Mining User's Location Intention from Mobile Search Log

Yifan Sun[1], Xin Li[1], Lin Li[2], Qi Liu[1], Enhong Chen[1](✉), and Haiping Ma[3]

[1] University of Science and Technology of China, Hefei, China
{sunyifan,leexin,}@mail.ustc.edu.cn, {qiliuql,cheneh}@ustc.edu.cn
[2] Wuhan University of Technology, Wuhan, China
cathylilin@whut.edu.cn
[3] IFLYTEK Co., Ltd, Hefei, China
mhp0814@mail.ustc.edu.cn

Abstract. Much attention has been paid to web search personalization and query optimization over the past decade. With the prevalence of smart phones, the mobile search results for the same query may vary in regard to the user's location. In order to provide more precise results for users, it's essential to take geographic location into account along with the user's input query. In this paper, we try to identify queries that have location intentions. For example, query "weather forecast" has a location intention of local city while "The Statue of Liberty" has a location intention of "New York city". To identify the location intention behind a query, we propose a novel method to extract a set of features and use neural network to classify queries. In the classification of queries without explicit location names, our experiment shows that our approach achieves 82.5% at F1 measure and outperforms baselines by 4.2%.

Keywords: Feature selection · Location intention · Query classification

1 Introduction

With the prevalence of smart phones and intelligent personal assistant such as Siri, the market share of mobile search has occupied half of the whole search market[1]. Thus, it becomes crucial to have an eye on the mobile search field. Conventional web search engines characterized by "one size fits all" provide the same results for the same keyword queries even though these queries from different users may contain different intentions. According to Welch's research [1], about 50% of web search queries with an intention of requesting local information, do not have explicit location names. If we can automatically identify queries that have a location intention, we can provide better user experience by saving user's time and reducing interaction times.

Here is a toy example. Some queries, such as "bus terminal" and "house price", have location intentions, but some other queries, such as "funny videos"

[1] http://ir.baidu.com/phoenix.zhtml?c=188488&p=irol-reportsAnnual

S. Zhang et al. (Eds.): KSEM 2015, LNAI 9403, pp. 407–420, 2015.
DOI: 10.1007/978-3-319-25159-2_37

and "jokes" do not. If a supplementary location name is added to the front ones, it might help the search engine to understand the user's intention and thus return more precise results. We term these two kinds of queries "location sensitive query" and "ordinary query" respectively in order to make consistency throughout this paper. Besides these two kinds of queries, there is also another scenario which is termed "fixed collocation query". When a user keys in "The Statue of Liberty", he most probably means the statue in New York city. This usually happens when users search for scenic spots, famous universities and other well-known places of interest.

In real life, if we can know a query is a "fixed collocation query" before searching, we can refine the query by adding a corresponding place name to it, if we can know the query is a "location sensitive query", we can utilize the locate function embbeded in cellphones and get location information to improve searching results. However, to classify queries according to its location intention is not a trivial problem. To achieve these goals, we face two challenges. Firstly, queries submitted to the search engine usually contain very short keywords. These keywords are insufficient to reveal user's real intention [2]. Secondly, in text classification, the demension of the feature space is very high. In this paper, we regard the problem of identifying the user's location intention as a classification problem, and we use our novel feature selection method and neural network classifier to achieve a high accuracy in the classification of "fixed collocation queries" and "location sensitive queries", which are two tasks we aim to address in this paper.

We conduct extensive experiments on a real-world dataset of mobile search log in comparison with five baseline methods. The results show the superiority of our method. In summary, we make the following contributions:

1) We propose a novel feature selection method in the classification of "location sensitive queries".

2) We devise a score function to measure the relevance between a word and a place name and this function is used to identify "fixed collocation queries".

3) The experiments on a real world dataset of mobile search log show that our approach outperforms the baseline methods.

The paper is organized as follows: Section 2 reviews some related works. Section 3 and Section 4 present our approaches to identify fixed collocation queries and location sensitive queries respectively. Section 5 describes our experiment results along with discussion. In the last section we conclude our work and point out possible directions for future work.

2 Related Work

In this section, we review the related work in personalized web search and user's intention recognition.

