Query Expansion and Query Fuzzy with Large-Scale Click-through Data for Microblog Retrieval

Wei Pang and Junping Du

***Abstract*—Searchers using microblog search often have difficulty in expressing their needs, they often struggle to modify queries by adding or removing terms, and this lead to unsatisfied experience. On the other hand, Microblogs content is limited in size, contain a few words and these will cause a real problem of term mismatch in microblog search.**

We propose a probabilistic model for query reformulation on click-through data, which can help user auto-complete their search need. We assume that the knowledge of good queries of former searchers can then be used for improving a poor query of following users, by expanding extra meaningful terms or fuzzing trivial query terms. The key idea is to discover terms that can grasp users' click need or carry less meaning from large-scale click-through data. Our results indicate the reformulated query can better describe users's real functional need. In the experiment, we try to reduce query drift and noise in click-through data by using crowd-sourced relevance feedback and smoothing methods. The experimental result is much better than that of searching only the original query, especial on gains of low-frequency and long-tailed query.

***Index Terms*—Query reformulation, query intent, query understanding, microblog retrieval.**

I. Introduction

Query document mismatch [1] is a severe problem in current search based on the bag-of-words method. Users and document often use different words to convey the same meaning, the relevant result is returned rely on matching between query terms and document terms [1]. Especially in microblog search, the tweet content often contains a few words, the mismatch problem is serious. There exist semantic gaps between query and document representation, such as problems like synonyms, polysemous, and paraphrase [1], [2]. To address the mismatch problem, an important direction is to per- form semantic matching [1], which focus on query and document understanding. Query reformulation plays a key role in query under- standing task, is an effective way to handle mismatch problem [1]. In practice during the search, many people often try to modify queries in order to make their actual need more concrete, and get a better relevant result, especially when seeking an unclear needs. However, with the easy availability of big data, e.g., click-through data, we are able to help users get a better search experience actively. In this paper, we target at the problem of helping user rewrite their query. Under the assumption that two queries have the similar meaning if they share more frequently co-click the same documents, we can discover the historical knowledge from users' relevance judgments between queries and documents. These bits of knowledge produced by former searchers can then be used for Microblog Retrieval to make suggestions to following users.

Query reformulation, also called query rewrite, consists of six major types [1], [3], [4], but here we only focus on two types, query expansion and query deduction. Query expansion is an efficient technique to enrich query by adding new terms into the original query [1]. Also, query expansion is demonstrated as an effective method to improve search relevance [1], [5]. Prior works have mainly based on thesaurus [2] and external resources [6], [7], such as WordNet [2], [6], [8], [9] and Freebase [10].

Indeed, based on pseudo- relevance feedback [1], [11], [12], query expansion obtain state-of- the-art result [1], by selecting expanded words from the initial returned documents, and then conduct search again. However, this method is highly relied on the initial search result. Recently, word embedding [7], [13]-[15] is also used to query expansion, selecting similar words of query terms as expansion keywords. Although these types of query expansion can augment some synonyms of the original query term, they can't help understand users' latent needs and functional needs, because of additional term lacks the knowledge of intention. Yiqun Liu *et al.* [16] propose a snippet click model, which mine keywords from snippets of documents clicked by users. Their basic idea is similar to ours, we all use the user's click behavior in common, but the implementation is very different in terms of the source of keywords. We select terms for expanding from historical queries, while theirs comes from the clicked document.

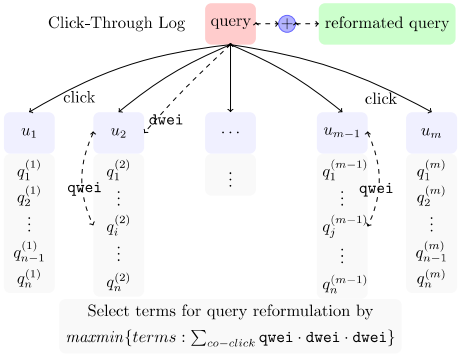

Fig. 1. An illustration of probabilistic model for query reformulation.

Query deduction is a technique to drop query terms with little informative from long query [17]. It enables improving long queries up to 25% in NDCG@5 [17]. However, query deduction has lower occurrence than query expansion [4], [18] among all types of query reformulation. That means people often tend to add terms when they are dissatisfied. However, removing some words [1] from user's raw query maybe not a good idea, in the worst case, the deduction could conceal user's practical search intention. Therefore, in this

Manuscript received June 5, 2018; revised April 22, 2019.

Wei Pang and Junping Du are with Beijing University of Posts and Telecommunications, Chia (e-mail: pangweitf@163.com, junpingdu@126.com).

case, we adopt an alternative strategy, by fuzzing trivial terms rather than removing directly. The key idea is to weakening their role in the retrieval process, we call this strategy as fuzzy technique. Moreover, we consider that query fuzzy technique can preserve as much initial search intention as possible, and can't reduce retrieval experience. However, the reformulated query might have a very different meaning [1], due to the spareness of social media content, query reformulation in microblog retrieval easily tend to topic drift [1].

To address the above issues, we directly model click-through data for query reformulation. (1) We propose a probabilistic model to discover useful or trivial terms from click-through data for expand- ing or fuzzing a query in microblog retrieval. The main assumption is that the added term comes from click-through log can better represent users' actual thinking process than from a third-party source, e.g., Probase or WordNet. Hence, we mine the expanded terms or fuzzy terms from historical queries in click-through log. Another reason is that few click behavior may be not reliable[16] due to noise or biases[1], [16], we focus on crowd-sourced relevance judgment since people's eyes are always discerning. (2) Leveraging historical pieces of knowledge, we present a query expansion method to make the original query more descriptive and accurate. (3) We adopt fuzzy strategy for query deduction, instead of removing strategy. (4) We integrate the proposed model with our's search engine and work well together. In the experiment, one interesting finding is that the proposed model can provide special terms that can express user's functional need. For example, a raw query is 'Huaian gym annual card', expansion term 'group-buying', which would convey actual useful need, then yield a good query.

