Using Hidden Activity and Sentiment Expansion to Improve Natural Language Question Location Search

Hsien-Hui Lee, Wen-Sen Cheng and Wen-Hsiang Lu

Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan, R.O.C
P78071010@gs.ncku.edu.tw, vinson7973@gmail.com, whlu@mail.ncku.edu.tw

Abstract. Many users use search engines to find location information while they plan to do some activities. However, they have to spend a massive effort to choose the proper query terms, instead of natural language questions, to retrieve useful information. After this enormous work, some candidate locations have selected. Users will also browse community question answering websites to get others opinions or advice about these candidate locations. Then, they choose final locations suitable for their activities and sentiment.

Our research is to provide one system that allows the user to submit natural language question and return a suitable location list that fit users question intent. At first, we analyze the natural language question structure and identify four question components: Question Entity, Question Context, Question Activity, and Question Sentiment. Secondly, some questions are not well-formated with these four parts; we will expanse suitable activity or sentiment terms related to question entity for this question searches. Finally, we use our proposed Activity-Sentiment-based Entity Ranking Model (ASERM) to calculate entity score and ranking this candidate entity list that closes to users natural language questions.

Experiment result shows that our proposed method ASERM can help the user to get entity list which matched their intent. And it shows ASERM really can enhance performance in entity search.

Keywords: Entity Search, Question Activity, Question Sentiment, Question Analysis, Answer Extraction.

1 Introduction

Many users used to acquiring information from the Internet. When they input some query terms and search engines will response search result web pages that contain these terms. Users can get more precision result as long as they choose the right words. However, users have to spend a massive effort to select the proper query terms to express users’ intents. If they input one whole natural language question in search engine web page, this question will be decomposed into several query terms by the search engine. Then, the result web page, returned by the search engine, shows the web pages which contain at least one query term. Users have to visit these web pages one by one and reorganize those web pages into useful information.

With the rise of social networks, more and more users share their comments, such as consumption experiences, opinions, and their sentiments on the Internet. For other users who want to plan a trip or activity, these comments will influence their decisions. However, these rich and diverse comments or sentiments spread around on many websites. Users still need to spend a lot of time to browse and organize them for useful information.

In this paper, we analyze users need via natural language question and recommend a ranked suitable entity website list. The first step of our work is Question Analysis. We divide one list-informational question into a combination of Question Entity, Question Context, Question Activity, and Question Sentiment. Question Entity describes the location entity type that the user wants to search. Question Context describes the factual information which user limit to this focus. Question Activity and Question sentiment are both users' need in this question. Next part is knowledge base training. We trained the relationship score between candidate location entity and these four question component in advance. The last step is to utilize our proposed Sentiment-Based Entity Recommend Model to rank suitable entity list and display it.

2 Related Work

2.1 List Inofrmational Natural Language Search

Natural language search is using human question, not current short query, to search for answers. Broder et al.[1] classified users query intent into three types, i.e., informational query, navigational query, and transactional query. Besides, Rose et al.[2] classified these three types of query more clearly. For example, Information queries can be divided into five sub-types, Directed, Undirected, Advice, List and Locate. Our research focuses on dealing with the problem of List-Informational search. List-informational query is user wants to obtain a list of similar entities. There have many types of research in entities expand. Wang et al.[3] use of set expansion to improve question answering when the expected answer is a list of entities belonging to a specific class.

2.2 Question Structure

In the past, conventional search engines treated the question as a long text and only consider the segmented words without considering question structure and semantic relationship. Recently, many types of research start to consider question structure, as question focus and question context. Ferret et al.[4] deduced question focus will be mapping to a Noun or Noun Phrase and can seek a list of entities. Duan et al.[5] define Question Topic that is the primary context of the question. Cao et al.[6] proposed MDL-based tree cut model to retrieve and rank other questions according to their likelihood of being good recommendations of the queried question.

However, other contents in natural language question are also valuable. In our work, we particular consider Question context, Question Activity, and Question Sentiment.

Lin et al.[7] try to designed question focus identification algorithm. They proposed a novel semantically related feature model (SRFM), which takes advantage of question focuses and their semantically related features learned from the massive number of collected training data to support the determination of question type.