Personalized Web Search. Considerable work has been done in the field of personalized web search. Bennett et. al [3] investigated how short-term and long-term user behavior interact, and how they can be used to personalize search results. Matthijs and Randlinski [4] collected user's browsing history via a browser add-on and used the language model to analyze the captured pages for personalizing search results. Xiang et. al [5] analyzed the user's context and used those contextual features to rank the results of subsequent queries. Kharitonov and Serdyukov [6] used user's age and gender for re-ranking and personalizing search results. Teevan et. al [7] analyzed the re-visitation pattern of users and classified at least 15% of all clicks as personal navigation in which the user repeatedly searches for the same page. In the field of query refining or suggestion, Bhatia et. al [8] mined frequently occurring phrases and n-grams from text collections and deployed them for generating and ranking auto-completion candidates. Santos et. al [9] extracted queries that frequently co-appeared in the same sessions to generate query suggestions. Ozertem et. al [10] presented a learning-to-rank framework for ranking query suggestions. What differentiates our work from the previous ones is that we mine "the wisdom of the crowd" from the mobile search logs. It does not require a particular user's behavior data or contextual information to be collected. After the identification of fixed collocation queries and classification of location sensitive queries, the benefits can be enjoyed by all users even if we do not know any of a particular user's information.

User's Intention Recognition. Different classification schemes have been proposed to categorize user's intention behind his search. Lee et. al [11] presented a set of features to automatically classify user's intention as either navigational or informational. Yi [12] and Kamvar [13] categorized the mobile queries into a taxonomy with a total of 23 top-level predefined categories which covers most of the areas in the information space. Chuklin [14] proposed a way to model user's intention distribution and bias due to different document presentation types. Dhar [15] utilized semi-supervised learning and user's previous search log to classify query intentions.

Welch [1], Vadrevu [16] and Gravano [17] exploited classification techniques to categorize queries according to their geographic intentions, which are much related to our work. In Vadrevu's work [16], they relied on query term co-occurrence in query logs and built three classifiers to identify regional sensitive queries. However, their regional sensitive queries is coarse-grained, such as "U.S.A.", "Japan" and "India". In contrast, our location sensitive queries have a hierarchy of three levels: provinces, cities, and counties, which is more practical. Gravano [17] defined a categorization scheme for queries where they represented queries by features and used several classifiers to determine query's location intention. In Welch's work [1], a tagging technique and different features extracted from query logs are combined to classify queries. Several supervised classifiers were tested. Both of their experiments get a precision at 90% with a recall less than 50%. Ourdia [18] classified queries into three classes using Kurtosis and Kullback-Leibler Divergence measures, and their experiment on a dataset containing 200 queries achieved the F1 measure of 0.800. Different from their works, we propose our novel method to extract a set of features and use neural network to identify the location intention of a query and our experiments on a real world dataset which contains 1,000 queries get the F1 measure of 0.825. We also implement methods proposed by Welch [1] and Ourdia [18] as baselines and classification results show that our method outperforms theirs.

3 Identify Fixed Collocation Queries

Fixed collocation queries always occur with a corresponding place name. In this section, we will illustrate how to identify such kind of queries.

3.1 Data and Preprocess

The mobile search log containing 1,402,744 mobile search queries in Chinese character is provided by IFLYTEK company [2]. As Chinese characters do not have tense or any other form variation, there is no need of stemming or normalizing. When implementing word segmentation and stop words elimination, we adopt two open source packages, i.e., Lucene [3] and IKAnalyzer [4].

3.2 Fixed Collocation Queries Identification

We build a dictionary of 3,223 names of places in China, including all 34 province names, all 333 city names, and all 2,856 county names. We consider co-occurrence frequency and term frequency as factors and devise a score function to identify fixed collocation queries.

Here is an example to show how our approach works. We plot two figures in Fig. 1, where Fig. 1 (a) represents the co-occurrence frequency distribution of keyword "Tian'an men" over a set of province names and Fig. 1 (b) represents the co-occurrence frequency distribution of keyword "sight spot" over a set of province names as well.

