Table 1: **Summarization of collected dataset for different cities (partially presented due to page limit).**

City	Shanghai	Beijing	Chongqing	Guangzhou	Chengdu	Shenzhen	Nanjing	Tianjin	Xian	Shenyang
Users	183,239	69,412	6,787	42,694	10,788	9,462	12,760	10,137	7,867	6,015
Restaurants	34,127	27,120	6,948	18,625	12,074	9,931	8,570	7,349	7,695	5,268
Dining check-ins	2,582,914	895,120	89,749	542,563	152,227	135,362	167,826	120,862	86,651	65,841

Figure 6: A restaurant's description on DianPing

if a user resides in Beijing, it is actually difficult and meaningless to recommend restaurants in Shanghai if she does not visit Shanghai frequently. 2) We remove those users who have fewer than 10 dining check-ins, which is to ensure sufficient observable dining history. 3) For a city, if the number of users who reside in this city is smaller than 5,000 or the number of restaurants located in this city is smaller than 1000, we remove all the users who reside in this city and their related check-ins. The reason is that when we generate the Top-K novel restaurants for a user in Section 5, all the restaurant candidates are selected from the city he resides. For each city, the context-aware tensor factorization method is applied, respectively. Thus, we need enough users and restaurants in a city to ensure the user-restaurant-time tensor factorization.

After the filtering procedures, we eventually obtained a collection of 361,218 unique users from 21 cities all over China, with 4,941,060 dining check-ins. Table 1 summarizes the statistics of the final dataset for different cities.

Furthermore, since the restaurant in our SinaWeibo check-in dataset is only described by geographic coordinates and category, to obtain the restaurant attributes discussed in Section 5.2, we need to link external data sources. As shown in Figure 6, DianPing (similar to Yelp, the largest online review website in China) is such a desired website where the restaurants are described by sufficient attributes in detail. First, for each of the 21 cities from our SinaWeibo check-in dataset, we crawled all restaurants located in this city from DianPing by grabbing web pages directly, and extracted prices, ratings, tastes, environments and service qualities from the raw webpages. Then, we applied a two staged method described in [32] (including both geographic filtering and title string match) to map each restaurant from SinaWeibo to a restaurant from DianPing. Finally, we explicitly observed 500 paired results, where 91.5% of the pairs are correctly matched (the match precision is good enough for later process). In addition, we find that only 5.7% of restaurants from SinaWeibo can not be mapped to any restaurant from DianPing.

To evaluate the performance of our proposed methods, for each user, her previous 90% dining check-ins were used as training data and the remaining dining check-ins are used as testing data.

7.2 Experiments for Novelty-Seeking Inference

In this subsection, we study the performance of novelty-seeking status inference. First, for each constraint feature, we constructed the target distribution according to the empirical distribution in the training data. For example, if 60% of the dining check-ins generated by the male is novel and 40% is regular, the target distribution for constraint feature "male" is $p(s = 1|\text{male}) = 0.6, p(s = 0|\text{male}) = 0.4$. Next, we compared our method (CRF with model features, optimized with constraints), shortened as "**CRF(M)+C**" against the following baselines,

- **LR** (Logistic Regression): This algorithm uses both the model features and constraint features for the Logistic Regression.

- **CRF+C** (CRF with only constraints): This algorithm uses the same settings as **CRF(M)+C**, except that it does not incorporate model features.

- **CRF(M)** (CRF with only model features): This algorithm uses the same settings as **CRF(M)+C**, except that it does not incorporate constraints.

- **NSTM** (Novelty-Seeking Trait Model): This Bayesian model presented in [43] is a state-of-the-art method to model user's sequential behavior in consideration of novelty-seeking and preference. To accommodate our scenario, we set a user's novelty-seeking status as 0 or 1 (it is originally set from 1 to 5 in [43]).

Table 2: The results of novelty-seeking inference.

	Accuracy	True Positive Rate	False Negative Rate
CRF(M)+C	0.823	0.786	0.845
LR	0.641	0.593	0.667
CRF(M)	0.795	0.762	0.811
CRF+C	0.633	0.624	0.639
NSTM	0.748	0.767	0.719

If we consider novelty-seeking status $s = 1$ as positive and the other as negative, the results of true positive rate, false negative rate as well as accuracy are shown in Table 2, where accuracy indicates the percentage of correct prediction for all check-ins in the testing data, true positive rate indicates for these novel check-ins ($s = 1$) in the testing data, the percenrage of correct prediction, and false negative rate indicates for these regular check-ins ($s = 0$), the percenrage of correct prediction. It is clear that our method significantly outperforms competitors in all three criterions. For example, by using the constraints, our method achieves an improvement over **CRF(M)**. Compared our method with **CRF+C**, it is obvious that model features are significantly crucial to the detection of novelty-seeking status. We can see that **LR** performs worst due to the fact that this method does not take the sequential dependency of novelty-seeking statuses into consideration. In addition, the reason why **NSTM** is defeated by our method **CRF(M)+C** is that **NSTM** only considers a user's dining sequence, while our method incorporates various model features as well as users' socio-demographic characteristics as constraints. Since regular restaurant recommendation is easier than novel restaurant recommendation as shown in Section 7.3 and Section 7.4, to give a more accurate recommendation as a whole, we would expect that the False Negative Rate is as large as possible (if a user want to visit regular restaurants, we expect that the error of predicting her novelty-seeking status as $s = 1$ is as small as possible). In view of this, our method also outperforms these baselines evidently in the False Negative Rate criterion.