2.3 Opinion Extraction

There are many researchers interested in user opinion extraction. Hu et al.[8] mainly extracted adjectives as opinion words. Popescu et al.[9] extract possible opinion words by using some rules and POS tag. Qiu et al.[10] proposed a novel propagation approach that exploits the relations between sentiment words and topics or product features that the sentiment words modify, and also sentiment words and product features themselves to extract new sentiment words. Although it can increase precision by the rule-based method, it isn't suitable for other languages like Chinese. Ku et al.[11] adopted a character-based method to calculate character Opinion Score in the corpus and it's suitable for Chinese.

3 Methods

3.1 Observation

We choose one question 'Which scenic spots are good for romantic dates in Tainan?' from Yahoo Answers. One user answered that 'Lover's Wharf', 'Golden Coast' are suitable places. To verify these recommendations, we observed some blog articles about 'Lover's Wharf' and found that some blog articles said it is suitable for dating, and there are also some blog articles described it as a beautiful and romantic place. We observe that there are some relationships between 'dating', 'beautiful' and 'romantic'. When users want to find a place, which is suitable for dating, they also find a place which is beautiful and romantic. Those hidden sentiments will affect users' choice.

From this observation, we can disintegrate this question into four parts. 'scenic spot' is Question Entity that user wants to get from the Internet. 'Tainan' is Question Context as query constraint. 'date' is Question Activity that user plans to do. 'romantic' is Question Sentiment that the user wants to create in this activity. Another observation is Question Activity and Question Sentiment often appears together in the articles. Therefore, we can use this characteristic to expand Question Activity and Question Sentiment if the question is not well-formated with these four parts.

3.2 System Architecture

Figure 1 illustrates the architecture of our system, and the system integrates the following four major processing parts: Question Analysis, Activity-Sentiment-Entity Base Training, Hidden Sentiment and Activity Expansion, Activity-Sentiment-based Entity Ranking Model. We will introduce these parts in the following sections.

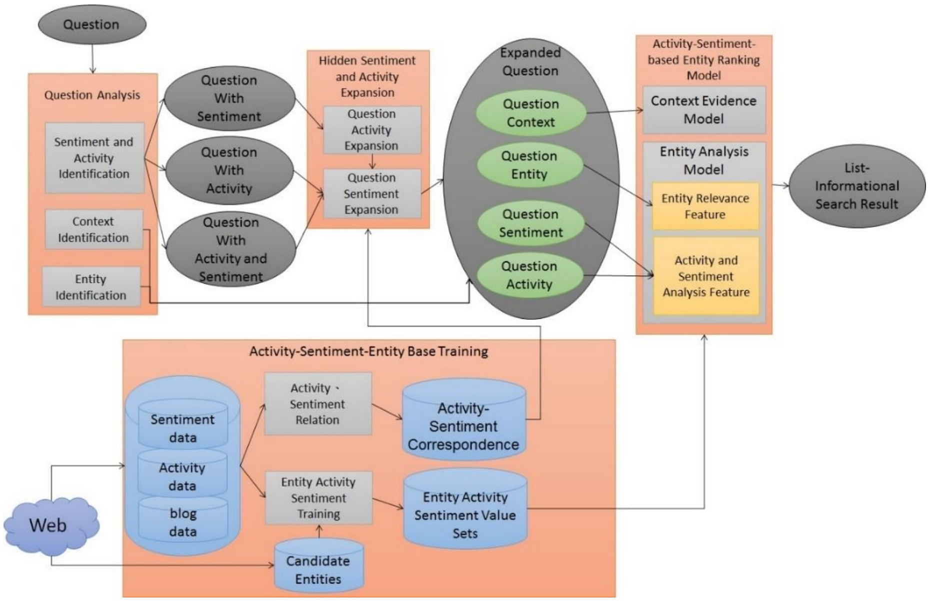

Fig. 1. System Framework

3.3 Question Analysis

In the Question Analysis part, the tasks are analyzed the question type and identify the question component.

Question Type Analysis

In our research, we focus on location entity and all kind of activities, like 'dining', 'travel', 'sport', 'entertainment' or 'nature scenes', etc. We randomly select 60 list-informational questions to analyze their question type. Table 1 shows the statistics of question classification. Most of the question type is 'Question with Activity', only a few questions didn't contain any activity or sentiment. Question without Activity or Sentiment is to search a single entity without mention about any user need or opinion, and we omit this question type in this work.

Table 1. The statistics of question type.