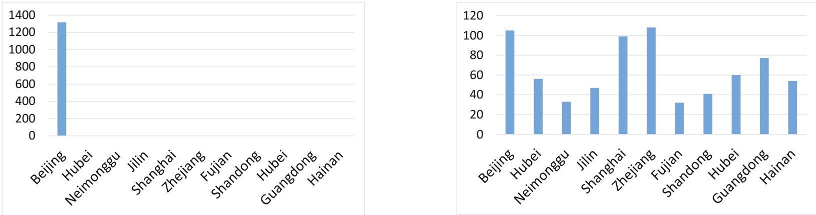
From Fig. 1 (a), there are a total of 1,334 queries that "Tian'anmen" co-occurred with a province name, and within which 1,317 queries have the word "Beijing". From Fig. 1 (b), there are 712 times that "sight spot" co-occurred with a province name, and we can see that the distribution is much more uniform.

There are many metrics that can represent a distribution, such as variance and Kurtosis measure. We use co-occurrence frequency as criterion because in variance and Kurtosis measures, there is a precondition that the results have a center point. These two criterions are used to measure how much data is gathered to the center point or how far the numbers are spread out. They are more suitable for continuous number distribution. In this problem, there is no center point. Thus, we propose a function including term frequency and co-occurrence frequency to find out fixed collocation queries.

$$f(Term_i) = \max_{1<j<N_i} \{\frac{P_{ij}}{N_i}\}. \tag{1}$$

$$Score_i = (TF_i - \alpha) \times f(Term_i). \tag{2}$$

[2] http://iflytek.com/
[3] http://lucene.apache.org/
[4] https://code.google.com/p/ik-analyzer/



(a) Co-occurrence frequency distribution of "Tian'an men". over province names

(b) Co-occurrence frequency distribution of "sight spot". over province names

Fig. 1. Distribution of keywords "Tian'an men" and "sight spot".

To each word $Term_i$, TF_i is its occurrence times. N_i is the times $Term_i$ occurs with a place name. P_{ij} represents the times $Term_i$ occurs with a place name j. Function $f(Term_i)$ measures the max proportion of a term with a place name. To ensure accuracy, we empirically choose 0.9 as the threshold of $f(Term_i)$. We preserve those words whose $f(Term_i)$ is larger than 0.9 and judge whether its occurrence times is larger than a threshold α. Only when the both conditions are satisfied, we regard it as a fixed collocation query. In the experiment, we will futher discuss the effect of threshold α.

4 Classify Location Sensitive Queries

We adopt conventional classification techniques to classify location sensitive queries upon which we propose our feature selection method.

4.1 Feature Selection Methods

In text classification, there will be a large number of features in training and testing within which there is only a small proportion that is essential. Thus, how to select the best features becomes a key issue. We propose our novel feature selection method and use five baselines for comparison. These baselines are Document Frequency (*DF*), Information Gain (*IG*), Chi-Square Test (*CHI*), Expected Cross Entropy (*ECE*) and Mutual Information (*MI*).

Document Frequency is the number of documents in which a term occurs. The assumption is that rare terms are non-informative for category prediction and will not influence global performance. It is the simplest method and in our experiment, we select top K words as features in terms of the *DF* value.

Information Gain is frequently employed as a term goodness criterion in the field of machine learning [19,20]. It measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document. For term t and class c_i, M is the number of classes:

$$IG(t) = -\sum_{i=1}^{M} P(c_i)logP(c_i) + P(t)\sum_{i=1}^{M} P(t|c_i)logP(t|c_i) + P(\bar{t})\sum_{i=1}^{M} P(\bar{t}|c_i)logP(\bar{t}|c_i).$$

In our experiment, we select top K words in terms of the IG value.

Mutual Information is a criterion commonly used in statistical language modelling of word associations [21,22]. If we considers the two way contingency table of a term t and a category c, where A is the number of times t and c occur, B is the number of times t occurs without c, C is the number of times c occurs without t, and N is the total number of documents. Then the mutual information criterion between t and c is defined to be:

$$MI(t,c) = log\frac{P(t \wedge c)}{P(t) \times P(c)} \approx log\frac{A \times N}{(A+C) \times (A+B)}.$$

MI(t,c) has a natural value of zero if t and c are independent. In our experiment, we select top K words in terms of the *MI* value.

