SEM Ads

The ad-group level bidding in (1) performs the best when each ad cluster is dedicated to a specific user intention. For this purpose, we segment the SEM ads into mutually exclusive ad clusters in terms of customer intention by two steps. Firstly, We build the customer-intention representation model which provides an embedding for each ad. Secondly, based on the embeddings, we develop a multi-stage clustering method that groups the massive ads into small to mid-sized groups.

3.1 Customer intention embedding model

The customer intention of an SEM ad is defined as the integrated purchase intention (of the set of search queries) that leads to the clicked ads on the search engine. For example, an ad may appeal to customers who search for "apple phone 8 case" or "iphone 9 case", if their intentions are the case covers for various versions of iphone. If two ads share a large portion of clicked search queries, their customer intentions should be close to each other. Therefore, we design the customer intention model to reflect the co-click relations among the SEM ads. We propose the following metric to capture such intention.

Interactive metric. The interactive metric (I) is designed to calibrate the similarity between customer intentions of two SEM ads. Given two SEM ads A1 and A2, we first obtain the numbers of co-clicks of the two ads and denote them as $C_{(A1coA2)}$ and $C_{(A2coA1)}$. Given the numbers of total historical clicks of the two ads C_{A1} and C_{A2}, the metric value for A1 and A2 is defined via:

$$I_{A1,A2} = \sqrt{\frac{C_{(A1coA2)} * C_{(A2coA1)}}{C_{A1} * C_{A2}}}, \tag{4}$$

See figure 2 for an illustration of interactive metric in a real-world example, which effectively discounts the popularity and exposure bias.

Contextual features of SEM ad. When a search query appears, the search engine will try to match it with the SEM ads according to the content of their landing pages. In light of that, we select the ads' website's text content as the main feature for the customer intention model, since the content should be a critical factor in customers' decision making. The text feature of an SEM ad is a combination of titles and descriptions of products contained in the ad's website. For the SEM ads with more than one product, we choose the three top products to constrain the length of the input feature. Once the features are extracted, they are processed and converted through the standard tokenization and padding procedures described in [8].

Transformer-based customer intention representation model. Recently, the attention-based encode-decode structure transformer has become the status quo architecture for natural language processing tasks [24, 29]. Motivated by the structure of the bidirectional transformer from [8], we built a transformer-based deep learning model for extracting the customer intention from the text features of SEM ads. As we show in Fig 3, for a given ad A and its tokenized feature T_A, the model will consecutively go through an initial embedding layer, 3 transformer layers, a dense pooling layer, and two feedforward layers before generating the final 512-dimension *normalized* output vector.

Training data. The data we use for training the representation learning model is the `search_term_report` from search engine, which provides the historical statistics of interactions (e.g. clicks, impressions) between SEM ads and their relevant search queries. Specifically, for each SEM ad, we will extract historical click numbers between the ad and each search query that leads to the clicks. Together with interactive metric I defined at 4, we create a data-set $\mathcal{D}$ containing all the tuples of SEM ads having co-clicked queries together with their interactive metric. In addition to the above positive instances, we need negative instances to cover larger support of the distribution. For that purpose, we sample a certain number of ad tuples without co-clicked queries and append the tuples onto the data-set $\mathcal{D}$ by assigning them with an interactive metric value of -1. For the best practice, the ratio between positive tuples and negative tuples should approximately equal to the average positive interactive metric in the feedback data.

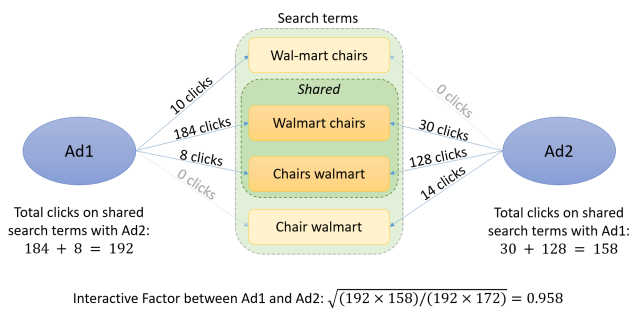

Figure 2. Interactive metric: an example

Model training. Let $f_\theta(\cdot)$ denote a customer intention model with parameter vector θ. Given an ad tuple (A_i, A_j) along with their interactive metric I_{ij}, we define the loss function as