As show in Fig. 1, it visualization the architecture of query reformulation framework, where u denote a document, q note a query, we select most important terms that express actual search need and then expand to the original query. Under the assumption above, we extract the knowledge of historical click information that is derived from click-through log data.

To experiment, we prepare two datasets, WeiBo Tweets data, and SougouQ click-through data [19]. Compare with three state- of-art methods, our model achieve better results than the baseline.

To summarize, we detail contributions as follows.

(1)We propose a probabilistic model on click-through data for query reformulation task, integrating query expansion and query reduction into a single framework.

(2)The knowledge derived from historical crowd-sourced relevance feedback of users give insights into understand query intention. It is helpful to reduce topic drift in query expansion task.

(3)We adopt fuzzy strategy for query reduction, instead of the original dropping strategy.

(4)Our approach explicitly improve the quality of search service, by transforming a tail query into a head query. As visualized in Fig. 2, on the x-axis there exist three queries, the queries that located both sides are low-frequency, the middle one is a hot query. The goal of this work is to definitely convert a low-frequency query into high frequency using query fuzzy or expansion technique.

In rest of this paper, Section II describes related works, especially about query expansion. Section III gives a formulation of the proposed model in detail. We integrate the proposed model with microblog retrieval engine in Section IV. In Section V, we perform experiments and show evaluation results. In Section VI, we summarize the conclude and findings.

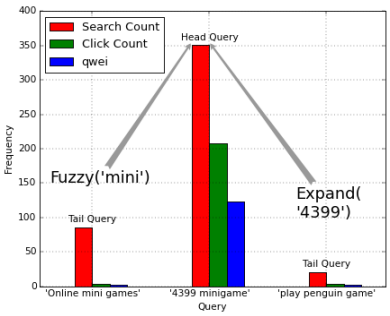

Fig. 2. An example of rewriting a tail query into a head query that they co-click the same 'http://www.4399.com/ ', where fuzzy('mini') means by query fuzzy method, expand('4399') via query expansion technique.

II. Related Work

Query reformulation plays a significant role in query understanding for semantic matching in search [1], [20], [21], and it helps to solve the mismatch problem [1] between query and document. Query reformulation is a conventional method to translate user query to another to make their meaning more effective. Generally, Query reformulation is consist of spelling error correction, stemming, query segmentation, query expansion, and query deduction (Hang Li *et al*., 2014). In this paper, we only focus on two basic types, query expansion and query deduction technique here. There are major three types of query reformulation, including adding terms, removal of words and substitution [1], [18], [22]-[24].

Query expansion is a well-known method to deal with term mis- match problem [1], [6], [9], [13], [25]. One typical way is to use external resources [1], [2] to select additional terms. Standard source of such additional terms are derived from knowledge bases, such as Free- base (Xiong *et al*., 2015), Probase (Wang *et al*., 2017), WordNet (Lu *et al*., 2015), ConceptNet and Wikipedia (Arguello *et al*., 2008; Maaike *et al*., 2016; Kotov et al., 2012). Search log [26] and Anor text [5]. An- other representative method is pseudo-relevance feedback [1], [11]. Under a strong assumption of the top-retrieved document are relevant, the expansion terms are selected from the initially returned documents, and then prompt a second search with the original query plus expansion terms. Recently, query expansion based on word embedding (Kuzi *et al*., 2016; Diaz *et al*., 2016; Roy *et al*., 2016; Liu *et al*., 2017) is widely applied. They train Word2Vec model over the entire corpus and select terms that are semantically related to the raw query in the word2vec space [7], [12]-[15]. These expansion methods tend to find synonyms or related words of a query word [6] and can solve a part of mismatch problems. Reinforcement Learning is also studied in query reformulation task [27], [28], authors(Nogueira *et al*., 2017) propose a neural network with reinforcement learning to model relationships of expansion terms and document recall, selecting some terms to maximize recall rate of relevant documents returned. In [7], authors(Qian Liu *et a*l., 2017) also use word embedding to

select top similar words with the initial query via cosine similarity, the authors propose four fuzzy rules to reweigh t the expansion words [7], these differ from our fuzzy technique, we focus on fuzzing uninformative terms in the raw query. In question answering system, authors (Buck *et al*., 2018) consider question rewrite as a query reformulation task that tries to mimic the process of question reformulation by users.

In contrast to query expansion, query deduction is another technique of query reformulation via removal of trivial terms from the query [1], [18]. But in our task, instead of removing a word, we focus on fuzzing superficial terms in the original query, we name this type of reformulation as query fuzzy. Compared to deduction, one advantage of fuzzy reformulation is that we can preserve as much search need of users' original intention as possible, meanwhiles increases in document recall rate.

III. Probabilistic Query Reformulation Model

A. Problem Definition

Given a query q, we use p(t|q) to represent the conditional prob- ability of a term t relate to the given query, our goal is to select the essential or trivial terms according to conditional probability. Click-through log data contain user queries and record clicked URLs of documents, which is the most valuable resource for query reformulation task. Typically, a user click a document after submitting a query means that the document is relevant about the query, although there exists an uncertain situation, such as random noise in click behavior [1], [16]. Suppose that queries may convey the same meanings when they share the common clicked documents [1]. Good queries can match well and get many relevant documents ranked at the top position, e.g., at the top three positions in the head page. But, a poor query might get bad search result, because user's actual need for the poor query is not enough to understand, or the satisfying results might rank later since the score is lower due to term mismatch problem. However, during the searching, although the search result is bad and irrelevant, there often exist some users to browse the later result until to find a relevant result, click the URL and complete this search action. Some users may struggle to rewrite a series of queries until to success, especially when a user seeking information is particular and particular [22]. Motivated by this, we design an active query reformulation approach to address the issues. In this paper, we aim at improving the users' search experience, by extracting the knowledge from the successful search experience.

