A Language Model Approach to Capture Commercial Intent and Information Relevance for Sponsored Search

Lei Wang
Yahoo! Global R&D Center, Beijing
Beijing, 100083, China
wanglei@yahoo-inc.com

Mingjiang Ye
Yahoo! Global R&D Center, Beijing
Beijing, 100083, China
mye@yahoo-inc.com

Yu Zou
Yahoo! Labs
Beijing, 100083, China
yuzou@yahoo-inc.com

ABSTRACT

A fundamental task of sponsored search is how to find the best match between web search queries and textual advertisements. To address this problem, we explicitly characterize the criteria for an advertisement to be a 'good match' to a query from two aspects (it should be relevant with the query from information perspective, and it should be able to capture and satisfy the commercial intent in the query). Correspondingly, we introduce in this paper a mixture language model of two parts: a commercial model which characterizes language bias of commercial intent leveraging on users' clicks on advertisements, and an informational model which is a traditional language model with consideration of the entropy of each word to capture informational relevance. We then introduce a regularized expectation-maximization (EM) algorithm model for parameters estimation, and integrate query commercial intent into the scoring function to boost overall click efficiency.

Empirical evaluation shows that our model achieves better performance as compared to a well tuned classical language model and deliberated TFIDF-pLSI model (6% and 5% precision improvement at our operating point in production environment of 30% recall, and 5.3% and 6.3% AUC improvement), and performs superior to the KL Divergence language model for tail queries (0.5% nDCG improvement). Live traffic test shows over 2% CTR lift and 2.5% RPS lift as well.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Measurement, Performance, Experimentation

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

Keywords

Online Advertising, Sponsored Search, Language Model, Commercial Intent, Relevance

1. INTRODUCTION

Sponsored search has become a $\$10B+$ business [15] in recent years. In sponsored search, most often, the advertisements are selected by 'exact match' between queries and bid phrases. This kind of intuitive methods, though are effective in general situations, do have some obvious drawbacks, such as low coverage for tail queries and not robust in precision. On the other hands, language model, a prominent IR model, though has been successfully applied to many problems and has gained superior performance, faces two challenges when applying to sponsored search: 1) *how to characterize and capture commercial intent.* In sponsored search, besides informational relevance, an equally, may be more, important objective is to identify and serve users' commercial intent which may bring in further commercial activities, and 2) *how to estimate model parameters reliably.* Advertisements are relatively short, usually just tens of words (in our work we do not take landing pages into consideration because they are usually very noisy and are not available all the time), not sufficient for model estimation.

Firstly we regard an advertisement as a 'good match' to a user query if it satisfies two criteria: 1) it is relevant with the query from information perspective, and 2) it captures and satisfies the commercial intent (if there is any) in the query. Correspondingly, from the demand perspective, we explicitly model users' intents in raising a query from two aspects, informational intent and commercial intent. For example when a user has an interest on iPad2. He may raise a query like 'iPad2 with Wi-Fi 3G' to learn some informational knowledge about features such as Wi-Fi and 3G, as well as to learn some commercial characteristics like the price and how to buy, and then leveraging on these information he may make a decision about whether to take further commercial activities.

Accordingly, from the supply perspective, a 'clever' advertiser is expected to deliberately organize his advertisements to satisfy both of the two kinds of intents concurrently to maximize the attraction of the advertisements. In this sense, we design our language model as a mixture model of an informational model and a commercial model, in which the informational model is a traditional language model with consideration of the entropy of each word for informational relevance, while the commercial model characterizes language bias of commercial intent leveraging on users' clicks

on advertisements. Further more, we explicitly introduce query commercial propensity into the scoring function to boost commercial queries.

Secondly, we leverage on both local and global information to try to estimate model parameters more reliably. We first utilize global information such as query set and click logs to compute the global word importance in expressing informational intent and commercial intent respectively, and then incorporate them with local information such as term frequency to estimate the informational model and commercial model. We then introduce a regularized EM algorithm to refine the parameter estimations together with the model selection probability estimation, which integrates the global and local information better. Empirical evaluation and live traffic performance confirms the effectiveness of our model.