λ^2 **Statistic** (*CHI*) measures the lack of independence between t and c and can be compared to the λ^2 distribution with one degree of freedom to judge extremeness. All the notations have the same definitions as before and D is the number of times both t and c do not occur. The λ^2 measure is defined as:

$$\lambda^2 = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)}.$$

The λ^2 statistic has a value of zero if t and c are independent. The weakness of the λ^2 statics is not to be reliable for low-frequency terms [23]. In our experiment, we select top K words in terms of the λ^2 value.

Expected Cross Entropy between two probability distributions over the same underlying set of events measures the average number of bits needed to identify an event drawn from the set [24]. In information theory, for term t and class c_i, the calculation is as follows:

$$ECE(t) = p(t)\sum_{i=1}^{M} p(t|c_i) \times log\frac{p(t|c_i)}{p(c_i)}.$$

In our experiment, we select top K words in terms of the ECE value.

In summary, apparently the *DF* measure is in favor of common terms over rare terms. It is not necessarily true in *IG* or *CHI* by definition. In theory, a common term can have a zero-valued *IG* or λ^2 score. It is proved by researches in past decades that the top three methods that get the best accuracy are λ^2 statistic, *DF*, and *IG*. However, they all have their own weaknesses.

4.2 Our Feature Selection Method

Our method is based on fuzzy set theory. Fuzzy sets are those whose elements have degrees of membership, which are introduced by Lotfi A. Zadeh [25] in 1965 as an extension of the classical notion of set. We mainly devise the membership function and the final score function.

Fuzzy Entropy. Information theory is concerned with quantification of information which is defined as the amount of information conveyed in an event and depends on the probability of the event. The definition is as follows:

$$I(A) = -logP(A).$$

The average information over all events is called the entropy. It is usually called Shannon entropy if it refers to the classical information entropy:

$$H(X) = -\sum_{k=1}^{n} P_k log P_k,$$

where X is a set of random variables and P_k is the set of all probabilities for the variables in X. $P_k = P[X = X_k]$, where $k = 1, 2, ..., n$.

The fuzzy entropy proposed by De Luca and Terminal [26] is shown in equation below. It is defined based on the concept of membership function where there are n membership functions (μ_i).

$$H_A = -K \sum_{i=1}^{n} \{\mu_i log(\mu_i) + (1 - \mu_i) log(1 - \mu_i)\}.$$

Membership Function Design. The design of membership function is the key point in calculation of fuzzy entropy. In short text classification, we consider two occasions as follows:

1) If one term occurs in one class frequently and seldom occurs in other classes, apparently it is a good feature.

2) In a given class, if one term spreads widely in many sentences or instances, it is a better feature than the one that only occurs in several instances.

The membership function is designed as follows:

$$\mu_{c_i}(t) = 4 \times (\frac{tf_{it}}{tf_t} - 0.5) \times (\frac{d_{it}}{C_i} - \frac{d_t}{N}), \tag{3}$$

where tf_{it} represents the number of times term t occurs in class c_i, tf_t represents the number of times t occurs in all classes, C_i is the total number of documents that belong to class c_i, d_t is the number of documents that contain t, d_{it} is the number of documents that contain t in class c_i and N is the total number of documents.

If a term follows the uniform distribution, the two parts in brackets both get a zero. From the definition, we can see $\frac{tf_{it}}{tf_t} \leq 0.5$, and $\frac{d_{it}}{C_i} - \frac{d_t}{N} \leq 0.5$. In order to make the maximum value equal to 1, we multiply them by 4.

Fuzzy Entropy Calculation. In regard to the definition of fuzzy entropy, we calculate our fuzzy entropy as:

$$FE(t) = -\frac{1}{m}\sum_{i=1}^{m}[\mu_{c_i}(t)log\mu_{c_i}(t) + (1-\mu_{c_i}(t))log(1-\mu_{c_i}(t))]. \tag{4}$$

$FE(t)$ means the fuzzy entropy of term t, $\mu_{c_i}(t)$ is the membership between term t and class c_i, and m is the number of classes.