Question Type	Percentage	Example
Question with Activity	63.3%	Which places in Taipei are suitable for shopping?
Question with Sentiment	11.4%	Where are the hottest spots in Kaohsiung?
Question with Activity and Sentiment	15%	Which romantic parks that are suitable for evening appointments in New Taipei City?
Question without Activity or Sentiment	10%	Which cinemas are in Tainan?

Question Components Identification

We randomly select 50 Chinese list-informational questions of these three question types and then use Chinese Knowledge and Information Processing (CKIP)[12] to recognize the POS tags[1]. The next step is to choose proper question components and count these POS tag type frequency manually. According to the statical result, we generate Question Component Identification Algorithm to identify Question components.

Table 2. Question Component Identification Algorithm

Question Component Identification Algorithm
Input: One List-informational Question Q **Output:** Question Entity, Question Activity, Question Sentiment and Question Context
1. Use the CKIP tagger to obtain the POS tags of all words in Q 2. Search nonspecific location terms and mark 'Na' in it as the Question Entity. 3. Search a word with the tag 'Na' or 'Nc' following the question word and mark it as the Question Entity. 4. Search a word with the tag 'Na' or 'Nc' following the word which with the tag 'DE', and mark it as the Question Entity. 5. Remove Question Entity, stop word in Q. 6. Identify whether the question Q contains recommendation word. If there is, use step 6.1; otherwise, use step 6.2. 6.1 Search unigrams with tags 'VA',' VCL','VB','A', and bigram with tags 'Vc+Na' following the recommendation word, and mark it as the Question Activity. Otherwise, search unigrams with tags 'VA',' VCL','VB','A', and bi-gram with tags 'Vc+Na' preceding the recommendation word and mark it as the Question Activity. 6.2 Search unigrams with tags 'VA',' VCL','VB','A', and bigram with tags 'Vc+Na', as the Question Activity. 6.3 If we can't find any Question Activity in step 6.1 or 6.2, search a word with the tag 'Na' from the back to front and mark it as the Question Activity. 7. If step 6 can't found any Question Activity, the Question Activity is set to 'null'. 8. Search a word with the tag 'VH' except recommendation word and mark it as the Question Sentiment. 9. If step 8 can't find Question Sentiment, then the Question Sentiment is set to 'null'. 10. Remove Question Activity, Question Sentiment, question word, recommendtion word and quantifier which qualifies question word in Q. 11. Search unigrams with tags 'Nc',' Ncd', ' Nd', and then search bigram with tags 'Ncd+Nc', as the Question Context.

[1] POS Tag used in this paper:

A: Non-predicative Adjective	Na: Common Noun	Nc: Place Noun
Ncd: Localizer	Nd: Time Noun	DE Preposition
VA: Active Intransitive Verb	VB: Active Pseudo-transitive Verb	
VC: Active Transitive Verb	VCL: Active Verb with a Locative Object	
VH: Stative Intransitive Verb		

3.4 Activity-Sentiment-Entity base Training

Activity-Sentiment-Entity base Training can decompose into three parts: Database Training, Activity-Sentiment Relation Training, and Entity-Activity-Sentiment Training.

Database Training

Our database contains three kinds of data, which are Sentiment data, Activity data and Blog data. We utilized one web crawler to collect questions which user asked in Yahoo Answers. And use the Sentiment and Activity Identification method to extract activity term and sentiment term. Finally, 70 activity terms and 34 sentiment terms collected. We also expanded sentiment data set from the National Taiwan University Sentiment Dictionary (NTUSD)[13] and finally expanded 2,214 sentiment terms. In the aspect of blog data, we use the spider to crawl PIXNET website and collect blog user name, and we gathered 5,492 bloggers and 2,464,142 blog articles at last.

Activity-Sentiment Relation Training

We proposed an Activity-Sentiment Relation formula to calculate relation score between activity and sentiment, as formula (1).

$$Activity\_Sentiment\_rel(A,S) = \frac{Dice(A,S)+P(S|A)}{2} \times IDF(S,D) \quad (1)$$

In this formula, $Dice(A,S) = \frac{2|A \cap S|}{|A|+|S|}$, $|A|$ is the number of articles which contains activity A, $|S|$ is the number of articles which contains sentiment S. $P(S|A)$ is a Conditional probability formula to calculate the probability of sentiment S while an activity A appears in same blog article. $IDF(S,D)$ is Inverse Document Frequency which diminishes the weight of terms that frequently occur in the document set D.

