RUCIR at NTCIR-12 IMINE-2 Task

Ming Yue[1], Zhicheng Dou[1], Sha Hu[2]*, Jinxiu Li[2], Xiaojie Wang[1], and Ji-Rong Wen[2]
Beijing Key Laboratory of Big Data Management and Analysis Methods, China
School of Information, Renmin University of China
[1]{yomin,dou,wangxiaojie}@ruc.edu.cn,
[2]{sallyshahu,jinxiu2216,jirong.wen}@gmail.com

ABSTRACT

In this paper, we present our participation in the Query Understanding subtask and the Vertical Incorporating subtask of the NTCIR-12 IMine-2 task, for both English and Chinese topics. In the Query Understanding subtask, we combine the extracted candidates from search engine suggestions and Wikipeida, and classify their verticals after clustering and ranking them. In the Vertical Incorporating subtask, we provide a general method for adapting traditional diversity algorithms to deal with predefined subtopics with classified verticals in diversification.

Team Name

RUC IR

Subtasks

Query Understanding (Chinese, English)
Vertical Incorporating (Chinese, English)

1. INTRODUCTION

In modern information systems, users type in some keywords and search engines return matched results. However, with an ambiguous or broad query, a retrieval system or search engine may misunderstand users' interests, by simply comparing the query with the corpus and returning a ranked result list. The goal of NTCIR-12 IMine-2 Task is to find potential intents for a query and classify each intent into one of six verticals. These verticals help us detect different user interests more precisely. The classified intents with their verticals can also be used to improve document ranking. The IMine-2 task consists of two subtasks: Query Understanding and Vertical Incorporating.

In the Query Understanding subtask, the system is required not only to return a ranked list of subtopic candidates for a given query, but also to identify the relevant vertical intent for each subtopic. A subtopic of a given query specializes or disambiguates the original query. These subtopics with their verticals present what information the users are interested in.

We first extract candidates from disambiguation pages in Wikipedia [3, 4]. We do not do any other operations on official query suggestions because they are already good results. Due to the fact that candidates are usually short and do not have enough information, we further retrieve top 300 results from the search engine and group them into clusters to find important candidates by using two different clustering algorithms. After that, we rank them by their relevance and diversity. Finally we make a classification to get each subtopic's vertical.

In the Vertical Incorporating subtask, our goal is to diversify search results in the top ranks, just like the Document Ranking subtask in IMine-1[1]. The unique part of VI task is that it classifies subtopics into verticals to solve diversification problem. The algorithms have to consider additional virtual documents involved by the verticals from the subtopics of a query.

We provide a general method to adapt traditional diversification algorithms to deal with the VI subtask. The mainly difference from traditional models is that we (1)consider verticals and virtual documents in diversity, and (2)understand subtopics by fine-grained information. We have tried this method on several state-of-the-art models, and report the results of PM2[6] as the basic method in this subtask.

2. QUERY UNDERSTANDING

We divide this subtask into two smaller tasks. One is subtopic mining, similar to IMINE, the former NTCIR subtask. The other one is a classification task, which can be treated as a classic machine learning problem. In NTCIR-12, we use query suggestions and knowledge bases to mine subtopics and classify the vertical intent of each subtopic.

2.1 Methodology

Step 1. Extracting Subtopic Candidates From Various Resources. Query suggestions from search engines are one of the official data sets. Besides this, we also use the knowledge base of Wikipedia. In Wikipedia, a disambiguation page describes different aspects for a specific term. We check each query in the task. If a query has a disambiguation page, the terms on the page would be considered as candidates.

*Corresponding author

[1]http://www.thuir.org/IMine/

All the external resources are shown as follows:

- Query Suggestion (Bing, Yahoo, Baidu)
- Query Completion (Google)
- Wikipedia, Baidu Baike

Step 2. Candidate Clustering. We first apply k-medoids clustering algorithm to group similar subtopics together. To calculate the similarity between two candidates, we retrieve top 300 results from Bing search engine for each candidate to build tf-idf vectors by selecting the most important 30 words extracted from the results. Thus, we could simply use cosine function to calculate each pair's similarity. The only difference between k-medoids and k-means algorithm is that, in every iteration, we select each cluster center from original data by k-medoids algorithm while in k-means we just set average distance as final results. After repeating several times, we could normally get a good clustering result. However, we must manually determine the value of k, which could be considered as the hyper-parameter. Another limitation of this method is that randomization of initial centers could possibly fall into the local optimization. These drawbacks of k-medoids algorithm remind us of carefully assigning the value of k and running several times to avoid it. In the experiment, we finally set $k = 12$ and run 20 iterations for each procedure.