With the availability of large-scale click-through log data, the weight of terms within query can be measured under clicked documents, and then transform the crowd-sourced feedback information into knowledge to conduct query reformulation. If a term plays the critical role in representing user's real search need, the weight of the term should be more significant than others, and then the term tend to be selected as an expanded term. In this way, the knowledge representations is learned by the term weight using maximum likelihood method. Let's explain in detail with an example, if many users issue query 'Alipay fast payment', then click an official website at the top position. While a few people submit a query 'Alipay' and click the same official website at the fourth page, then it is very likely that we can combine the two queries together by the same click behavior, they possibly share the same search need. Suppose that if search engine could find these valuable information, the engine can aid users to understand intention actively. We extract knowledge from good queries and obtain some meaningful terms, e.g., 'fast payment' reflect users' functional need. Moreover, if a user searches for 'Alipay', search engine automatically expand it, perform search with the new query 'Alipay fast payment', achieving more fruitful search results. And more importantly, many relevant documents are boosted into the top position in terms of the expanded terms. On the contrary, a query 'natural logarithm transformation' [1], 'natural' is invaluable in understanding user need, so we fuzz the trivial 'natural' in order to obtain more relevant results.

As listed in Table I, we define some symbols used in this paper. Suppose we group all the queries by co-click, which are likely to express the same need. $Q = \{q^{(i)}\}^N$ denotes a set of queries that i=1 share the same clicked documents, $U = \{u^{(i)}\}^M$ notes as the i=1 co-clicked documents given that Q, N is the number of queries, u indicates a document and M is the total number of documents. Let t indicates a query term, q indicates a query.

TABLE I: Notations Used in Our Model

Symbol	Description	Symbol	Description
$q, q^{(i)}$	a query, the i-th query	Q	a set of queries
u	a document	$\mathcal{U}$	a set of documents
t	a query term	$u^{(j)}$	the j-th document
$p(t\|q)$	conditional probability of term t given a query q		
$qwei$	the click weight of a particular query under given document, noted as $p(q\|u)$		
$dwei$	the click weight of a particular document under given query, noted as $p(u\|q)$		

B. Formalize Probabilistic Model

To extract knowledge from click-through data, we consider a probabilistic model for click-through data. To simplify the discussion somewhat, we initially consider two co-clicked queries $q^{(i)} = \{t^{(i)}\}^n$ and $q^{(j)} = \{t^{(i)}\}^m$, each query is decomposed as a bag n n=1 m m=1 of words, consist of n and m query terms respectively. If we want to choose some terms from $q^{(j)}$ for reformulating $q^{(i)}$, how to measure the weight of each query term of $q^{(j)}$. In this work, our main objective is to calculate conditional probability, $p(t \in q^{(j)}|q^{(i)})$.

From a complex network perspective, it is also formalized as a 2nd order random walk model. As shown in Fig. 3, we construct a click network using users' click behavior.

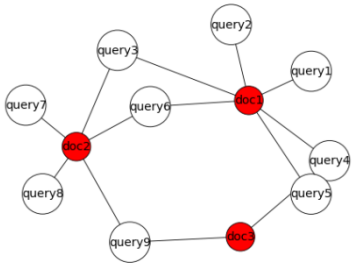

Fig. 3. Visualization of network graph, constructed with query and it's clicked document, where the red node is document, the white one means a query, the edge represent a click occurs.

Suppose we starting at a query $q^{(i)}$, and simulate 2 random

walk pass a document, then lead to $q^{(j)}$. Actually, there exist multiple paths between the two queries. Along each path, we define the conditional probability $p(t \in q^{(j)}|q^{(i)})$ using the multiplication rule. The term weight can be expressed as the sum conditional probabilities of multiple paths.

$$p\left(t \in q^{(j)} \mid q^{(i)}\right) = \int_{u \in U} p(t|u)p(u)du \tag{1}$$

where $p(u)$ denotes the marginal distribution of clicked document u, $p(t|u)$ is term weight of t under the clicked document u. Notes that in this work $p(u)$ can be extended to include $q^{(i)}$ and $q^{(j)}$, then it is dened as below.

$$p(u) = \prod_{q \in \left\{q^{(i)}, q^{(j)}\right\}} p(u|q)p(q) \tag{2}$$

Based on the analysis of the relationship between $q^{(i)}$ and $q^{(j)}$, we establish a probabilistic model from the viewpoint of random walk. The more important the walk path is, the larger conditional probability will get. That is, the term might play a pivotal role in representing user's needs within $q^{(i)}$. As an approximation, we calculate $p(t|u)$ by using its source query $q^{(j)}$, and $p(t|u)$ is given by $p(t|u) \approx p(q^{(j)}|u)$. The new model is then described as

$$p\left(t \mid q^{(i)}\right) = \sum_{k=1}^{m} \frac{p\left(u^{(k)}|q^{(j)}\right)}{\{dwei\ part\}} \frac{p\left(q^{(j)}|u^{(k)}\right)}{\{qwei\ part\}} \frac{p\left(u^{(k)}|q^{(i)}\right)}{\{dwei\ part\}} \tag{3}$$

Let $p(q^{(j)}|u^{(k)})$ denotes the click weight of a special query $q^{(j)}$ underlying the given clicked document. Similarly, $p(u^{(k)}|q^{(i)})$can be represented as the clicked weight of a special document $u^{(k)}$ when given a query $q^{(i)}$. Formally, we give a detailed calculation as following subsections.

1) Query weight score

Definition III. A. Query weight, abbreviated as qwei, which is a weight distribution of co-clicked queries underlying the same document. The empiric formula is de ned as:

$$p(q|u) = \frac{\frac{click\text{-}count^2(<q,u>)}{search\text{-}count(q)}}{\alpha \sum_{x \in \{query\ coclick\ u\}} \frac{click\text{-}count^2(<x,u>)}{search\text{-}count(x)}} \tag{4}$$

where search-count(q) denotes the total search count of query q, click-count(< q, u >) is the total clicked times between the pair of < q, u >. The symbol clicked_count squared we used in the formula because users' clicking is an important evidence in favor of relevance.