2. MIXTURE LANGUAGE MODEL

As illustrated in formula (1), in classical language model, we generally use a model θ which is a distribution over words to characterize a people's language habit. For documents retrieval tasks, we usually regard a document D as a sample from θ, and then use this sample to estimate a model θ_D to approximate θ (the V in formula (1) is the vocabulary of the language of the document D). Maximum likelihood estimation (MLE) is widely used in the estimation, according to which $P(w|D)$ is equal to the relative term frequency of w in D as in formula (2). We then assume that the query is generated from some language model as well, and in this sense the likelihood that the query Q is generated from θ_D can be used to score D.

$$\theta \simeq \theta_D = \{P(w|D)\}_{w \in V} \tag{1}$$

$$P(w|D) = \frac{tf(w,D)}{\sum_w tf(w,D)} \tag{2}$$

For sponsored search, we assume that a person has different language biases in expressing commercial intent and informational intent. The generation procedure for each query word then includes two steps: the user first implicitly selects a language model corresponding to his instant intent (commercial or informational), and then select a word within the selected language model. With these assumptions, we define our language model θ (or θ_D in estimation) as a Bernoulli mixture model of two multinomial models, a commercial language model θ_C and an informational language model θ_I:

$$\theta_D \simeq \theta = Bern(\theta_C, \theta_I) \tag{3}$$

In the more precise form in formula (4), α_i is the model selection probability with constrain that $\sum_{i \in \{I,C\}} \alpha_i = 1$.

$$P(w|\theta) = \sum_{i \in \{I,C\}} \alpha_i P(w|\theta_i) \tag{4}$$

The model parameters estimation and model selection probability estimation appear in the following sub sections.

2.1 Commercial Model

We use the commercial model θ_C to characterize an advertiser's language bias when he expresses commercial intent. Given a document D, intuitively θ_C can be estimated by regarding the commercial portion of D as a sample, which can be formulated as formula (5), in which C denotes '*the person is under commercial intent*' or can be equally defined as '*the commercial model θ_C is selected*', so $P(w|C,D)$ defines the words distribution of the commercial portion of D.

$$\theta_C = \{P(w|C,D)\}_{w \in V} \tag{5}$$

With this definition, challenge comes soon that it is hard to tell the commercial portion of a document from other portions, and thus we can't use MLE to estimate the model directly. Instead, we transform the formula using Bayesian method as formula (6), in which the task is transformed to compute $P(w,C|D)$.

$$P(w|C,D) = \frac{P(w,C|D)}{P(C|D)} = \frac{P(w,C|D)}{\sum_w P(w,C|D)} \tag{6}$$

Our lines of thinking to estimate $P(w,C|D)$ originate from two perspectives, global context perspective and local context perspective. We heuristically compute two estimations denoted as $P_g(w,C|D)$ and $P_l(w,C|D)$ from the two perspectives respectively, and then linearly combine them as the final estimation of $P(w,C|D)$.

2.1.1 Global Context Perspective

We estimate $P_g(w,C|D)$ as formula (7), in which $P(w|D)$ can be estimated using MLE as in formula (2), and $P(C|w,D)$ refers to '*given D as a prior sample, and suppose that a person raises a word w, the probability that w derives from the commercial intent of the person*'. As illustrated in formula (8), we make an independency assumption here that the posterior probability of a person's commercial intent given a word he has raised is a global characteristic of the word (an example of global characteristic is IDF) determined by the global context such as global query set or click feedback set, while has no relationship with the local context in which the word exists (here the local context refers to the document D). In this way, the problem is transformed to compute $P(C|w)$.