Due to the disadvantages of k-medoids method, we use another method called QT (Quality Threshold)[1] to group similar candidates. Compared with the k-medoids algorithm, QT finds the largest clusters whose diameters do not exceed a user-defined threshold. The QT algorithm assumes that all data are equally important, and the cluster with maximum points is selected in each iteration. First, we choose a maximum diameter threshold. Second, we choose a new subtopic to build a candidate cluster, including the points closed to the group within the threshold. Third, we save the candidate cluster and then recurse iteratively.

In this paper, we find that the number of clusters generated by QT algorithm is usually smaller than that by k-medoids method. This is reasonable because some queries like "T test" are not ambiguous and thus the number of their subtopics is small with only two or three clusters.

Step 3. Candidate Ranking and Diversification. We use Maximum Marginal Relevance (MMR) framework [2, 5] to evaluate both relevance and diversity of mined candidates. The MMR model selects the best unranked subtopic in the rest candidates and appends it to the rank list. The MMR model measures relevance and diversity independently and provides a linear combination between them. Thus, the subtopic we choose next could be both relevant and novel. Of course we do not choose the object that has already been ranked.

Given the relevance function $Rel(.)$ and diversity function $Div(.,.)$, the MMR framework could be defined as following:

$$d_{i+1} = \arg\max\{\lambda Rel(d) + (1-\lambda)Div(d, D_i)\}$$

where λ in [0, 1] is a combination parameter, and then

$$D_{i+1} = D_i \cup \{d_{i+1}\}$$

where d_i is i-th ranked subtopic and D_i is the ranked collection. The function $Div(d, D_i)$ measures the diversity of subtopic d when D_i is given. In our experiments, this diversity function is defined as the negative cosine similarity.

Another problem comes into existence because subtopic candidates are usually short and contain little information. Therefore we retrieve the top 300 relevant documents from Bing search engine for each subtopic d. Thus we could build tf-idf vectors from these documents to represent the meanings of subtopics and calculate the cosine similarity between them.

For the subtopic relevance function $Rel(.)$, we simply use the former cluster results to get each subtopic's relevance to the original query. By counting the number of points in the cluster that this specific subtopic belongs to and calculating the average length of the candidates in the cluster, we could use the following formula to describe the relevance function:

$$Rel(d) = \beta ClusterCnt + (1-\beta)IAL$$

where β in [0, 1] is a combination parameter, $ClusterCnt$ is the normalized number of cluster objects and IAL denotes the inverted average length.

Step 4. Candidate Classification. Query Understanding subtask also requests us to identify the relevant vertical for each mined subtopic. In this paper, we first take intuitive simple rule-based classification, then we treat this task as a classic machine learning problem to solve.

We find that subtopic candidates such as "T test" and "What is GPU" have the explicit vertical intent Encyclopedia, and usually the search result pages we retrieve from search engines have some common features. Many of the url links of search results contain words like "knows", "how", and "wiki" as well as "defenition" and "dict" in the titles and snippets. This is also true for other verticals except Web vertical. For News vertical, links and titles may contain "new","latest" or "daily". While for Shopping vertical, "sale","deal", and "coupon" may appear in the links and titles. These give us an inspiration that we can set user-defined rules to identify which vertical class a subtopic lies in. The more the rules, the better the classification results. In practice, for each vertical intent we set some rules mentioned above and count the numbers of links or titles that such rules appear in the search results. To compare and select final vertical, we accept a normalized counting number which is simply divided by total number of search results.

As we mention above, this simple rule-based method needs more common futures to predict a better result. To avoid this, we consider it as a classification problem in machine learning field. At first we select some common features that are the same as those of the rule-based method. When the classification is supervised learning, training data are needed. So we collect about 10,000 queries from former NTCIR tasks and extract some from specific web pages as training queries. Next, for some verticals, by analyzing the structure of a search result page, we could determine its class. For example, if a candidate has a strong News vertical, the search result page possibly has structure . Thus we could get a batch of training samples. For other verticals expect Web, we use the above simple rule-based method to generate the rest training data. After that, we select positive and negative training samples and obtain a trained classifier for each vertical except Web by applying SVM algorithm. Now, given a subtopic, we put it into each classification model and identify its class. If there is no clas-

Table 1: Chinese query understanding results.