2) Document weight score

Definition Document weight, short for dwei, means a weight distribution of documents clicked by the same query and also has an empiric formula:

$$p(u|q) = \frac{\frac{click\text{-}count^2(<u,q>)}{search\text{-}count(q)}}{\beta \sum_{x \in \{url\ clicked\ by\ q\}} \frac{click\text{-}count^2(<u,y>)}{search\text{-}count(q)}} \tag{5}$$

where α and β are factors of round numbers, in our experiment, we set to 0.01. In practice, we can find many pairs of $\{q^{(i)}, q^{(j)}\} \cdots j\ i$ that co-click the same documents. Such crowd-sourced judgment derived from users are valuable in inferencing what a user wants to know, and yield insights that how people who generate a query using keywords step by step. More generally, integrating all co-clicked pairs < q, u > that are associated with $q^{(i)}$, we sum total probabilities of the term as follows:

$$p\left(t|q^{(i)}\right) = \sum_{j \neq i} p\left(t \in q^{(j)} \mid q^{(i)}\right) \tag{6}$$

Terms (words or phrases)[1] are ranked in descending order according to Equ. 6. In the experiment, we choose expanded terms with the top k for query expansion, fuzzy terms with tail l for query fuzzy. In this way, we observe that the reformulated query enable to grasp the search need. As an illustration detail, Table II visualize the procedure for expanding or fuzzing, Table IV show some examples for query reformulation.

TABLE II: AN ILLUSTRATION OF THE PROCESS OF EXPANDING OR FUZZING FOR A NEW QUERY

New query	Candiant List		Remark
	Term	Prob.†	
query	$term_1$	$prob_1$	**Top k terms for expand-ing.**(Grasp user click need)
		∨	
	$term_2$	$prob_2$	
		∨	
	⋮	⋮	
		∨	
	$term_k$	$prob_k$	
	⋮	⋮	
	$term_{n-l}$	$prob_{n-l}$	**Tail l terms for fuzzing.** (Carry less click information in the original query)
		∨	
	⋮	⋮	
		∨	
	$term_{n-1}$	$prob_{n-1}$	
		∨	
	$term_n$	$prob_n$	

† ranking in descending order by probability.

C. Merger Query Expansion and Fuzzy

As visualize in Table II, we merger query expansion and query fuzzy strategy together into a single framework. Moreover, there are some benefits of crowd-sourced relevance judgment from many web searchers. It is helpful to reduce topic drift and mismatch problem, since the historical knowledge from a huge amount of click-through data give insights into understand query intention. Another advantage of the model is its explanatory, we can easily track why the term is adding or fuzzing. However, query reformulation easily bring up a series of problems, including ambiguous terms, irrelevant statistically correlated terms[2], and the more severe problem is topic drift [1], [18] which indicating a change in query intent. To avoid these issues, we adopt three strategies that seem to be working well together.

1) We reduce the coverage ratio of query expansion and fuzzy algorithm, in order to hold down the number of impacted queries. Furthermore, to deal with noise in the click-through data [1], [16], we use some smoothing methods, details in Section III.D.1;

2) We utilize knowledge extracted by user behavior in click- through log data, which represent historical relevance knowledge of crowd-sourced feedback by many users;

3) Information retrieve mainly include three steps, intersecting on inverted index, matching and ranking [2]. In our work, the expanded term or fuzzy term plays an auxiliary role in retrieving. Firstly, in intersecting phase, the expanded or fuzzy term is transparent to the engine. Even though we introduce irrelevant or ambiguous terms, they do not affect the postings intersection algorithm, so they do not reduce the recall rate. On the other hand, the fuzzy term is not participated in intersecting, naturally, the recall rate will be increased significantly in query fuzzy task. Secondly, in matching and ranking phase based on the set of initial documents are retrieved from the index, we are interested in the documents that contain the expanded terms, and we tune the relevance score via the proximity-weighted scoring function sum over the expanded terms. Similarly, if some of the matched words are the fuzzy terms, the score will be discounted.

D. Parameter Setting

Here we provide two methods to estimate the model parameters, dwei and qwei. Firstly, we only consider a single document, that means, group all the queries that co-click a single specific document, we call this method as local query model. Then query is each paired with everyone in the group, and calculate the conditional probability in each pair according to Equ 1-5. Local query model is described as Algorithm 1.

Secondly, given a query, which might not only click a single document, but also share many documents with other queries. We sum the probability along another click paths according to Eq. (6). We view this as a global query model, which extend the search space of candidate terms. Summarize in Algorithm 2.

Algorithm 1: Local Query Model

Input: Click-through data
Output: probability $p(t|q^{(i)})$ for every click term
1. Extract all click pairs$< q, u >$ in the data;
2. Statistics on *search_count* and *click_count*;
3. $Q = \{q^{(i)}\}_{i=1}^{N}$ a query cluster by co-clicked the u ;
4. **while** *each* $< q^{(i)}, q^{(j)} > \in Pairs(Q)$ **do**
   5. $square \leftarrow clicked\_count^2(< q, u >)$;
   6. $score \leftarrow Wilson(square, search\_count(q))$;
   7. calculate *qwei*, *dwei* by Equ 4-5;
   8. weight each term, compute $p(t|q^{(i)})$ with Equ. 3;

Algorithm 2: Global Query Model

Input: $p(t|q^{(i)})$, Click-through data
1. Extract all pairs$< q, u >$ in the click-through data;
2. **while** *each u* **do**
   3. **while** *each* $< q^{(i)}, q^{(j)} > \in Pairs(Q$ *co-click with u)* **do**
      4. sum $p(t|q^{(i)})$ if $t \in q^{(j)}$;

Output: given $q^{(i)}$, rank terms in reversed order;
6. Top k terms for expanding $q^{(i)}$;
7. Tail l terms for fuzzing $q^{(i)}$;

From Algorithm 2, we obtain the expanded terms and fuzzy terms for each query, and store the model data using a hash lookup table. To avoid repetitive terms and reduce memory space, we decompose model parameters into two tables. As show in Fig. 4, one table is HashLookUp Table in descending order, store hash value for query with their additional information, including hash of terms, the pointer that denotes the o set of terms in the terms table. The other one is terms table, only memory unique terms, consist of expansion terms and fuzzy terms. During looking up, the key is the hash value of the query, and then perform hash search as quickly as possible.