$$-I_{ij} \log \sigma\big(f_\theta(T_{A_i})^T f_\theta(T_{A_j})\big), \quad (5)$$

where $\sigma(\cdot)$ is the sigmoid function. The inner product of $f_\theta(T_{A_i})^T f_\theta(T_{A_j})$ captures the cosine similarity between the embeddings of (A_i, A_j), given that output vectors of the model $f_\theta(\cdot)$ are normalized. The structure of the model, together with the procedure for calculating the loss function, are presented in Figure 3. The optimization problem for finding the optimal θ is now given by:

$$\theta^\star = \arg\min_{\theta \in \Theta} \sum_{(A_i, A_j) \in DT} -I_{ij} \log \sigma\big(f_\theta(A_i)^T f_\theta(A_j)\big), \quad (6)$$

The objective (6) indicates that the larger the interactive metric between two ads, the more impact this ad instance will carry when determining model parameter θ. Including the negative instances will allow the model to further separate ads that lack a shared customer intention. Moreover, using negative samples can avoid over-fitting and the corner case where all SEM ads having a similar embedding. We use the ADAM[16] optimizer, a variant of stochastic optimization [15], [11] for training (6).

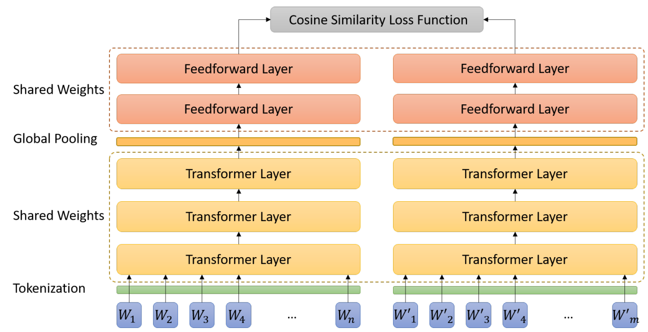

Figure 3. SEM ads customer intention embedding model

3.2 Multi-stage SEM ads clustering algorithm

In what follows, we discuss clustering with ads embedding. Due to the high volume of ads in modern SEM, though many efficient machine learn models have been introduced [13, 20, 25, 26, 32], it is still impractical to apply the clustering algorithms that require computing all the pair-wise distances [**?**]. Here, we present a multi-stage method that leverages the SEM ads' taxonomy and significantly reduces the computation demand.

SEM ads classification. The first step of the multi-stage clustering algorithm is to classify each SEM ad into one of the *product types*, which can be any taxonomy that is labeled for the items: electronics, beverage, etc. Most companies have the predefined taxonomy for each item, which should be actively exploited. SEM ad with only one item can be directly concluded to its product type, and serve as the training sample of the taxonomy classification model. For SEM ads with more than one item, we train a feedforward neural network to predict each ads' product type, which takes the embedding of the SEM ad as input.

Clustering within each product type. Following the classification, we apply the "bottom-up" Agglomerative clustering using embedding vectors as features to create mutually exclusive ad groups for the SEM ads within each product type. Naturally, the cosine distance is employed as the *linkage metric*, and it also allows us to determine the threshold based on which the final clusters are formed. We point out that the first classification step significantly reduces the computation complexity compared with directly clustering all the ads.

4 Predicting RPC for SEM ads

In the next step, we build a machine learning model for each ads cluster to predict the key quantity of RPC_g, i.e. the revenue per click, whose role was illustrated in Section 2.2.

Features. The features we use for predicting *RPC* can be categorized into three classes: 1. the historical feedback statistics such as clicks and conversions; 2. the activity metric for the ad's landing pages such as bounce rate; 3. contextual features of the ad, such as aggregating the ads embedding for each ad group.

Model selection. There are varieties of machine learning models[30] for predicting *RPC*. Due to space limitations, we do not dive into their pros and cons. In section 5, we will show the performances of linear regression, TabNet [3] and Gradient Boosting regression tree.