Run Name	D♯-nDCG@10	V-score	QU-score
rucir-Q-C-1Q	0.5721	0.5792	0.5757
rucir-Q-C-2Q	0.5721	0.5269	0.5495
rucir-Q-C-3Q	0.4584	0.4393	0.4489
rucir-Q-C-4Q	0.5423	0.5200	0.5311
rucir-Q-C-5Q	0.6264	0.7434	0.6849

Table 2: English query understanding results.

Run Name	D♯-nDCG@10	V-score	QU-score
rucir-Q-E-1Q	0.6182	0.5044	0.5613
rucir-Q-E-2Q	0.6181	0.5626	0.5904
rucir-Q-E-3Q	0.3927	0.4405	0.4166
rucir-Q-E-4Q	0.6349	0.4818	0.5583
rucir-Q-E-5Q	0.7099	0.6724	0.6911

sifier that matches the subtopic, we assume its vertical intent is Web.

2.2 Experiments

2.2.1 *submitted runs*

We submit the following five runs for both Chinese and English Query Understanding subtask:

- rucir-Q-C/E-1Q: combine the suggestions and Wikipedia resources, and use k-medoids clustering method with a trained classifier.
- rucir-Q-C/E-2Q: combine the suggestions and Wikipedia resources, and use k-medoids clustering method with simple rule-based classification.
- rucir-Q-C/E-3Q: combine the suggestions and Wikipedia resources, and use QT clustering method with a trained classifier.
- rucir-Q-C/E-4Q: only use the suggestions resource, and use k-medoids clustering method with a trained classifier.
- rucir-Q-C/E-5Q: only use the suggestions resource, and no clustering or classification method is used.

2.2.2 *Experimental Results*

Table 1 and 2 show the evaluation results of our submitted runs. We observe that the trained classifier works in Chinese subtopic classification by comparing rucir-Q-C-1Q and rucir-Q-C-2Q. In addition, we find that it may not perform well when using QT clustering algorithm in both Chinese and English tasks. What is more, comparing rucir-Q-E/C-1Q and rucir-Q-E/C-4Q, we find that additional resources like Wikipedia do improve the results. Last but not least, our baseline method beats all other methods. The most probable reason is that when we mining subtopics from other resources, the dirty candidates are simultaneously involved in, which may decrease the accuracy and final results.

3. VERTICAL INCORPORATING

3.1 Methodology

The Vertical Incorporating subtask aims to diversify the search result documents based on the predefined subtopics with classified vertical information. The outputs of the Query Understanding subtask are used as the input subtopics in this VI task. The unique part of VI task is that it utilizes classified subtopics to solve diversification problem. Each vertical will add a virtual document into the results to join the diversification. Given a query, the system should rank relevant virtual documents higher in the list while maximizing the diversity of the results.

We extend a state-of-the-art diversity algorithm PM2 [6] to deal with the verticals and virtual documents. We measure the relevance between the document and the subtopic by the expanded key words, which is summarized from the subtopic's query suggestions of Bing Related API. We assign special scores for virtual documents in relevance measuring, and treat virtual documents as normal documents in result diversifying. This is a general method to adapt traditional diversity algorithms to the VI subtask. Experimental results show that this general method can achieve reasonable improvement in diversification.

The details of our method are described below.

Step 1. Data preparation. We extract the content from three parts of a document: title, snippet, and body. These content words will be used to measure the relevances of documents and subtopics later.

Step 2. Subtopic expansion. In the VI subtask, the input subtopics of a given query show the user intents with additional vertical information. Different subtopics describe different aspects of a query, while different subtopics may be related to the same verticals. We understand a subtopic by its content words. To get more information, we want to add more key words for the subtopic. For example, when retrieving subtopics (suggestions) for query "PS", there is a subtopic "PlayStation 4". Involving key words such as "game" or "price" will help us to find more relevant webpages of the original subtopic "PlayStation 4".

One direct way of subtopic expansion is to extract more words. To be consisted with the subtopic resources, we extract key words by Bing Search API. Specifically, we issue the subtopic to Bing and retrieve the related queries of the subtopic as the fine-grained information. We summarize the information and take these key words as the expanded subtopic. Continue the above example. When inputting subtopic "PlayStation 4" as a query into Bing, we find three related queries: "PlayStation 4 new video game", "PlayStation 4 best price", and "PlayStation 4 controller". Thus the expanded subtopic is "PlayStation 4 video game best price controller", which is more accurate to recognize the related documents than the original subtopic "PlayStation 4".