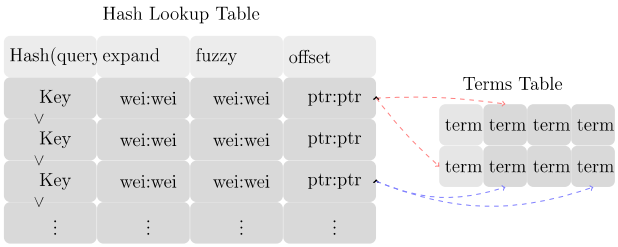

Fig. 4. The storage structure for model data, keys is sorted in ascending order, and this convenient to conduct binary search.

Estimate score with smoothing methods

From a statistical perspective, a severe problem in click-through data is the unbalanced and limited sample size, leading to the score is biased, since the score of dwei or qwei is similar to CTR(Clickthrough rate), are also de ned as a ratio of click numbers to search(or impressions) numbers. Another problem is that click-through data usually have much noise and sparsenesss [1], [16], [26], the ratio affected by noise is also biased. Take an example in Table III, if we have 1 click and 2 impressions, then the CTR would be 50%, while if we have 50 clicks and 100 impressions, the CTR is the same. But, clearly, the two cases have different meanings, the later one is more reliable. However, the Wilson score of 1 click in 2 impressions is significantly lowered to 21.13%. Therefore, we use two tricks as following to alleviate the issues. (1) We square the number of click count for reducing sparseness of click information in the data. (2) We use Wilson score confidence interval [29], [30], abbreviated as Wilson score, to compute Click-Through Rate, avoiding biased ratio caused by sample size determination [29]. Similarly, we also adopt another weighting function Equ. 7 introduced by Pennington *et al*. [31] for discounting the search count and click count.

$$f(x) = \begin{cases} \left(\frac{x}{x_{\max}}\right)^{0.75} & if\ x < x_{\max} \\ 1.0 & otherwise \end{cases} \quad (7)$$

TABLE III: SMOOTHING OF CLICK-THROUGH RATE USING WILSON SCORE COMPARED TO TRADITIONAL CTR (SON SCORE OF CTR METHOD. THE WILSON SCORE OF CTR

search	click	traditional CTR	Wilson CTR†
2	1	0.5	0.2113
20	10	0.5	0.3908
100	50	0.5	0.4502
1000	500	0.5	0.4841

Compare with traditional CTR method as shown in Table III, it is clear that Wilson score method help inhibits noises in click-through data. In search engine task, 50 clicks occurs in 100 impressions might represent stronger and more relevant feedback than one click out of 2 impressions. In SougouQ dataset, the number of impressions is no more than 20 possess about 73.76 percent, the Wilson score of CTR can balance the impact of small sample size caused by random noises [1].

TABLE IV: COMPARISONS ON ORIGINAL QUERY AND NEW QUERY

Original query	Candiant terms	Reformulation	New Query
sina	weibo, video	expansion	sina *Expand*(weibo, video)
sohu	video, news	expansion	sohu *Expand*(video, news)
trade war	ZTE, negotiate	expansion	trade war *Expand*(ZTE, negotiate)
amazing ! this logo	!, this	fuzzy	amazing logo *Fuzzy*(!, this)
The wife wants to go home in half an hour	The, wants, to, in	fuzzy	wife go home half an hour *Fuzzy*(The, wants, to, in)

IV. SEARCH: MATCHING AND RANKING

For practical use, we will focus on applying the proposed model to search engine system, involving in three parts, query understanding phase, matching and ranking phase. In query understanding phase, for a new query, generating expanded terms or fuzzy terms. An important theme to notice, what kind of query need to reformulate. In this work, according to the experimental result shown in Fig. 5, we use the following two thumbs of rules, short query for expanding and long query for fuzzing [17].

A. Expanding Short Query

Definition IV.1. Short query, the size of query terms is less than 5, is in need of expanding for improving the retrieved accuracy.

How many distinct terms k we chose to expand? With more terms for query expansion, it might cause problems in topic drift [1], [32]. While less terms expand to a query would not bene t. To solve this tradeoff issue, our focus is the relationship between the CTR and the number of query terms, as heuristics information to help us select a proper number of expanded terms. Finally, we empirically set up the number of expanded terms k to 2, can be visualized and verified graphically in Fig. 5.

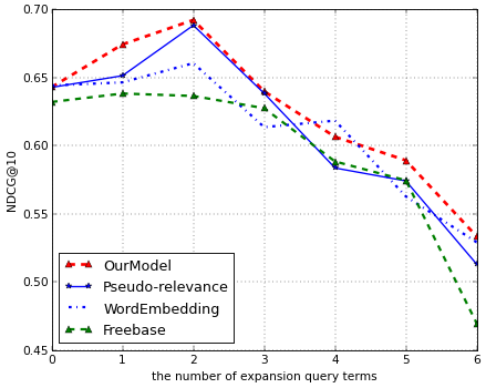

Fig. 5. Relationship between the number of expanded terms and its NDCG@10. The number is very close to 2 we achieve the optimum value of NDCG.

B. Fuzzing Long Query

Definition IV.2. Long query, the size of query terms is more than 7. We restrict query fuzzy strategy to long query, and fuzz trivial query terms for increasing document recall rate.

Query fuzzy strategy, a type of query reformulation, is denied as the reverse formation of query expansion through fuzzing query terms with little meaning. We consider trivial terms as the minimum weight of conditional probability $p(t|q^{(i)})$, and also set l to 2, as shown in Table II.