Model training We choose the clicks-weighted square error as the loss function for model training because ad groups with higher clicks often have more impact on the business. Formally, by denoting the parameter of the model by $\eta \in \mathcal{H}$ and the total clicks of ad group by C_g, the objective function is given by:

$$\eta^\star = \underset{\eta \in \mathcal{H}}{\arg\min} \frac{\sum_{g \in \mathcal{G}} C_g (r_\eta(X_g) - RPC_g)^2}{\sum_{g \in \mathcal{G}} C_g}, \tag{7}$$

where r_η is the RPC predicitive model.

5 Experiments and Analysis

We conducted both offline and online experiments to answer the following questions:

Q1: Can ads clustering improve the RPC prediction accuracy by addressing the spareness of feedback data?

Q2: Does the proposed two-step framework improve the business performance?

Offline experiment: prediction accuracy comparison

The offline experiment is designed to test whether the proposed clustering methods address the sparseness issue and improve RPC prediction accuracy. To this end, we select a set of ads with a total number of ~20 million, and compare the accuracy of RPC predictions of 1. directly applying RPC prediction on each SEM ads (the baseline singular-ad-based algorithm); 2. clustering SEM ads before predicting RPC for each ad cluster (our cluster-based bidding algorithm). For fair comparison, we evaluate the performance metric based on each ad and set the predicted RPC of each ad equivalent to the predicted RPC of its belonging ad cluster when using the second approach. According to the operation protocol of Walmart, we predict the weekly RPC as described in Section 4. For the proposed approach, we apply the methods introduced in Section 3 to cluster SEM ads into ad groups, and aggregate the ad features within each ad group. The summary statistics for the ad groups and the original SEM ads are displayed in Table 1.

In table 1, the proportion of feature missingness is calculated based on the non-contextual features, and due to Walmart's privacy policy, the variances of the RPC response variable are presented as percentage proportions to the largest among the two datasets. Table 1 manifest the two benefits of ads clustering: 1. the feature sparseness is dramatically improved as exemplified by the reduced missing feature proportion, 2. the reduced variance of the response variable indicates that the clustering algorithm tends to produce a more robust output for the downstream *RPG* modeling.

Table 1. SEM ads vs ad groups: Data-set overview

	SEM ads	SEM ad groups
Dataset sample size	19.6 M	1.8M
Missing feature (proportion)	91.6%	54.4%
Non-empty response ratio	6.7 %	36.4 %
Relative response RPC variance	100%	55%

We experimented with three machine learning models for predicting the weekly RPC: *linear regression* (LR) model, *TabNet* and *gradient boosting regression tree* (GBRT). We split the dataset into training, validation, and test by $80\% - 10\% - 10\%$, where the test dataset is used to report the predictive accuracy of the trained models. In addition to the click-weighted MSE (WMSE) mentioned at section 4, we also include the click-weighted MAE (WMAE) as performance metric. The performances of the trained models are displayed on table 2. Figure 4 presents an example of the gradient boosting trees when applied to the baseline and our approach, under their best hyper-parameter combinations. Due to the privacy policy, we provide the accuracy metric with respect to the baseline model, which is Linear regression(LR) on the singular-ad-based algorithm. The model training, including hyper-parameter tuning is conducted on a Linux system with 64 core 2.80GHz CPUs and 800 GB memory.

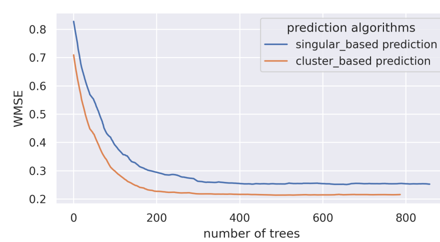

Figure 4. Relative WMSE of the baseline and our approach when using GBRT.

The results from Table 2 and Figure 4 suggest that RPC prediction via ad clustering consistently achieves better performances compared with the singular ad prediction. Further, the computational time for training RPC at the cluster level is considerably less than the singular-ad level.

Online experiment: business efficiency comparison

We design the online AB testing experiment to see whether the clustering-based bidding improves business performance,

Table 2. The RPC prediction accuracy (relative to LR on singular-ad setting), and the offline model training time.