$$P_g(w,C|D) = P(w|D)P(C|w,D) \tag{7}$$

$$P(C|w,D) \simeq P(C|w) \tag{8}$$

We refer to Sandeep's work in estimating advertisability of tail queries [11] to estimate $P(C|w)$ utilizing click feedback. We make two assumptions: 1) $P(C|w)$ is uniform for (independent of) users and advertisers, and 2) each instance of click or not click on advertisements served for a query is a referendum of the commercial intent of each word in the query independently. The computation method is as formula (9), in which $c(q)$ indicates the number of times users click some advertisements after raising the query q, while $n(q)$ indicates the number of times users raise q without clicking any advertisements.

$$P(C|w) = \frac{\sum_q^{w \in q} c(q)}{\sum_q^{w \in q} (c(q) + n(q))} \tag{9}$$

2.1.2 Local Context Perspective

Another thinking line for this problem is to introduce topics as mediator as follows.

$$P_l(w,C|D) = \sum_i P(w|t_i)P(C|t_i,w)P(t_i|D) \tag{10}$$

In formula (10), $P(w|t_i)$ and $P(t_i|D)$ denotes words distribution on a topic and topics distribution on a document

respectively. Both of them can be solved by pLSI [8][7] or LDA [2] etc., and $P(C|t_i, w)$ refers to '*suppose a person issues a word w to express a topic t_i, the probability that w derives from the commercial intent of the person*'. As in formula (11), we make an independency assumption here that the posterior probability of a person's commercial intent given the word he has raised is totally decided by the local context in which the word occurs, that is the topic the person is to express using the word, but has no relationship with the global context. In this way, the problem is transformed to compute $P(C|t_i)$, which boils down to compute $P(t_i|C)$ and $P(t_i)$.

$$P(C|t_i, w) \simeq P(C|t_i) \propto \frac{P(t_i|C)}{P(t_i)} \tag{11}$$

In our implementation, we first train 1000 topics utilizing pLSI (we sample 100 million queries no matter commercial or not from an industrial search engine's query logs in one month, and use the corpus of the top 5 search result pages for each query to train the topics), and then leverage on fold-in algorithm [3] to compute $P(t_i|C)$ on a set of click-through data of query-advertisement pairs in three months and compute $P(t_i)$ on the corpus of all advertisements of the three months. Here we regard $(P(t_i|C))$ as the topics distribution on the corpus of clicked advertisements, in which an advertisement clicked i times will be counted $1 + \log i$ times in occurrence to weight its importance, while regard $P(t_i)$ as the topics distribution on corpus of all advertisements no matter shown or not.

2.2 Informational Model

Besides commercial intent, we want to capture the ordinary 'relevance' as well. Generally this can be achieved by a traditional language model such as [12]. In our work we refine the traditional language model to explicitly characterize 'relevance' from the informational perspective. We use the informational model θ_I to characterize an advertiser's language bias when he expresses informational intent. Similar with θ_C, intuitively θ_I can be estimated by regarding the informational portion of D as a sample, which can be formulated as formula (12), in which I denotes '*the person is under informational intent*' or can be equally defined as '*the informational model θ_I is selected*', and thus $P(w|I, D)$ defines the words distribution of the informational portion of D.

$$\theta_I = \{P(w|I, D)\}_{w \in V} \tag{12}$$

We make an assumption that the relative frequency of a word in the informational portion of a document can be approximated by the proportion of the quantity of information from information theory perspective, and thus we define $P(w|I, D)$ as the ratio of information content of word w in D towards the sum of information content of all words in D. In formula (13), $ic(w)$ denotes the information content of w , which equals to the term frequency of w in D (that is $tf(w, D)$) multiplies the self information of w (that is $-\log P(w)$). We first estimate $P(w)$ using MLE on a corpus of 100 million user queries sampled from an industrial search engine's query logs, and then use Bayesian smoothing to smooth it with the one estimated on a larger archive of web pages of the search engine.

$$P(w|I, D) = \frac{ic(w)}{\sum_w ic(w)} = \frac{tf(w, D) \log P(w)}{\sum_w tf(w, D) \log P(w)} \tag{13}$$

2.3 Model Selection Probability Estimation and Parameters Refinement

In this section we present a regularized EM algorithm to estimate the model selection probability and to refine the model parameters estimated in the above two subsections in the mean time.