C. Query Understanding

Query understanding is a key role in search engine [1]. Coming a new query, we first check whether or not the query is a long or short query, if so, then compute the hash value of the query, and finally conduct binary search on the hash lookup table. Then the strategy of query reformulation is triggered. The original query, plus expanded or fuzzy terms, together with their weights, are used to perform retrieve documents.

D. Matching between Query and Document

In matching phase based on the inverted index, we can't take the expansion terms or fuzzy terms into account. The latent relevant documents can be matched by intersecting the posting list associated with the other query terms. Therefore, the document recall rate might be not decreased by expansion or fuzzy operation.

E. Ranking of Relevance Score

By initially matched documents, we boost the documents that contain expansion terms with a boost weight, is de ned as follows.

Definition IV.3. Boost weight of expanded terms(notes as ex), compared to original query terms(notes as ori), the boost weight is estimated as:

$$Boost = \alpha \times \frac{wei(ex)idf(ex)}{\sum_{ori} idf(ori)} Jaccard(ex,ori)$$

Jaccard (ex,ori) is a function that denote Jaccard similarity coefficient between hit sentences of expansion terms and original query terms, is given by:

$$Jaccard = \frac{|HitSents(ex) \cap HitSents(or)|}{|HitSents(ex) \cup HitSents(ori)|}$$

idf (t) = 1.0+log(Df /(df (t)+0.5)) is the inverse document frequency of term t , here $d\,f\,(t)$ is the number of documents in which t occurs, Df is the total number of documents in the corpus, wei() stand for the expansion weight. In ranking phase, we access id f value for every term which would be pre-calculated and stored in indexing. This boost weight is added to relevance scores between query and document, and then the score of most relevant documents may be ahead during ranking. We achieve the aim of query reformulation in helping users to get the better result by one-time search.

In terms of hitting fuzzy terms that contain little click information in the original query, the document may be demoted, in order to avoid matching irrelevant words.

Definition IV.4. Demote weight is used to demote fuzzy terms in matching between query and document. We demote weight as below.

$$Demote = \begin{cases} \alpha \frac{|BM25(fuzzy\ terms)|}{|BM25(all\ terms)|} & if\ \ fuzzy\ term\ match \\ 1.0 & otherwise \end{cases}$$

where φ is a damping coefficient, we set to 0.85 by default. BM25 is a widely used method for computing term score [1], [7]. Both Demote and Boost weight are as weighting factors adjusting final ranked score of document.

V. EXPERIMENTS AND EVALUATION

A. Data Sets

Because we prepare two datasets for the experiment. One is SogouQ dataset consist of search and click-through log

released by the Sogou Labs [19]. We perform our model over the SogouQ data in order to discover terms that can grasp users' click need or carry less meaning, after this, we utilize the good terms to expand, the trivial terms to fuzzy a new query in Microblog retrieval. SogouQ dataset contains the query, and it's clicked document per line, in which include four columns: query, rank position, click order and clicked document, where click order means which ranked position the click behavior occurs. In total, there are 20.43M queries. After data preprocessing, we obtain 3.02M of distinct query collection, 8.2M separate documents. After word segmentation into a bag of words, we get a total number of unique words is 695, 148, the average length of queries is 2.488. The distribution of query length as shown in Fig. 6. The other one is MicroBlog data comes from Sina from 2013 through 2018; it contains 74.6K tweets, we use MicroBlog data to make an index for evaluating our model and baseline methods.

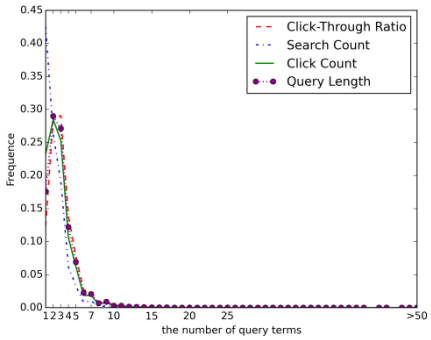

Fig. 6. Statistics of query length distribution, the number of query terms relate to click-through rate, the discounted weight of search count and click count by Equ. 7.

B. Examples of Query Reformulation

Table IV and V details some examples, we find that the expanded terms might grasp user's need in functional-level. Furthermore, the expanded term is primarily showed on two aspects: categorical term, which represent the topic of query, and functional term, which express user's real search need. The experimental results also demonstrate that the terms derived from historical click-through data can better represent user's real thinking logic.

TABLE V: EXAMPLES OF QUERY REFORMULATION IN DETAILS

clicked a given document, such as www.4399.com/						
Query	Search	Click	*Score*†	*qwei*	*dwei*	Reformulation
play penguin game	1	1	0.2113	0.2804	1.0	expanded:4399
4399 online game	86	4	0.1242	0.1648	0.045	fuzzed:online
www + 4399	1	1	0.2113	0.2804	1.0	fuzzed:www,+; expanded:game
office amusement game	1	1	0.2113	0.2804	1.0	expanded:4399
game	7731	60	0.3133	0.4158	0.021	expanded:4399
4399 + !	1	1	0.2113	0.2804	1.0	fuzzed:+,!
return air ticket	1	1	0.2113	76.31	1.0	expanded:commission charges
002008 sina	8	2	0.2147	0.3157	1.0	expanded:stock; fuzzed:002008
hexun	704	249	0.9883	7.228	0.47	expanded:fund,microblog
cigarette	36	5	0.3488	0.2358	0.1763	expanded:wine,specialty store
How to quickly remove melasma	7	2	0.2356	0.5202	0.3053	fuzzed:How, to, quickly

† the score is estimated with Wilson score confidence interval, computing by Wilson(Click², Search).