Final Score Function. As we know, different feature selection methods have different emphases and drawbacks. *CHI* method does not take the term frequency into account and has a preference to low term frequency words. In order to overcome this weakness, we use tf_{it}/tf_t to represent the term frequency. In binary classification, tf_0 means the term frequency in negative class and tf_1 means the term frequency in positive class. tf_t is the sum of them. We use

$$ICHI(t) = Max\{tf_0/tf_t, tf_1/tf_t\} \times CHI(t), \tag{5}$$

as the improved *CHI* results. After calculation, we get the results sets of *ICHI(t)* and *FE(t)* and normalize each result set to [0, 1]. Then, we combine normalized *FE(t)* with *ICHI(t)* by a parameter β as the final score function:

$$FEICHI(t) = \beta \times Norm\{ICHI(t)\} + (1-\beta) \times Norm\{FE(t)\}, \tag{6}$$

where $0 \leq \beta \leq 1$.

The description of our algorithm is shown in algorithm 1:

Algorithm 1. *FEICHI* feature selection method

Input: a) D={ $q_1, q_2, ..., q_N$ } be a set of N training set queries
b) Two predefined classes, C= $\{c_1, c_2\}$
c) T= { $t_1, t_2, ..., t_n$ } is the set of n terms in the vocabulary
d) K is a threshold on the number of terms to be selected, $\beta \in$ [0,1]
Output: A set of reduced terms TR

Steps:

TR $\leftarrow \emptyset$
for each $t_i \in T$ **do**
  Calculate $FEICHI(t_i)$ according to equation (6)
end for
Sort $FEICHI(t_i)$, $\forall$ $t_i \in$ T in descending order and the corresponding order of terms are $tr_1, tr_2, ..., tr_n$
for i $\leftarrow$ 1 to K **do**
  TR $\leftarrow$ TR $\cup$ tr_i
end for
return TR

4.3 Classification Schema

Many classifiers can be chosen such as naive bayes, linear regression, support vector machine, decision tree and neural network [27]. After feature selection, we test all these classifiers on various datasets and the results of neural network are more stable and reliable, so we choose neural network as our classifier.

5 Experiment Results

We evaluate the effectiveness of our feature selection method on a real world mobile search log provided by IFLYTEK [5]. Our experiments are five-fold.

1) The identification of fixed collocation queries is shown in *Exp. 1*.

2) We evaluate the performance of our novel feature selection method in comparison with five baseline methods, which will be analyzed in *Exp. 2*.

3) The performance of our method to identify location sensitive queries, along with the comparison to two state-of-the-art methods, is illustrated in *Exp. 3*

4) The influence of feature set size K is explored in *Exp. 4*.

5) We also analyze the effect of training set data size to the performance of our method, which is shown in *Exp. 5*.

5.1 Description of Dataset

Our dataset is approximately 950 MegaBytes and contains 1,402,744 mobile search queries which are in Chinese character, where 92,438 queries contain an explicit place name. In *Exp. 1*, we use these 92,438 queries as training set. In the next four experiments, we select 3,000 queries which contain an explicit place name as positive training set and adopt filter method to get negative training set. We build a dictionary that contains words related to location sensitive queries such as "where", "nearby" and "nearest". After filteration and selection, we ask 10 persons who are in master degree to examine the negative training set and pick out the false ones. Eventually we get a negative training set and a positive training set, each containing 3,000 queries. We randomly select 3,000 balanced queries as dataset 1 and split the remaining data into dataset 2 which contains 2,000 balanced queries and dataset 3 which contains 1,000 balanced queries. We public our datasets which can be downloaded from this link [6].

5.2 Evaluation Metrics

In order to evaluate the effectiveness of class assignments, we use the standard precision, recall and F1 measure. The definitions are as follows:

$$\text{precison} = \frac{\text{number of correct positive predictions}}{\text{number of positive predictions}}$$

[5] http://iflytek.com/

[6] http://pan.baidu.com/s/1mgmV3cs

$$\text{recall} = \frac{\text{number of correct positive predictions}}{\text{number of positive examples}}$$

$$\text{F1} \quad \text{measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall } + \text{ precision}}$$

These scores are computed for the binary decisions on each individual class and then are aggregately averaged over all classes. A good algorithm should produce as high a recall value as possible without sacrificing precision. The closer the values of precision and recall are, the higher the F1 measure is. The value of F1 measure lies between 0 and 1 and a high value of F1 measure is desirable for good classification.