Predictive model	WMSE(Relative to LR Singular)		WMAE(Relative to LR Singular)		Training time	
	Singular based	Cluster based	Singular based	Cluster based	Singular based	Cluster based
LR(reference point)	100%	92%	100%	86%	8m	2m
TabNet	30%	27%	35%	31%	15h	2.5h
Gradient boosting	25%	21%	29%	23%	16h	2h

Table 3. Online AB testing results.

Test	Metric(Relative to control)	Control	Test
AA period	Spend	100%	101%
	RPS	100%	99%
AB period	Spend	100%	102%
	RPS	100%	109%

which is mainly reflected by the revenue per spend (RPS). We use the stratified sampling to select 200,000 SEM ads across different product types as our target ads pool, and compare the RPS of clustering-based bidding algorithm and traditional singular-ad-based bidding algorithm when applied on the selected SEM ads.

Here, we leverage the Draft & Experiment platform from Google Adwords[1] to create a pair of control and test campaigns that host the 200k SEM ads. The singular-based and clustering-based bidding algorithms are applied to the control and test campaign, respectively. We set a common RPS goal for the two algorithms to evaluate bids.

The control and test campaigns are launched simultaneously, and during the test, Google evenly split incoming traffic to ensure a fair comparison. The experiment session consists of one week of AA test, and one following week of AB test. During the *AB* period, we keep the spending between control and test campaigns close through some proportional bids adjustments.

The online testing results are presented in table 3, where we present metrics relative to the control campaign. The test campaign's RPS on the AB period exemplifies that clustering-based bidding is able to achieve improved performance compared with the singular-ad-based-bidding. To further justify our conclusion, we perform a paired *t test* on the RPS of two campaigns, which shows that our experiment reached a t-statistics of 2.1 with the p-value of 0.02.

6 Conclusion

This paper introduces a two-step clustering-based SEM bidding system that integrates modern representation learning with the Transformer language model. We describe the detailed development infrastructure that may bring insights to both practitioners and researchers in this domain. The offline and online experiments show that the proposed system compares favorably to the alternatives in terms of accuracy and training efficiency. Our successful deployment for Walmart e-commerce further reveals combining clustering with modern representation learning as a scalable solution for industrial bidding systems.

[1]https://ads.google.com, where the max capacity for a campaign is 200,000.

References

[1] V. Abhishek and K. Hosanagar. 2012. Optimal bidding in multi-item multi-slot sponsored search auctions. In *EC '12*.

[2] Gagan Aggarwal, S. Muthukrishnan, D. Pál, and Martin Pál. 2009. General auction mechanism for search advertising. In *WWW '09*.

[3] Sercan O Arik and Tomas Pfister. 2019. Tabnet: Attentive interpretable tabular learning. *arXiv preprint arXiv:1908.07442* (2019).

[4] Tilman Borgers, I. Cox, Martin Pesendorfer, and V. Petrivcek. 2008. Equilibrium Bids in Sponsored Search Auctions: Theory and Evidence.

[5] C. Borgs, J. Chayes, Nicole Immorlica, K. Jain, O. Etesami, and Mohammad Mahdian. 2007. Dynamics of bid optimization in online advertisement auctions. In *WWW '07*.

[6] Y. Chen, W. Liu, J. Yi, Anton Schwaighofer, and T. W. Yan. 2013. Query clustering based on bid landscape for sponsored search auction optimization. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013).

[7] Savas Dayanik and M. Parlar. 2013. Dynamic bidding strategies in search-based advertising. *Annals of Operations Research* 211 (2013), 103–136.

[8] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[9] Tianming Du, Yanci Zhang, Xiaotong Shi, and Shuang Chen. 2020. Multiple Slice k-space Deep Learning for Magnetic Resonance Imaging Reconstruction. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*. 1564–1567. https://doi.org/10.1109/EMBC44109.2020.9175642

[10] J. Feldman, S. Muthukrishnan, Martin Pál, and C. Stein. 2007. Budget optimization in search-based advertising auctions. *ArXiv* abs/cs/0612052 (2007).

[11] Yuzhen Han, Majid Mazouchi, S. Nageshrao, and H. Modares. 2021. A Convex Programming Approach to Data-Driven Risk-Averse Reinforcement Learning. *ArXiv* abs/2103.14606 (2021).