Entity-Activity-Sentiment Training

We use a spider to crawl location entity information and opinions from three travel-related websites: Ipeen, Okgo, and Travelking as candidate entity. The number of blog articles that contain these entities is used as Entity frequency score. If our collected activity terms or sentiment terms appear in these articles, count the article number as Activity-Entity score and Sentiment-Entity score.

3.5 Hidden Activity and Sentiment Expansion

If the question belongs to Question with Sentiment which means this question doesn't contain Question Activity. We utilize Activity-Sentiment relation score to calculate which activity has the highest score with Question Sentiment and take this activity as Question Activity. Use the same method to deal with Question with Activity and set expanded sentiment as top the one sentiment. If the question contains both activity and sentiment, keep the original question sentiment as top one sentiment.

3.6 Activity-Sentiment-based Ranking Model

Model Inference

Given a list-informational natural language question Q, we would like to return a list of entity websites E that satisfy this given question. This answer list can be denoted as A_E. We try to propose a question analysis method to divide a list-informational natural language question Q into a quadruple question structure Q = {Q_C, Q_E, Q_S, Q_A}, where Q_C is question context, Q_E represents question entity, Q_S and Q_A stand for question sentiment and question activity respectively. As to the answer part, each answer A_e of an entity e in answer list A_E can be expressed as a quadruple answer structure A_E={E_e, T_e, S_e, A_e}, where E_e is entity context evidence page, T_e represents entity type of E_e, S_e and A_e stand for entity sentiment summation score and entity activity summation score respectively. We will use the maximum conditional probability formulation to calculate its relevant score and get the answer list.

$$A_e = arg \max_{A_e \in \mathrm{A}_E} P(\mathrm{E_e}, \mathrm{T_e}, \mathrm{S_e}, \mathrm{A_e} | \mathrm{Q_C}, \mathrm{Q_E}, \mathrm{Q_S}, \mathrm{Q_A}) \qquad (2)$$

Formula (2) is our proposed Activity-Sentiment-based Entity Ranking Mode (abbreviated as ASERM). According to our observation, E_e is independent of T_e, S_e, A_e. Moreover, Context evidence page E_e is corresponding to Q_c ,and T_e, A_e, S_e are corresponding to Q_E, Q_A, Q_S. Therefore, we rewrite formula (2) as formula (3):

$$P(\mathrm{E_e}, \mathrm{T_e}, \mathrm{S_e}, \mathrm{A_e} | \mathrm{Q_C}, \mathrm{Q_E}, \mathrm{Q_S}, \mathrm{Q_A}) \approx P(\mathrm{E_e} | \mathrm{Q_C}) P(\mathrm{T_e}, \mathrm{S_e}, \mathrm{A_e} | \mathrm{Q_E}, \mathrm{Q_S}, \mathrm{Q_A}) \qquad (3)$$

According to formula (3), we can decompose ASERM to two models, namely Context Evidence Model and Entity Analysis Model. Context Evidence Model (CEM) uses question context Q_c to generate context evidence page E_e. As formula (4)

$$\mathrm{E}_e = arg \max_{E_e \in E_E} P(\mathrm{E}_e | Q_C) \qquad (4)$$

Entity Analysis Model (EAM) use question entity Q_E, question activity Q_A and question sentiment Q_s to generate entity ranking R_e score. As formula (5)

$$\mathrm{R}_e = arg \max_{\mathrm{R}_e \in \mathrm{R}_E} P(\mathrm{T_e}, \mathrm{S_e}, \mathrm{A_e} | \mathrm{Q_E}, \mathrm{Q_S}, \mathrm{Q_A}) \qquad (5)$$

Context Evidence Model

For the given question context Q_C, we try to find out proper context evidence page E_e. We use context evidence feature function to reach our goal.

$$f_{context\_evidence}(E_e, Q_C) = \begin{cases} 1 & if\ context\ evidence\ page\ E_e\ contains\ the\ question\ context\ Q_C \\ 0 & otherwise \end{cases} \qquad (6)$$

Entity Analysis Model

We select several feature functions to describe Entity Analysis Model (EAM) and use log-linear model to calculate the score of each candidate entity.