In classical EM algorithm [5][10], we generally denote the log likelihood of a document D to be generated from a mixture model as formula (14), in which α_j denotes the model selection probability. In order to maximize $L(D)$, in E step, EM algorithm estimates the expectation of the complete likelihood denoted as $Q(\psi; \psi^o)$ in formula (15), in which ψ denotes all parameters while ψ^o denotes values of ψ estimated in the last iteration, and $z(w, j)$ here is a hidden variable which denotes the probability that w is generated from model θ_j and boils down to the form in formula (16), and $\mu(1 - \sum_j \alpha_j)$ is the Lagrange Multiplier corresponding to the constrain that $\sum_j \alpha_j = 1$. Then in M step, EM algorithm finds the values of ψ which maximize $Q(\psi; \psi^o)$, that is to find $\psi^{(o+1)} = \underset{\psi}{\text{argmax}} Q(\psi; \psi^o)$.

$$L(D) = \sum_w tf(w, D) \log \sum_j \alpha_j P(w|\theta_j) \tag{14}$$

$$\begin{aligned} Q(\psi; \psi^o) = &\sum_w tf(w, D) \sum_j z(w, j) \log(\alpha_j P(w|\theta_j)) \\ &+ \mu(1 - \sum_j \alpha_j) \end{aligned} \tag{15}$$

$$z(w, j) = P(\theta_j|w, \psi^o) = \frac{\alpha_j^o P^o(w|\theta_j)}{\sum_{i \in \{C,I\}} \alpha_i^o P^o(w|\theta_i)} \tag{16}$$

The key idea of our regularized EM algorithm is as follows: the model selection probability and the refined model parameters estimations should maximize the likelihood that the document is generated from the mixture model as much as possible, with regularization that the refined estimations should be similar with the original estimations (computed as in subsection 2.1 and 2.2) as much as possible. In this way, we expect to integrate the local information and global information better.

$$\begin{aligned} Q^R(\psi; \psi^o) =& \lambda Q(\psi; \psi^o) + (1 - \lambda) \sum_j D_{KL}(\theta_j^g \parallel \theta_j) \\ \simeq& \lambda \sum_w tf(w, D) \sum_j z(w, j) \log \alpha_j P(w|\theta_j) \\ &- (1 - \lambda) \sum_j \sum_w P(w|\theta_j^g) \log P(w|\theta_j) \\ &+ (1 - \lambda) \sum_j \sum_w P(w|\theta_j^g) \log P(w|\theta_j^g) \\ &+ \mu(\sum_j \alpha_j - 1) \\ &+ \nu \sum_j (\sum_w P(w|\theta_j) - 1) \end{aligned} \tag{17}$$

As illustrated in formula (17), in the regularized EM algorithm, we denote the original model parameters estimations got from subsections 2.1 and 2.2 as θ_C^g and θ_I^g respectively, and denote the refined parameters to be computed

by regularized EM algorithm as θ_C and θ_I correspondingly. We then define the regularized expectation of complete likelihood $Q^R(\psi;\psi^o)$ for EM algorithm as a combination of two portions: 1) the expectation of complete data likelihood $Q(\psi;\psi^o)$ as in classical EM algorithm, and 2) the KL-Divergence of original estimations towards new estimations $\sum_j D_{KL}(\theta_j^g \parallel \theta_j)$. In formula (17), $\mu(\sum_j \alpha_j - 1)$ and $\nu\sum_j(\sum_w P(w|\theta_j)-1)$ are Lagrange Multipliers with respect to limitations of $\sum_j \alpha_j = 1$ and $\sum_w P(w|\theta_j) = 1$.