Step 3. Relevance measuring. We use the BM25[7] retrieval model to calculate the relevances between the documents and the queries. For a given query, we sort the documents by their BM25 scores and take this rank as the non-diversified baseline in our experiments.

The relevance between document d and subtopic t_i is similarly measured by BM25 scores. Here the expanded subtopics are used instead of the original subtopics. We normalize the score values and denote them as $rel(d, t_i)$.

According to the official settings in VI subtask, a virtual

Table 3: Results of submitted runs for unclear queries in Vertical Incorporating subtask. The best result is in bold. Statistically significant differences among the submitted runs are marked with $*$, $\star$, $\circ$, $\triangle$, $\dagger$.

	Chinese Unclear Queries			English Unclear Queries		
Run Name	D♯-nDCG@10	D-nDCG@10	I-Recall	D♯-nDCG@10	D-nDCG@10	I-Recall
rucir-V-C/E-1M* [SExp+QU]	**0.7395**$^{\star\circ\triangle\dagger}$	**0.5342**$^{\circ\triangle\dagger}$	**0.9449**$^{\star\circ\triangle\dagger}$	**0.8249**$^{\star\circ\triangle\dagger}$	**0.6565**$^{\star\triangle\dagger}$	**0.9933**$^{\circ\triangle}$
rucir-V-C/E-2M* [SExp+Sug]	0.7079†	0.5268$^{\circ\triangle\dagger}$	0.8890	0.7807	0.5912	0.9701
rucir-V-C/E-3M$^{\circ}$ [noSExp+QU]	0.6884	0.4682	0.9086	0.7994	0.6534$^{\star\triangle\dagger}$	0.9454
rucir-V-C/E-4M$^{\triangle}$ [noSExp+Sug]	0.6801	0.4799	0.8802	0.7719	0.5847	0.9591
rucir-V-C/E-5M† [Baseline]	0.6593	0.4444	0.8742	0.7800	0.5723	0.9876

document should be added into the results if a new vertical is covered by a related subtopic for the query. The virtual document is viewed as the "best" document for this vertical. For the subtopics within the vertical, we think the virtual document is highly relevant and set their relevance scores to the maximum value. For the other subtopics, we view the virtual document as irrelevant and set their relevance scores to 0. In diversification, we treat these virtual documents the same with normal result documents.

Step 4. Result diversifying. We implement the state-of-the-art PM2 [6] algorithm. It maximizes the diversity of selected documents by two processes: finding the best subtopic based on current selected documents, and choosing the next best document by the selected subtopic.

Firstly, we follow the Sainte-Lague equation to compute the quotient q_i for each subtopic t_i. It values the subtopic based on its weight w_i and its seat s_i occupied by previous selected documents. The subtopic with the largest quotient value q^* is selected as the best subtopic t^*.

$$q_i = \frac{w_i}{2s_i + 1}$$

Then, we check all the unselected documents D to select the next best document d^*, which should be highly relevant to current best subtopic t^* and relatively related with other subtopics. Parameter λ controls the balance as follow.

$$d^* = \arg\max_{d \in D}[\lambda \cdot q^* \cdot rel(d, t^*) + (1 - \lambda) \cdot \sum_{t_i \neq t^*} q_i \cdot rel(d, t_i)]$$

Once document d^* is added into the selected document set, its highly related subtopics will be occupied more. We update the occupied seat s_i by the normalized relevance between document d^* and subtopic t_i, before the next loop begin.

$$s_i = s_i + \frac{rel(d^*, t_i)}{\sum_{rel(d,t_j)>0} rel(d^*, t_j)}$$

The algorithm repeats the above processes to iteratively select next best documents from D to the diversified output list.

3.2 Experiments

3.2.1 Dataset

We use the official IMine-2 Web Corpus document collection for both English and Chinese subtasks. For each query, the collection retrieves top 500 results from Bing Search API. Some documents cannot be parsed (e.g., .pdf, .ppt, .doc, and pps), or have no content body. We ignore these documents because we cannot understand them. Finally, we get 42,869 documents for 50 English queries and 38,408 documents for 50 Chinese queries. We use a 5-fold cross validation to tune the parameter λ.