[12] D. Hillard, Eren Manavoglu, H. Raghavan, C. Leggetter, E. Cantú-Paz, and R. Iyer. 2010. The sum of its parts: reducing sparsity in click estimation with query segments. *Information Retrieval* 14 (2010), 315–336.

[13] Robert A. Jarrow, Rinald Murataj, Martin T. Wells, and Liao Zhu. 2021. The Low-volatility Anomaly and the Adaptive Multi-Factor Model. *arXiv preprint arXiv:2003.08302* (2021).

[14] Cheng Jie. 2018. *Decision Making Under Uncertainty: New Models and Applications.* Ph.D. Dissertation.

[15] Cheng Jie, A. PrashanthL., M. Fu, S. Marcus, and Csaba Szepesvari. 2018. Stochastic Optimization in a Cumulative Prospect Theory Framework. *IEEE Trans. Automat. Control* 63 (2018), 2867–2882.

[16] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2015).

[17] Kun Lin, Cheng Jie, and Steven I. Marcus. 2018. Probabilistically distorted risk-sensitive infinite-horizon dynamic programming. *Automatica* 97 (2018), 1–6. https://doi.org/10.1016/j.automatica.2018.07.028

[18] Mohammad Mahdian and Grant Wang. 2009. Clustering-Based Bidding Languages for Sponsored Search. In *ESA*.

[19] Furcy Pin and P. Key. 2011. Stochastic variability in sponsored search auctions: observations and models. In *EC '11*.

[20] A. PrashanthL., Cheng Jie, M. Fu, S. Marcus, and Csaba Szepesvari. 2016. Cumulative Prospect Theory Meets Reinforcement Learning: Prediction and Control. In *ICML*.

[21] Elías Segundo. 2016. Predict the Click-Through Rate and Average Cost Per Click for Keywords Using Machine Learning Methodologies.

[22] W. Shen, Binghui Peng, Hanpeng Liu, Michael Zhang, Ruohan Qian, Y. Hong, Z. Guo, Zongyao Ding, Pengjun Lu, and Pingzhong Tang. 2020. Reinforcement Mechanism Design, with Applications to Dynamic Pricing in Sponsored Search Auctions. *ArXiv* abs/1711.10279 (2020).

[23] Theodore L Turocy, Elizabeth Watson, and Raymond C Battalio. 2007. Framing the first-price auction. *Experimental Economics* 10, 1 (2007), 37–51.

[24] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. *ArXiv* abs/1706.03762 (2017).

[25] Zigeng Wang, Xia Xiao, and Sanguthevar Rajasekaran. 2020. Novel and efficient randomized algorithms for feature selection. *Big Data Mining and Analytics* 3, 3 (2020), 208–224.

[26] Xia Xiao, Zigeng Wang, and Sanguthevar Rajasekaran. 2019. Autoprune: Automatic network pruning by regularizing auxiliary parameters. *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)* 32 (2019).

[27] Da Xu, Chuanwei Ruan, Jason Cho, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Knowledge-aware complementary product representation learning. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 681–689.

[28] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Product knowledge graph embedding for e-commerce. In *Proceedings of the 13th international conference on web search and data mining*. 672–680.

[29] Yanci Zhang, Tianming Du, Yujie Sun, Lawrence Donohue, and Rui Dai. 2021. Form 10-Q Itemization. *ArXiv* abs/2104.11783 (2021).

[30] Xinyan Zhao, M. Zhan, and Cheng Jie. 2018. Examining multiplicity and dynamics of publics' crisis narratives with large-scale Twitter data. *Public Relations Review* 44 (2018), 619–632.

[31] Liao Zhu. 2020. *The Adaptive Multi-Factor Model and the Financial Market*. eCommons.

[32] Liao Zhu, Sumanta Basu, Robert A. Jarrow, and Martin T. Wells. 2020. High-Dimensional Estimation, Basis Assets, and the Adaptive Multi-Factor Model. *The Quarterly Journal of Finance* 10, 04 (2020), 2050017.

[33] Liao Zhu, Haoxuan Wu, and Martin T. Wells. 2021. A News-based Machine Learning Model for Adaptive Asset Pricing. *arXiv preprint arXiv:2106.07103* (2021).