There are closed form solutions for the regularized EM problem as follows. The estimation of α_j remains the same as in original EM, while $P(w|\theta_j)$ reflects the influence from both the document and the original estimations. $z(w,j)$ is same as in formula (16).

$$\alpha_j = \frac{\sum_w tf(w,D)z(w,j)}{\sum_j\sum_w tf(w)z(w,j)} \tag{18}$$

$$P(w|\theta_j) = \frac{(1-\lambda)P(w|\theta_j^g) - \lambda tf(w,D)z(w,j)}{(1-\lambda) - \lambda\sum_w tf(w,D)z(w,j)} \tag{19}$$

3. COMMERCIAL QUERY BIASED SCORING FUNCTION

Generally in classical language model we utilize the query likelihood function $P(Q|\theta_D)$ to score the document, which denotes the likelihood that the query Q is generated from the language model θ_D estimated by document D.

3.1 Query Commercial Intent Integration

In sponsored search we have a special and critically important demand. We want to serve more advertisements to queries with high commercial intent than queries with low commercial intent. The benefits are double folded: 1) it improves user experience by satisfying users who need to reach advertisements to facilitate their commercial activities without bothering users who do not want to see advertisements at present, and 2) it increases the number of advertisements candidates going to the second (ranking) stage for valuable queries, which helps to increase the probability to draw users' clicks. In this sense, we integrate query commercial intent into query likelihood function.

$$P(Q,C|\theta_D) = P(Q|\theta_D)P(C|Q,\theta_D) = P(Q|\theta_D)P(C|Q) \tag{20}$$

In formula (20), C denotes '*user is under commercial intent*', and $P(Q,C|\theta_D)$ denotes '*the likelihood that a user raises a query Q from the language model estimated from D and the user is to express his commercial intent*'. Because the commercial intent is only expressed by Q, it is independent with θ_D. The method to compute $P(C|Q)$ is same with Sandeep's work of query's advertisability [11] as follows, in which $S \subseteq Q$ and $|S| \leq k$ and k is a truncating parameter to avoid biasing long queries.

$$P(C|Q) = \max_S\{1 - \Pi_{w\in S}(1 - P(C|w))\} \tag{21}$$

3.2 The Scoring Function in Practice

In practice, we use the Bayesian transformed function to score advertisements as formula (22).

$$P(D,C|Q) = \frac{P(Q,C|D)P(D)}{P(Q)} = \frac{P(Q|D)}{P(Q)}P(C|Q)P(D) \tag{22}$$

Here we interpret $P(Q|D)$ as $P(Q|\theta_D)$, which further equals to $\Pi_{w\in Q}P(w|\theta_D)^{tf(w,Q)}$ with the query words independency assumption. For words which exist in query Q without existing in D we assign to them a non-zero probability in the model θ_D as $P(w|\theta_D) = \rho P(w)$, in which $P(w)$ is the global prior probability, and ρ is a document specific parameter which equals to $\frac{\eta}{|D|+\eta}$ [14] (η is the parameter of Dirichlet smoothing that is widely used in traditional language models). With further assuming $P(Q) = \prod_{w\in Q}P(w)^{tf(w,Q)}$, we can compute $\frac{P(Q|D)}{P(Q)}$ as formula (23).

$$\begin{aligned}\frac{P(Q|D)}{P(Q)} &= \prod_{w\in Q}\left(\frac{P(w|\theta_D)}{P(w)}\right)^{tf(w,Q)} \\ &= \prod_{w\in Q\bigcap w\in D}\left(\frac{P(w|\theta_D)}{P(w)}\right)^{tf(w,Q)} \\ &\times \prod_{w\in Q\bigcap w\notin D}\left(\frac{\rho P(w)}{P(w)}\right)^{tf(w,Q)}\end{aligned} \tag{23}$$

And thus we get the final scoring function as formula (24).

$$\begin{aligned}score(D,Q) &= \log P(D,C|Q) \\ &= \sum_{w\in Q\bigcap w\in D} tf(w,Q)\log\frac{P(w|D)}{P(w)} \\ &+ \sum_{w\in Q\bigcap w\notin D} tf(w,Q)\log\rho \\ &+ \log P(C|Q) + \log P(D)\end{aligned} \tag{24}$$

4. EVALUATION

In this section we first evaluate the effectiveness of our model on editorial data set by comparing with a classical baseline language model, the TFIDF and the TFIDF-pLSI vector space model, as well as the deliberated KLD language model, and then show live traffic performance. Our evaluation dataset consists of 87690 query-ad pairs (14000 queries) manually labeled by professional editors.