Entity Relevance Feature

Entity Relevance Feature is used to determine entity type is accord with question entity.

$$f_{entity\_rel}(T_e, Q_E) = \begin{cases} 1 & if\ at\ least\ one\ of\ \ T_e = Q_E \\ 0 & otherwise \end{cases} \tag{7}$$

Activity Adjusted-Relevance Feature

We multiply Activity Relevance Feature and Activity Frequency Feature in this feature function. We believe that the conditional probability and frequency are both important factors to entity ranking. So, we utilize Activity Relevance Feature and Activity Frequency Feature to consider them respectively. And use this feature to consider those two factors are multiplied.

$$f_{activity\_adj\_rel}(A_e, Q_A) = \frac{\text{Activity\_Entity\_score}(Q_A,\text{e})}{\text{Entity\_frequency\_score(e)}} \times \frac{\text{Activity\_Entity\_score}(Q_A,\text{e})}{\max_{e \in E} \text{Activity\_Entity\_score}(Q_A,\text{e})} \tag{8}$$

Sentiment Summation Adjusted-Relevance Feature

We also multiply Sentiment Summation Relevance Feature and Sentiment Summary Frequency Feature in this feature function.

$$f_{sentiment\_sum\_adj\_rel}(S_e, Q_A, Q_S) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \frac{\boldsymbol{\alpha_i}}{\sum \boldsymbol{\alpha_i}} \frac{\text{Sentiment\_Entity\_score}(Q_S,\text{e})}{\text{Entity\_frequency\_score(e)}} \times \frac{\text{Sentiment\_Entity\_score}(Q_S,\text{e})}{\max_{e \in E} \text{Sentiment\_Entity\_score}(Q_S,\text{e})} \tag{9}$$

$$where\ \boldsymbol{\alpha_i} = Activity\_Sentiment\_rel(Q_{S_i}, Q_A) \times \frac{1}{2^{(Rank(Q_{S_i})-1)}}$$

Activity and top one Sentiment Relevance Feature

This feature calculates the conditional probability of question activity and the top one sentiment of the entity. We use this algorithm to calculate the number of blog articles which contain entity name, Question Activity and top one sentiment.

$$f_{\text{activity\_top\_sentiment\_rel}}(A_e, S_e, Q_A, Q_S) = \frac{\text{Activity\_top\_sentiment\_score}(Q_A,Q_{St},\text{e})}{\text{Entity\_frequency\_score(e)}} \tag{10}$$

$$where\ \boldsymbol{Q_{St}} = \ arg \max_{Q_s \in Q_S} Activity\_Sentiment\_rel(Q_A, Q_S)$$

4 Experiment

4.1 Experiment Setup

Data Set

In our research, we focus on "location" entity and all kind of activities. Experimental data set contains two parts, questions and candidate entities. We collect 250 List-informational questions from Yahoo Answers which are inquiring about location entity. After inspecting these questions, we remove 24 questions which didn't contain any activity or sentiment. In the following experiment, we randomly select 120 questions to be test data. In our work, we want to recommend a list of location entity for users. To collect a large number of entity names, we utilize web spider to collect candidate entity from three websites; they are Ipeen, Okgo, and Travelking. We crawl all kind of location entity include the restaurant, park, playground, natural attractions, etc. We collected 95,328 entities from Ipeen, 2,847 entities from Okgo and 2,154 entities form Travelking. And then we gathered them together to remove duplicate entities. Finally, we collected 96,693 different entities.

Evaluation Matrix

To evaluate the whole system performance, we adopt three evaluate methods in our work. Precision is used to evaluate the performance of Question Analysis. For assessing the performance of Activity-Sentiment-based Entity Ranking Model (ASERM), we adopt the Inclusion Rate. And we take Top-N Inclusion Rate to evaluate whether the Top-N results from our method contain the correct answer or not. Furthermore, we also adopt Normalize Discount Cumulative Gain (NDCG) to assess the performance of ASERM.

Four judges are employed in this stage to determine correctness score for extracting the list of answers, where correct or incorrect. If two judges mark as correct, this result score is 1. If three judges mark as correct, this result score will be 2, and so on. Finally, if the results score high than 1, it will be considered as the correct entity.

4.2 Evaluation of Question Analysis

The overall results of the question analysis are shown in Table 3. Obviously, identifications of question type and question component identification achieved above 80%.