5.3 Exp. 1: Identify Fixed Collocation Queries

We use the 92,438 queries as the training data and function $Score_i$ proposed in section 3 as the criterion. As the value of threshold α changes, the number of fixed collocation queries as well as the classification accuracy, changes too, which is shown as Fig. 2.

When the term frequency is less than 5, we can see that there are many incorrect pairs. A word with a very low frequency happens to co-occur with a place name. As α increases, the reliability of the results increases as well. When α surpasses 80, accuracy begins to decline because more true fixed collocation queries are ignored than false ones. We manually check the results in terms of the accuracy and choose 80 as the threshold. Thus, we get an accuracy of 91%. We list several fixed collocation queries in Table 1.

Table 1. Queries and corresponding place names

Query	Place name
Guiyuan temple	Wuhan city
Yu Opera	Henan province
Lanzhou	Gansu province
Zhuizi	Henan province
Greeting Pine	Huangshan city
Panfu Road	Guangzhou city
The Classical Gardens	Suzhou city
Roast duck	Beijing city
The Captial	Beijing city
Chongqing University	Chongqing city
Guangzhou Daily	Guangzhou city
Bangzi	Hebei province
Daqing Oil Field	Daqing city

Fig. 2. The relation among threshold α, number of pairs left, and classification accuracy

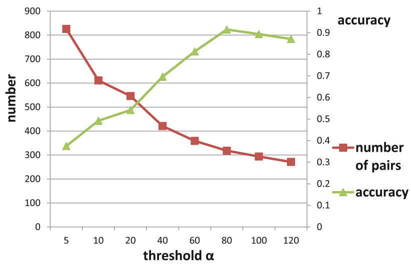


Table 2. Feature selection methods and classifiers of F1 measure with top 1,000 features

	NB [a]	Liblinear [b]	SVM	RBFNetwork	Tree
DF	0.692	**0.730**	0.544	0.704	0.695
IG	**0.498**	0.485	0.486	0.484	0.486
CHI	0.700	**0.753**	0.539	0.711	0.695
MI	0.537	0.572	0.486	**0.581**	0.494
ECE	**0.499**	0.485	0.480	0.484	0.486
FEICHI	0.703	**0.767**	0.54	0.757	0.696

[a] Naive Bayes Classifier
[b] Linear Regression Classifier

Table 3. Classification results of 1,000 queries

Method	Precision	Recall	F1
Ourdia et. al	0.805	0.797	0.783
Welch et. al	0.911	0.592	0.718
Our work	0.849	0.828	**0.825**

5.4 Exp. 2: Comparison of Feature Selection Methods

Dataset 1 is used in this experiment. Firstly, we tag the sentences in positive training set with 1 and negative training set with 0 as the class label. Secondly, we get the segmentation form of each sentence and remove the place names from each sentence. Thirdly, we utilize different feature selection methods to get top 1,000 features and then use multiple classifiers to get the results. We implement six feature selection methods and use classification tool Weka [7] to get the results. The value of β is set to 0.8 through cross validation. The results are shown in Table 2, in which the first column represents the feature selection methods and the first row represents the classifiers.

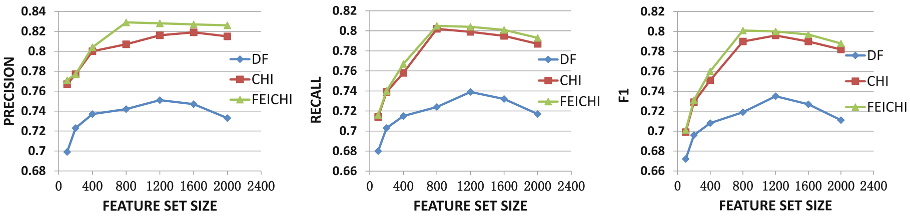
From Table 2, *CHI* method gets the best F1 score of 0.753 among baselines and *DF* method gets 0.73 at F1 score. When using *CHI* and *DF* methods, the linear regression and neural network classification methods outperform the others. Our method gets the highest score of 0.767 due to the reason that we take the weakness of *CHI* into account and combine fuzzy entropy method to help improve the performance. Both *IG* and *ECE* methods do not perform well, which indicate that these two methods are not suitable for short text classification.