4.1 Compare with Baseline Language Model

In this section we compare the performance of mixture language model with that of baseline language model. We evaluate the effectiveness of biasing query commercial intent in scoring function as well. The details are as follows.

Baseline LM: we use Hema Raghavan's refinement [14] of the classical Multinomial language model for sponsored search as baseline. In Hema's evaluation, this baseline performs considerately better than the query-rewriting method and TFIDF. In implementation we use $P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)}$ as scoring function and use MLE and Dirichlet smoothing to estimate $P(w|D)$ (both unigrams and phrases from a large phrases dictionary are used). We set the smoothing factor η to be 0.5 which achieves the best performance in our dataset. $P(D)$ derives from the host trust score corresponding to the URL of the advertisements.

Mixture LM: the methods and dataset used to implement our mixture language model are the same as described in section 2 and section 3. The combination coefficient of the $P_g(w,C|D)$ in computing θ_C is 0.6. In the regularized EM algorithm, we re-scaled the $Q^R(\psi;\psi^o)$ to comparable granularity with $\sum_j D_{KL}(\theta_j^g \parallel \theta_j)$ and set λ to be 0.3. These two parameters are chosen empirically by evaluations.

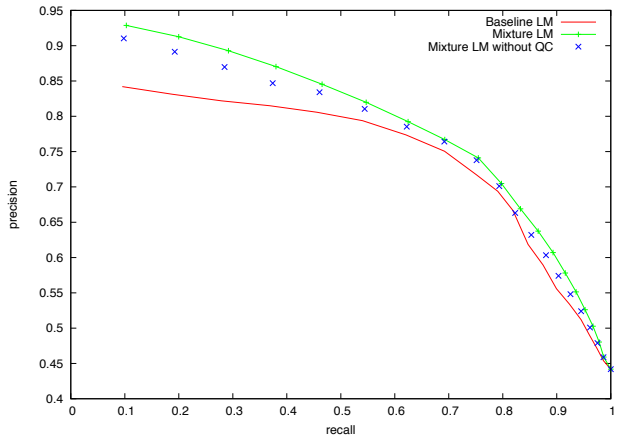

Figure 1: comparison with Baseline LM in PR.

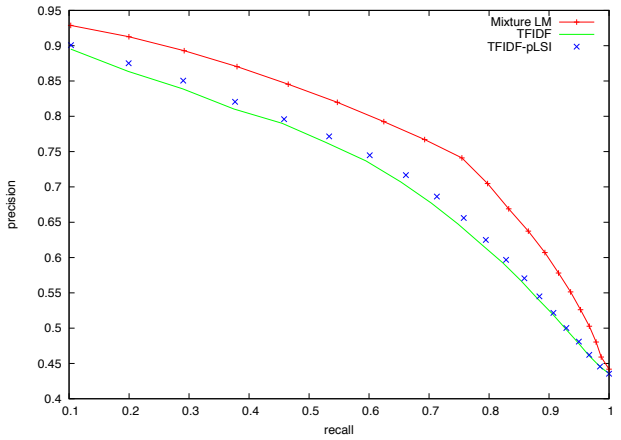

Figure 2: comparison with TFIDF and pLSI in PR.

Mixture LM without QC: the only difference with 'Mixture LM' is that we do not use the commercial queries biased scoring function but use the one same as Baseline LM.

Our Mixture LM performs better than Baseline LM. As in Figure 1, the precision improvement when recall is less than 0.4 is in range 5%-9%. At our operating point in product environment (of 0.3 recall) the precision improvement is 6%. On tail portion when recall is above 0.8 there is about 1% precision lift as well. In terms of ROC, from Table 1 we can see that the improvement in AUC is 5.3%.