Table 3. The precision of question analysis

Identification of Question Component	**Precision**
Question Context	95.83%
Question Entity	94.17%
Question Activity	85.00%
Question Sentiment	93.33%
Question type	81.67%

4.3 Evaluation of Hidden Sentiment and Activity Expansion

We employed three judges to determine whether the sentiment is related to the activity. If there are two or more judges consider this sentiment is associated with the activity, this result could be recognized. We have 70 activity terms and 2,214 sentiment terms, it needs much time for label all 154,980 (70*2214) relations. So, we labeled the top 50 sentiments from Activity-Sentiment relation formula. Figure 2 shows the precision of label result, we can see that the precision dropped when over top-20 sentiment. Figure

3 shows the inclusion rate of label result, we found that the top-3 inclusion rate achieves above 70% and the top-20 inclusion rate achieves 100%.

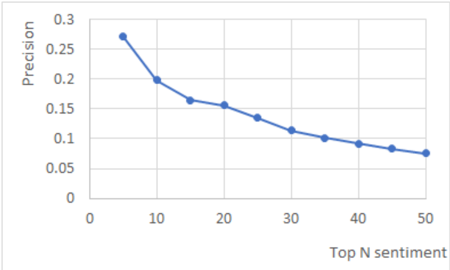

Fig. 2. Precision of Sentiment label result

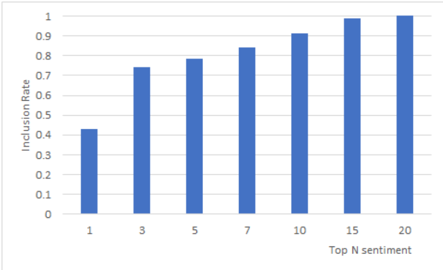

Fig. 3. Inclusion Rate of Sentiment label result

4.4 Evaluation of Activity-Sentiment-based Ranking Model

Parameter Estimation

To train the weights for each feature function, we select some training questions and label all candidate entities to calculate the weight. Because we are more familiar with entities in Tainan, we choose the questions about Tainan to be our training questions. In our List-informational question set, 33 questions about Tainan, we select 20 questions from them and label all candidate entities. Because we didn't limit the entity type, every entity in Tainan can be the candidate entities. Finally, we labeled 25,498 entities and used them for training the weights for each feature. The trained weight shows in table 4.

Table 4. The trained weight of each feature function

Feature Name	weight
Entity Relevance Feature	0.266
Activity Adjusted-Relevance Feature	0.348
Sentiment Summation Adjusted-Relevance Feature	0.296
Activity and top one Sentiment Relevance Feature	0.090

We can see that Entity Relevance Feature and Relevance Frequency Multiply Feature are more significant than Activity and top one Sentiment Relevance Feature. But because the Entity Relevance Feature only contains two result score 0 and 1, the influence of Relevance Frequency Multiply Feature is more important, and the activity feature is slightly important than sentiment feature. Furthermore, the weight of Activity and top one Sentiment Relevance Feature is only 0.090. We speculate that because this feature use entity name, question activity, and the top one sentiment at the same time and the number of blog article which contain those terms are small, thus the result includes a lot of noise.

Baseline

To test our proposed ASERM, we propose three baselines and compare with four situations.

ASERM_label:

Our model with employing manually labeled hidden sentiment.

ASERM_auto:

Our model with employing automatic labeled top-4 hidden sentiment.

ASERM_non_sentiment (abbreviated as ASERM_f6):

We want to compare the influence of activity-based feature; in this baseline, we only use Entity Relevance Feature and Activity Adjusted-Relevance Feature in Entity Ranking Model.

ASERM_non_activity ((abbreviated as ASERM_f7):

Because of the same reason, we only use Entity Relevance Feature and Sentiment Summation Adjusted-Relevance Feature in this baseline.

Experiment Result

We show the result with different evaluation metrics. Figure 4 shows the precision result, the left figure displays the precision which labels score more than 1, which means at least two judges label this result as the correct result. Figure on the right side indicates the precision which at least three judges label this result as the correct result. We can see the precision between 0.4 and 0.5 in the left figure. In the situation which labels score more than 0.2, the precision is between 0.3 and 0.4.

And then we utilize the inclusion rate to evaluate our performance in figure 5. We also consider two situations which are label score more than 1 and more than 2. In the left figure, we can observe that top-10 inclusion rate of four models al-most achieve 100% and the top-3 inclusion rate of ASERM, no matter utilize auto or label data, achieve above 80%. In the right figure, the trend of the four models are similar to the left figure, but the performance of ASERM_label is slightly better than others when over top-3.