By comparing the classifiers only, both *SVM* and *Tree* models never get the best classification results. Linear regression model gets three best results, which follows our intuitions. Sentence is the linear combination of words, each making a different contribution to the sentence, so it is appropriate to use linear regression in text classification. On the contrary, *SVM* and *Tree* models can not describe the characteristics or the structure of sentences, which leads to their poor F1 results in query classification.

5.5 Exp. 3: Comparison of Location Sensitive Queries Classification

In the detection of location sensitive queries, Ourdia [18] uses Kurtosis and Kullback-Leibler Divergence to measure the relevance between a query and a place names while utilizing *SVM* model for classification on a dataset containing

[7] http://www.cs.waikato.ac.nz/ml/weka/



(a) Precision (b) Recall (c) F1 measure

Fig. 3. Precison, Recall and F1 measure of three feature selection methods with different feature set size K.

200 queries. F1 score reported is 0.8. Welch [1] proposes "base queries" concept and utilizes clustering method to find base queries. Welch's experiment on 102 queries gets a precision of 0.94, however, the recall is 0.46. In summary, both of their F1 scores are less than 80%.

We evaluate our method along with their methods on dataset 3 with K set to 1,600. β is set to 0.5 through cross validation. The experiment results are shown in Table 3. We can see that our method achieves 82.5% at F1 score which outperforms the aforementioned two methods.

5.6 Exp. 4: Influence of Feature Set Size K

We vary feature set size from 50 to 2,000 and implement our method and two baseline methods on dataset 2. Results show that under any circumstances our method performs the best. We use neural network classifier on dataset 2 by setting β to 0.8 gained from *Exp. 2* and show the precision, recall and F1 value in Fig. 3.

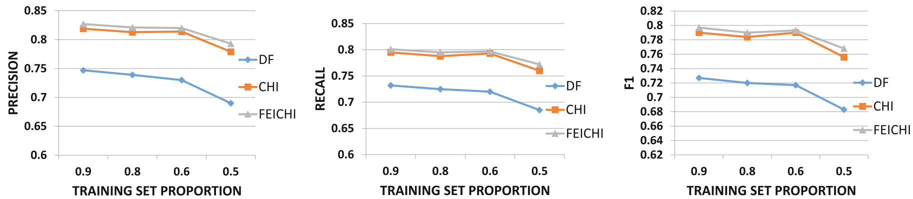
As the feature set size K increases from 100 to 2,000, all of the curves first rise and then fall. It agrees with our expected results and priori knowledge. In the rising stage, more and more good features are selected for classification and in the falling stage, more and more irrelevant features are selected which make the classification result worse.

From Fig. 3, it is clearly shown that our method continuously outperforms the baselines. When the size of feature set is small, the improvement is not apparent. However, when K rises between 800 and 1,600, there is a significant improvement of 3% in precision.

5.7 Exp. 5: Influence of Training Set Proportion

To find the influence of the split percentage of training set and testing set, we evaluate our method and baseline methods on dataset 2. We still choose $\beta = 0.8$ and neural network as classifier, the results are shown in Fig. 4.

When the proportion of training set decreases, the precision, recall and F1 measure tend to decline. Our method still outperforms the others when the training set proportion varies from 0.9 to 0.5. In summary, our *FEICHI* feature selection method does have an improvement over baselines.



(a) Precision (b) Recall (c) F1 measure

Fig. 4. Precison, Recall and F1 measure of three feature selection methods with different training set proportion.

6 Conclusion

In this paper, we propose an approach for identifying fixed collocation queries via "the wisdom of the crowd" and location sensitive queries via *FEICHI* and neural network. Specifically, we devise a score function to identify queries with a fixed corresponding place name. We propose our *FEICHI* feature selection method and get a better performance than the five baseline methods. We utilize neural network classifier and achieve 82.5% at F1 measure on the queries that have implicit location intentions. All the experiments are conducted on a real-world mobile search log. In future we plan to study how to identify the intentions of ambiguous search queries.

Acknowledgments. This research was partially supported by grants from the National High Technology Research and Development Program of China (Grant No. 2014AA015203), the Fundamental Research Funds for the Central Universities of China (Grant No. WK2350000001) and the Anhui Provincial Natural Science Foundation (Grant No. 1408085QF110).