Both the mixture model estimation methods and commercial query biased scoring function (QC for abbreviation) contribute to the improvement. However, they contribute in different ways. Mixture model estimation focuses on advertisements side and contributes through differentiating terms' roles in expressing people's intents, while QC focuses on query side and contributes through relatively boosting scores for high commercial intent queries which deserve more advertisements.

4.2 Compare with TF-IDF and pLSI

In this section we compare the performance of mixture language model with that of TFIDF vector space model and TFIDF-pLSI combined vector space model. Before implementing language model, we have deployed well tuned TFIDF and TFIDF-pLSI combined model in production whose performance are very promising. The details of candidates are as follows.

TFIDF: we elaborately refined the classical TFIDF model [16] for evaluation. We introduced term importance t leveraging on click feedback and tuned different weights z for different zones (title, description and bid phrases) by simulated annealing. The advertisements side feature weight for term w is then $t(1+\log(\sum_i tf(w, zone_i)z_i))\log idf$, and that of query side is similar. The matching score is cosine similarity. Both unigrams and phrases are used.

TFIDF-pLSI: in order to integrate semantic benefits into matching, we combined TFIDF with pLSI topic model linearly. We trained 1000 topics and utilize fold-in algorithm [8][3] to compute $P(t_i|Q)$ and $P(t_i|D)$, and then use cosine similarity for scoring. We elaborately adjusted the combination coefficient to reach the optimal performance.

Mixture LM performs superior than TFIDF and TFIDF-pLSI combination. As illustrated in Figure 2, for recall in range of 0.1 and 0.8 the precision lift is between 4% and 12%. Particularly for recall in the area we are the most interested in (near 0.3) for production deployment, the precision improvement is about 5%. In terms of ROC, from Table 1 we can see that the improvements in AUC are 6.1% and 5.0%.

The improvement on tail portion is some more significant than that on head portion. The reasons are two-folded. First, TFIDF does have good discrimination ability on head portion. Our investigation has confirmed that it performs better than Baseline LM when recall is low. The integration of term importance derived from click feedback further sharpens it's performance in this area. Second, language model is more robust. Its scores distribution is more concentrated than TFIDF. Even Baseline LM performs better TFIDF when recall is in middle and tail area. Mixture LM strengthens this advantage further.

Besides, we find that semantic match does help to improve syntactic match. Combined with pLSI, TFIDF gains about 1% precision lift in our interested recall range.

Table 2: comparison with KLD LM on nDCG

Methods	nDCG@1	nDCG@3	nDCG@5
Mixture LM	0.732	0.736	0.753
KLD LM	0.723	0.731	0.747

4.3 Compare with KLD Language Model

In evaluation, we differentiated hot and tail queries according to whether they have at least three 'relevant advertisements' in the logs of an industrial search engine in one month. An advertisement is regarded as a 'relevant advertisement' for a query if the absolute click number of the query-advertisement pair exceeds a threshold l_1 and the expected click number predicted by click modeling exceeds a threshold l_2. With these limitations, there are about 9000 queries regarded as 'hot queries' and the left 5000 queries are 'tail queries'. We use $\sum_w P(w|\theta_Q)\log P(w|\theta_D)$ for scoring. For hot queries, we integrated 'relevant advertisements' into query models estimation. The combination coefficient λ of background prior $P(w|C)$ is 0.05 in EM estimation.

We first retrieve the top n advertisements for each query from the two models and then evaluate precision and recall

Table 1: AUC values of different models

Methods	Mixture LM	Baseline LM	TFIDF	TFIDF-pLSI
AUC	0.837	0.784	0.776	0.787

on different n, which comes out that the overall precision of KLD LM is slightly better than Mixture LM, about 0.5% overall. However, as illustrated in Table 2, the performance on nDCG is opposite, in which Mixture LM is about 0.5% better than KLD LM. These results show that these two models are comparable for hot queries generally. KLD LM is somewhat better at telling the good from the bad, while Mixture LM is somewhat better at distinguishing the best from the fair. For tail queries, Mixture LM is 2%-4% better than KLD LM. Considering the overlap of advertisements in feedback collection and editorial dataset which may have boosted the performance of KLD LM and the volume of tail queries, we think Mixture LM is superior for our advertisements retrieval tasks.