Finally, we utilize NDCG to evaluate our performance in figure 6. Because our label score could be 0, 1, 2 or 3. In IDCG, we set the score of all results to 3. We can find that the NDCG value of ASERM_label and ASERM_auto are between 0.55 and 0.6, and the NDCG value of ASERM_f6 and ASERM_f7 are between 0.5 and 0.55. Except for the top-1 NDCG value, the performance of ASERM_label is better than other baselines.

Fig. 4. Top-n answer precision result

Fig. 5. Top-n answer inclusion rate

Fig. 6. Top-n answer NDCG

5 Conclusion

Our research provides a different aspect extracting a suitable answer list for user's needs. And on the other hand, we also efficiently predicted utilize some hidden sentiment terms which the user may not decide. Before extracting answer entity, it's necessary to realize question structure by using question analysis. We classify question into three types and decompose one question into four components: Question Context, Question Entity, Question Activity, and Question Sentiment. Our method achieves above 80% precision.

We utilize user blog article to find the relation between Activity and Sentiment; and proposed an automatic method to find the hidden sentiment which behind the activity. As to the answer, we proposed Activity-Sentiment-based Entity Ranking Model (ASERM) in this paper. This model can be divided into two sub-models to find suitable entity: Context Evidence Model (CEM) and Entity Ranking Model (ERM). In the Entity Ranking Model, we utilize the log-linear model to calculate entity ranking score and propose four different feature functions which use entity type, activity, and sentiment.

In our experiment, we compare our model with ASERM_non_sentiment and ASERM_no_activity to examine whether the activity feature and sentiment feature help. Furthermore, we analyze the sentiment found by our method or manual label. Experiment result shows that our proposed method and model can help a user to find a suitable entity which they want/plan to do an activity in there.

References

1. Andrei Broder.: A taxonomy of web search. SIGIR Forum 36, 2 (September 2002), 3-10 (2002)
2. Daniel E. Rose and Danny Levinson.: Understanding user goals in web search. In: 13th international conference on World Wide Web on Proceedings, pp.13-19. ACM, New York, NY, USA. (2004)
3. Richard C. Wang, Nico Schlaefer, William W. Cohen, and Eric Nyberg. Automatic set expansion for list question answering. In: Conference on Empirical Methods in Natural Language Processing on Proceedings, pp 947-954. Association for Computational Linguistics, Stroudsburg, PA, USA. (2008)
4. Olivier Ferret, Brigitte Grau, Hurault-plantet, M & Illouz, Gabriel & Monceaux, Laura & Robba, I & Vilnat, Anne. Finding an Answer Based on the Recognition of the Question Focus. In: TREC 2001 on Proceedings, pp 362-370. (2001)
5. Duan, Huizhong & Cao, Yunbo & Lin, Chin-Yew & Yu, Yong. Searching Questions by Identifying Question Topic and Question Focus. In: ACL-08: HLT on Proceedings, pp. 156-164. Association for Computational Linguistics, Columbus, Ohio, USA (2008)
6. Yunbo Cao, Huizhong Duan, Chin-Yew Lin, Yong Yu, and Hsiao-Wuen Hon. Recommending questions using the mdl-based tree cut model. In: 17th international conference on World Wide Web on Proceedings, pp.81-90. ACM, New York, NY, USA (2008)

7. Lin, Shu-Jung & Lu, Wen-Hsiang. Learning Question Focus and Semantically Related Features from Web Search Results for Chinese Question Classification. In: AIRS2006 on Proceedings, pp 284-296. Springer, Berlin, Heidelberg. (2006)
8. Minqing Hu and Bing Liu. Mining opinion features in customer reviews. In: 19th national conference on Artifical intelligence on Proceedings, pp 755-760, Anthony G. Cohn (Ed.). AAAI Press. (2004)
9. Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In: Human Language Technology and Empirical Methods in Natural Language Processing (HLT '05) on Proceedings, pp 339-346. Association for Computational Linguistics, Stroudsburg, PA, USA (2005)
10. Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Expanding domain sentiment lexicon through double propagation. In: 21st international jont conference on Artifical intelligence (IJCAI'09) on Proceedings, pp 1199-1204 , Hiroaki Kitano (Ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2009)
11. Ku, Lun-Wei & Chen, Hsin-Hsi. (2007). Mining opinions from the Web: Beyond relevance retrieval. Journal of the American Society for Information Science and Technology 58(12), 1838-1850 (2007)
12. CKIP Homepages, http://ckip.iis.sinica.edu.tw/CKIP/engversion/index.htm
13. NTUSD Homepages, http://academiasinicanlplab.github.io/