References

1. Welch, M., Cho, J.: Automatically identifying localizable queries. In: Proceedings of the ACM SIGIR, pp. 1185–1186 (2008)
2. Kamvar, M., Baluja, S.: Deciphering trends in mobile search. Computer **40**(8), 58–62 (2007)
3. Bennett, N., Radlinski, F., White, R.W., Yilmaz, E.: Inferring and using location metadata to personalize web search. In: Proceedings of the SIGIR, pp. 135–144 (2011)
4. Matthijs, N., Radlinski, F.: Personalizing web search using long term browsing history. In: Proceedings of the WSDM, pp. 25–34 (2011)
5. Xiang, B., Jiang, D., Pei, J., Sun, X., Chen, E., Li, H.: Context-aware ranking in web search. In: Proceedings of the SIGIR, pp. 451–458 (2010)
6. Kharitonov, E., Serdyukov, P.: Demographic context in web search re-ranking. In: Proceedings of the CIKM, pp. 2555–2558 (2012)

7. Teevan, J., Liebling, D.J., Ravichandran Geetha, G.: Understanding and predicting personal navigation. In: Proceedings of the WSDM, pp. 85–94 (2011)
8. Bhatia, S., Majumdar, D., Mitra, P.: Query suggestions in the absence of query logs. In: Proceedings of the SIGIR, pp. 795–804 (2011)
9. Santos, R.L.T., Macdonald, C., Ounis, I.: Learning to rank query suggestions for adhoc and diversity search. Information Retrieval, 1–23 (2012)
10. Ozertem, U., et al.: Learning to suggest: a machine learning framework for ranking query suggestions. In: Proceedings of the SIGIR, pp. 25–34 (2012)
11. Lee, U., Liu, Z., Cho, J.: Automatic identification of user goals in web search. In: Proceedings of the WWW, pp. 391–400 (2005)
12. Yi, J., Maghoul, F., Pedersen, J.: Deciphering mobile search patterns: a study of Yahoo! mobile search queries, In: Proceedings of the WWW, pp. 257–266 (2008)
13. Kamvar, M., Baluja, S.: A large scale study of wireless search behavior: Google mobile search. In: Proceedings of the AIGCHI, pp. 701–109. ACM (2009)
14. Chuklin, A., Serdyukov, P., de Rijke, M.: Using intent information to model user behavior in diversified search. In: Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E., Serdyukov, P. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 1–13. Springer, Heidelberg (2013)
15. Dhar, S., Swain, S., Mishra, B.S.P.: Query intent classification using semi-supervised learning. In: Proceedings of the IJCA, pp. 40–43 (2014)
16. Vandrevu, S., Zhang, Y., Tseng, B., Sun, G., Li, X.: Identifying regional sensitive queries in web search. In: Proceedings of the WWW, pp. 507–514 (2008)
17. Gravano, L., Hatzivassiloglou, V., Lichtenstein, R.: Categorizing web queries according to geographical locality, In: Proceedings of the CIKM, pp. 325–323 (2003)
18. Bouidghaghen, O., et al.: Personalizing mobile web search for location sensitive queries. In: Mobile Data Management, pp. 110–118. IEEE (2011)
19. Mitchell, T.: Machine Learning, pp. 36–39. McCraw Hill (1996)
20. Quinlan, J.R.: Induction of decision trees. Machine Learning **1**(1), 81–106 (1996)
21. Church, K.W., Hanks, P.: Word association norms, mutual information and lexicography. In: Proceedings of the ACL, pp. 76–83 (1989)
22. Wiener, E., Pedersen, J.O., Weigend, A.S.: A neural network approach to topic spotting. In: Proceedings of the DAIR, pp. 317–332 (1995)
23. Dunning, T.E.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics **19**(1), 61–74 (1993)
24. De Boer, P.-T., et al.: A tutorial on the cross-entropy method. Annals of Operations Research **134**(1), 19–67 (2005)
25. Zadeh, L.A.: Fuzzy sets. Information and Controls **8**(3), 338–353 (1965)
26. Luca, A.D., Terminal, S.: A definition of non probabilistic entropy in the setting of the fuzzy set theory. Information and Control **20**, 301–312 (1972)
27. Kotsiantis, S.B., et al.: Supervised machine learning: a review of classification techniques, pp. 3–24 (2007)