C. Evaluation Result

We compare with three baseline methods. The first method is pseudo-relevance feedback based query expansion [1], [11], [12], the expanded terms obtained from top-ranked documents initially retrieved, here the title of top 10 documents are used as resources, and we find expanded terms using TF-IDF algorithm. The second one is based on word embedding for query expansion [7], [12], [14], [15], expanded terms are selected by cosine similarity in word vector space. We train word vector on the MicroBlog data using word2vec tool [33] in advance, and use the top 2 most similar words related to query terms for expanding. The third one is selecting terms from Freebase [10], similarly, we also use top 2 words as resources. We compare the performance of the original queries with the reformulated queries separately. We manually label 600 pairs of query and it's most relevant tweets, each query have 3 relevant tweets in average, which are used to quantify the rank quality. Finally, we evaluate relevant results via blind testing between the raw query and the reformulated query, using P@5, NDCG@5 and Mean Average Precision (MAP) respectively. As we can see from Fig. 7, the overall performance of precision, recall and F1-score show that our model gain better result. In addition, we also compare result of top-retrieved documents side by side, one side is the new result and the other is the baseline, measure the search result by a reviewer in blind trial.

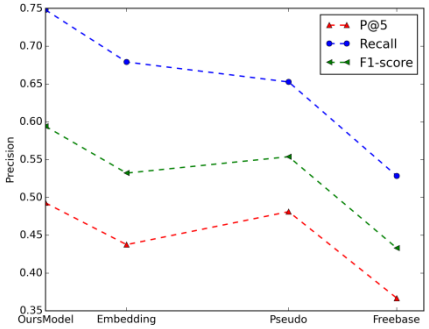

Fig. 7. Query reformulation results compare with four methods on WeiBo tweets, the evaluation queries select from SogouQ randomly.

TABLE VI: THE OVERALL PERFORMANCE COMPARISON OF OUR MODEL AND BASELINES ON WEIBO DATASET

Method	Measures			Query impacted	User review†		
	MAP	*NDCG@5*	*P@5*		Good	Bad	TheSame
OursModel	0.2824	0.5443	0.5056	17.81%	21.5%	6%	72.5%
Embedding based query reformulation	0.2490	0.4785	0.4401	53.60%	11.5%	6.5%	75.50%
Pseudo-relevance based query expansion	0.2406	0.5372	0.4943	48.50%	16.28%	10.23%	73.49%
Freebase based query expansion	0.2166	0.3672	0.3741	34.43%	10.87%	9.39%	79.74%
Avg-improvement	+0.0469	+0.0833	+0.0875	low coverage	+8.61%	-2.70%	-

† We select 200 queries randomly from SinaHotSearch query collection for blind testing by a reviewer.

In experiment, we randomly select 200 queries from the affected queries in SogouQ for testing. The result as shown in Table VI, where 'Good' denotes result of the reformulated query is better than the original query's, 'Bad' is the opposite, 'TheSame' means they are as good as each other. Clearly, our model is better than that of using other methods, that means the terms comes from historical queries usually contain user's real search need. But the number of query impacted is much lower than the baselines, since we try to reduce topic drift [1]. We also note that there is a tradeoff between query drift and query reformulation. Given a query, we add more terms usually lead to diversify the search results, while, it is easy to fall into the problem of topic drift. Therefore, we restrict the coverage ratio of our algorithm in order to only utilize the high frequency of historical clicked information for helping following users.

Another experiment on 150 queries in Table VII, the result shown that our model gain better result. However, the

limitation of our model is that the coverage ratio of affected query is very lower than the baseline methods. In future works, we plan to investigate how to make high efficient use of large-scale click data in query reformulation.

TABLE VII: COMPARISON PERFORMANCE OF PARAMETER K WITH NDCG@K

Method	NDCG@k			
	$k = 1$	$k = 3$	$k = 5$	$k = 10$
OursModel	0.1688	0.3109	0.5384	0.5715
Embedding based	0.1207	0.1821	0.4647	0.5244
Pseudo-relevance based	0.1164	0.2420	0.5272	0.5263

† k we set to 1, 3, 5 and 10 respectively.

VI. CONCLUSIONS

In this paper, we propose a probabilistic model for query reformulation on large-scale click-through data, and moreover, we integrate query expansion and query fuzzy into a single framework. The basic idea is that we try to use historical knowledge derived from former users to help following users. We adopt fuzzy strategy of trivial terms to reformulate long query, rather than removing directly. Experiments show that the expanded terms are usually closely related to the original query. The expanded terms have two types of roles, the categorical term that represent the topic of query, and the functional term that often make user's intention more concrete and accurate. The reformulated query provide more enriched description than the original query, and naturally diversify the search results. In future works we will investigate query understanding by Generative Adversarial Networks (GAN), simulating the process of how to yield a good query.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions.

REFERENCES

[1] H. Li and J. Xu, "Semantic matching in search," *Foundations and Trends in Information Retrieval*, vol. 7, no. 5, pp. 343–469, 2014.

[2] C. D. Manning, P. Raghavan, and H. SchÃijtze, *Introduction to Information Retrieval*, Cambridge University Press, Cambridge, MA, 2008.

[3] K. Kinley, D. Tjondronegoro, H. Partridge, and S. Edwards, "Human-computer interaction: The impact of users' cognitive styles on query reformulation behaviour during web searching," in *Proc. the 24th Australian Computer-Human Interaction Conference*, Melbourne, Australia, November 2012, pp. 299-307.

[4] J. Huang and E. N. Efthimiadis, "Analyzing and evaluating query reformulation strategies in web search logs," in *Proc. the 18th ACM Conference on Information and Knowledge Management*, Hong Kong, China, November 2009, pp. 77-86.

[5] V. Dang and B. W. Croft, "Query reformulation using anchor text," in *Proc. the Third ACM International Conference on Web Search and Data Mining*, New York, USA, February 2010, pp. 41–50.

[6] C. Carpineto and G. Romano, "A survey of automatic query expansion in information retrieval," *ACM Computing Surveys (CSUR)*, vol. 44, no. 1, pp. 1–50, January 2012.