4.4 Live Traffic Performance

Besides the offline evaluations, in order to compare the online performance of different models, we split the live search traffic into different 'buckets', a Baseline LM bucket, a TFIDF-pLSI bucket and a Mixture LM bucket. Users in each bucket are served by advertisements retrieved by the corresponding model in addition to other matching approaches already existing in the system such as query rewriting, advertisements expansion and so forth. As such, the CTR (click through rate) and RPS (revenue per search) of the Mixture LM bucket is 3.0% and 3.6% higher than that of TFIDF-pLSI bucket; and the improvements are 2.0% and 2.5% respectively comparing with the Baseline LM.

5. RELATED WORK

The fundamental work on advertisments selection of sponsored search mainly focus on query expansion [13] or bid term generation [15]. Besides IR technologies, some machine learning approaches are exploited as well, such as Dustin's relevance model [6] and Broder's 'learn when to advertise' model [4]. In recent years, some work pay attention to the commercial nature of online advertising, such as Pandy's work in estimating advertisability of tail queries [11] and Ashkan's work on queries' commercial intent [1].

Early work of language model mainly remain in the family of query-likelihood scoring [12]. As a generalization, the KL-divergence language model [18] which supports feedback integration through estimating a query language model is regarded as a 'state-of-the art' approach [17]. Flourishing works in query language model estimation include Zhai's mixture model feedback method [18] and Croft's relevance model [9] etc. Very recently, Raghavan [14] introduced the classical language model into searching advertising.

6. CONCLUSIONS

In this paper we focus on characterizing and capturing the relevance and commercial intent in sponsored search. We develop a mixture language model of a commercial model and an informational model for this purpose. The evaluation on both editorial data and live traffic confirm the effectiveness of our model. For future work, we are still investigating the generalization of our work to content match advertising as well as some personalization areas such as sentimental analysis and news recommendation.

7. ACKNOWLEDGMENTS

This work is sponsored by Yahoo! Global R&D Center, Beijing.

8. REFERENCES

[1] A. Ashkan and C. L. Clarke. Term-based commercial intent analysis. In *SIGIR*, pages 800–801, 2009.

[2] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res*, 2:993–1022, 2003.

[3] T. Brants and F. Chen. Topic-based document segmentation with probabilistic latent semantic analysis. In *CIKM*, 2002.

[4] A. Broder, M. Ciaramita, and M. Fontoura. To swing or not to swing: learning when (not) to advertise. In *CIKM*, pages 1003–1012, 2008.

[5] A. P. Dempster and N. M. Laird. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statist. Soc. B*, pages 1–38, 1977.

[6] D. Hillard and S. Schroedl. Improving ad relevance in sponsored search. In *WSDM*, pages 361–369, 2010.

[7] T. Hofmann. Probabilistic latent semantic analysis. In *UAT*, 1999.

[8] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[9] V. Lavrenko and W. B. Crof. Relevance-based language models. In *SIGIR*, pages 120–127, 2001.

[10] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *WWW*, 2008.

[11] S. Pandey and K. Punera. Estimating advertisability of tail queries for sponsored search. In *SIGIR*, 2010.

[12] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR*, 1998.

[13] F. Radlinski, A. Broder, and P. C. etc. Optimizing relevance and revenue in ad search: A query substitution approache. In *SIGIR*, pages 403–410, 2008.

[14] H. Raghavan and R. Iyer. Probabilistic first pass retrieval for search advertising: from theory to practice. In *CIKM*, pages 1019–1028, 2010.

[15] S. Ravi, A. Broder, and E. Gabrilovich. Automatic generation of did phrases for online advertising. In *WSDM*, pages 341–350, 2010.

[16] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 23(5):513–523, 1998.

[17] C. Zhai. Statistical language models for information retrieval a critical review. *Foundations and Trends in Information Retrieval*, 2(3):137–213, 2008.

[18] J. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM*, 2001.