CIKM 2014 Competition:  
Second Place Solution  
Zhanpeng Fang and Jie Tang  
Department of Computer Science, Tsinghua University

**Task**
- Given a sequence of query sessions
  - Example
    - Class1 Query1 –
    - Class1 Query1 Title1
    - Class2 Query2 –
    - Class2 Query2 Title2
    - Class2 Query2 Title3
- Classify the class label of test queries

**Challenge**
- Encoding character
  - Only little prior knowledge can be used
- Heterogeneous data
  - Query, title, session information
- User search behavior
  - How to incorporate user search behavior to help the classification task?
- Unlabeled data
  - How to utilize the large scale unlabeled data?

**Result**
- 0.9245 (public) / 0.9245 (private)
- 2nd place winner
- Achieve in 4 days, from Sep. 27th to Sep. 30th EST

---

**Feature Extraction**

**Bag of word**
- Given a query Q
- One gram, two grams, last gram of Q
  - $0.9091 > 0.9105$
- One gram, two grams of the clicked titles
  - $0.8452 \rightarrow 0.9091$, top 12 in the leaderboard!
- Higher grams give a bit more improvement
- More bag of words features?
  - Queries in the same session of Q?
  - Titles in the same session of Q?
  - Performance decreases, $0.9091 \rightarrow 0.89x$

**Search behavior**
- Macro features
  - #total search, average length of clicked titles, length of the query
  - $0.9091 > 0.9105$
- Session class features
  - For each potential class C, calculate:
    - #class C queries in the same session
    - #class C queries in the next/previous query
  - $0.9105 > 0.9145$
- Same session’s queries features
  - Only use similar queries!
  - Use Jaccard to measure similarity between queries
  \[ J(A, B) = \frac{|A \cap B|}{|A \cup B|} \]
  - Bag of words feature for same session’s queries that are similar to the query Q
  - $0.9145 > 0.9182$, utilizing the large scale unlabeled data!
  - Performance decrease for adding same session’s titles

---

**Learning Models**

**Learning setting**
- Given a query Q
- Treat each class label respectively
- Train a classification model to predict the probability that Q belongs to a specific class
- Take the class labels with probability > 0.5 as the classes of the query Q
- If there are more than 2 labels, keep the top two with largest probability

**Models**
- Logistic regression
  \[ P(y) = \frac{1}{1 + e^{-y(x)}} \]
  - Use the implementation of Liblinear
- Factorization machine
  \[ \hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} (v_{i,j} v_{j,i}) x_i x_j \]
  - Use the implementation of LibFM
- Gradient Boosted Decision Trees
  \[ F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \]
  - Use the implementation of XGBoost

**Ensemble**

**Step 1. Feature Extraction**
  a. Bag of words features
  b. Search behavior features

**Step 2. Individual Model Learning**
  a. Logistic regression
  b. Factorization Machine
  c. Gradient boosted decision trees

**Step 3. Ensemble Results**
  a. Obtain the prediction results of individual models
  b. Use logistic regression to ensemble to results

---

**Experimental Results**

Performance on different features and different models.  
GBDT is the best individual model.  
Ensemble can effectively improve the performance