[7] Q. Liu, H. Huang, J. Lu, Y. Gao, and G. Q. Zhang. "Enhanced word embedding similarity measures using fuzzy rules for query expansion," in *Proc. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 571–578.

[8] M. L. Lu, X. B. Sun, S. W. Wang, D. Lo, and Y. C. Duan, "Query expansion via wordnet for effective code search," in *Proc. IEEE 22nd International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Montreal, QC, Canada, March 2015, pp. 545–549.

[9] J. Ooi, X. Q. Ma, H. W. Qin, and S. C. Liew, "A survey of query expansion, query suggestion and query refinement techniques," in *Proc. 4th International Conference on Software Engineering and Computer Systems*, Kuantan, Pahang, Malaysia, 2015.

[10] C. Y. Xiong and J. Callan, "Query expansion with freebase," in *Proc. the 2015 International Conference on The Theory of Information Retrieval*, Northampton, Massachusetts, USA, 2015, pp. 111–120.

[11] J. Singh and A. Sharan, "A new fuzzy logic-based query expansion model for efficient information retrieval using relevance feedback approach," *Neural Computing and Applications*, vol. 28, no. 9, pp. 2557–2580, 2017.

[12] M. Almasri, C. Berrut, and J.-P. Chevallet, "A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information," in *Proc. European Conference on Information Retrieval*, Springer, Cham, March 2016, pp. 709–715.

[13] S. Kuzi, A. Shtok, and O. Kurland, "Query expansion using word embeddings," in *Proc. the 25th ACM International on Conference on Information and Knowledge Management*, Indianapolis, IN, USA, 2016, pp. 1929–1932.

[14] F. Diaz, B. Mitra, and N. Craswell, "Query expansion with locally-trained word embeddings," in *Proc. the 54th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2016, vol. 1, pp. 367–377.

[15] D. Roy, D. Paul, M. Mitra, and U. Garain, "Using word embeddings for automatic query expansion," in *Proc. SIGIR Workshop on Neural Infor- mation Retrieval*, Pisa, Italy, July 2016.

[16] Y. Q. Liu, J. W. Miao, M. Zhang, S. P. Ma, and L. Y. Ru, "How do users describe their information need: Query recommendation based on snippet click model," *Expert Systems with Applications*, 2011, vol. 38, no. 11, pp. 13847–13856.

[17] N. Balasubramanian, G. Kumaran, and V. R. Carvalho, "Exploring reductions for long web queries," in *Proc. the 33rd international ACM SIGIR conference on Research and development in information retrieval*, , Geneva, Switzerland, July 2010 pp. 571–578.

[18] D. Odijk, R. W. White, A. H. Awadallah, and S. T. Dumais, "Struggling and success in web search," in *Proc. the 24th ACM International Conference on Information and Knowledge Management*, Melbourne, Australia, 2015.

[19] Sogouq. (May 10, 2018). [Online]. Available: http://www.sogou.com/labs/resource/q.php

[20] M. Sloan, H. Yang, and J. Wang, "A term-based methodology for query reformulation understanding," *Information Retrieval Journal*, 2016.

[21] M. Mottin, F. Bonchi, and F, Gullo, "Graph query reformulation with diversity," in *Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, NSW, Australia, August 2015, pp. 825-834.

[22] R. Pramanik, S. Pal, and M. Chakraborty, "What the user does not want?: Query reformulation through term inclusion-exclusion," in *Proc. the Second ACM IKDD Conference on Data Sciences*, Bangalore, India, 2015, pp. 116–117.

[23] L. D. Bing, W. Lam, T.-L. Wong, and S. Jameel, "Web query reformulation via joint modeling of latent topic dependency and term context," *ACM Transactions on Information Systems*, vol. 33, no. 2, 2015.

[24] M. Sloan, H. Yang, and J. Wang, "Web query reformulation via joint modeling of latent topic dependency and term context," *Information Retrieval Journal*, vol. 18, no. 2, pp. 145–165, 2015.

[25] Y. S. Wang, H. Y. Huang, and C. Feng, "Query expansion based on a feedback concept model for microblog retrieval," in *Proc. the 26th International Conference on World Wide Web*, Perth, Australia, 2017, pp. 559–568.

[26] X. H. Wang and C. X. Zhai, "Mining term association patterns from search logs for effective query reformulation," in *Proc. the 17th ACM Conference on Information and Knowledge Management*, Napa Valley, California, USA, October 2008, pp. 479–488.

[27] R. Nogueira and K. Cho, "Task-oriented query reformulation with reinforcement learning," in *Proc. Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, September 2017, pp. 585–594.

[28] C. Buck, J. Bulian, M. Ciaramita, A. Gesmundo, N. Houlsby, W. Gajewski, and W. Wang, "Ask the right questions: Active question reformulation with reinforcement learning," in *Proc. International Conference on Learning Representation*, Vancouver, Canada, May 2018.

[29] M. Thulin, "The cost of using exact con dence intervals for a binomial proportion," *Electronic Journal of Statistics*, vol. 8, no. 1, pp. 817–840, 2014.

[30] Wallis and A. Sean, "Binomial confidence intervals and contingency tests: mathematical fundamentals and the evaluation of alternative methods," *Journal of Quantitative Linguistics*, vol. 20, no. 3, pp. 178–208, 2013.
[31] J. Pennington, Richard Socher, and Christopher D. Manning, "Glove: Global vectors for word representation," *Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.
[32] D. Zhou, X. Wu, and W. Y. Zhao, "Query expansion with enriched user profiles for personalized search utilizing folksonomy data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 7, pp. 1536–1548, July 2017.
[33] Google Code, *Archive-Long-term Storage for Google Code Project Hosting*, 2018.

Wei Pang is currently a Ph.D. student. He studies at the Department of Computer Science, Beijing University of Posts and Telecommunications, China. His research interests include natural language understanding, information retrieval, and data science.

Junping Du is a professor at the School of Computer Science, Beijing University of Posts and Telecommunications, China. She is also a Ph.D. tutor. Her research interests include artificial intelligence, data mining and social network.