We implement the best run of query understanding subtask as one type of the input subtopics. Each subtopic has a predicted relevant vertical. In vertical incorporating, each vertical will involve a virtual document as a "best" result of the vertical. During document ranking, if the result list tries to contain the results of a vertical, the list should include the virtual document of this vertical.

In addition, we choose official query suggestions from Bing Related API as another type of the input Chinese and English subtopics. To get the verticals for query suggestions, we use the k-medoids clustering method introduced in QU subtask to classify each query suggestion to its most possible related vertical. And we set an uniform weight for the subtopics of the query.

3.2.2 Submitted Runs

We submit the following five runs for both Chinese and English Vertical Incorporating subtask:

- rucir-V-C/E-1M: follow the above methodology, and use the submitted run rucir-Q-C/E-1Q of QU subtask as the input subtopics.
- rucir-V-C/E-2M: follow the above methodology, and use the official query suggestions with our predicted verticals as the input subtopics.
- rucir-V-C/E-3M: follow the methodology except step 2 (subtopic expansion), and use the submitted run rucir-Q-C/E-1Q of QU subtask as the input subtopics.
- rucir-V-C/E-4M: follow the methodology except step 2 (subtopic expansion), and use official query suggestions with our predicted verticals as the input subtopics.
- rucir-V-C/E-5M: the non-diversified baseline ranked by the BM25 model [7].

3.2.3 Experimental Results

We show the results of our submitted runs for unclear queries in Vertical Incorporating subtask in Table 3. We use the two-tailed paired t-test for statistically significance testing and report a significant difference if the p-value is lower than 0.05. We mark the details of the runs for easy understanding: `SExp/noSExp` denotes the run follows the subtopic expansion step or not; `QU/Sug` means the run's input subtopics come from the QU subtask or the official query suggestions.

Table 3 shows that the our method with QU subtopics outperforms the best among the submitted runs for both Chinese and English unclear queries, in terms of D♯-nDCG, D-nDCG, and I-Recall. Specifically, rucir-V-C-1M and rucir-V-E-1M significantly outperform their other runs in terms of D♯-nDCG ($p<0.05$ with two-tailed paired t-tests). This means that our proposed method achieves reasonable improvement in VI subtask, by using the subtopics from QU subtask.

To check the influence of input subtopics, we test our model with the official query suggestions, i.e., rucir-V-C/E-2M. In Table 3, rucir-V-C/E-2M outperforms their baseline (rucir-V-C/E-5M), but underperforms rucir-V-C/E-1M. The results indicate that the proposed method works better when using QU subtopics rather than using query suggestions, and the output of QU subtask works well in diversification.

Recall that our method includes subtopic expansion (See step 2 in Section 3.1) to understand each input subtopic by its fine-grained information. We ignore this step in rucir-V-C/E-3M to see whether this subtopic expansion is necessary. The result shows that rucir-V-C/E-1M significantly outperforms rucir-V-C/E-3M in terms of D♯-nDCG and I-Recall. So the subtopic expansion part is very useful in diversification. One of the possible reasons is that, by involving fine-grained information of the input subtopic, our model can recognize more related documents for the subtopic, and value related documents covering more fine-grained information.

4. CONCLUSIONS

In this paper, we described our approaches for Query Understanding and Vertical Incorporating subtasks in NTCIR-12 IMine-2 task. In the Query Understanding subtask, our methods achieved high D♯-nDCG@10 values. However, some enhanced methods performed not as well as our baseline method. The reason is that the official selected intents mainly come from suggestion data sets. In the future we will do more work to handle this problem. In the Vertical Incorporating subtask, our model archived its best performance when using the results of Query Understanding subtask as the input subtopics, for both Chinese and English unclear queries.

ACKNOWLEDGMENT

This work was partially supported by the National Key Basic Research Program (973 Program) of China under grant No. 2014CB340403, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China No. 15XNLF03, and the National Natural Science Foundation of China (Grant No. 61502501).

5. REFERENCES

[1] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Research*, 9(11):1106–1115, November 1999.

[2] J. Carbonell, J. Goldstein. The use of mmr,diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval SIGIR '98*, pages 335-ÍC336, New York, NY, USA, 1998.

[3] H. Yu, F. Ren. TUTA1 at the NTCIR-11 IMine Task. In *Proceedings of NTCIR-11 workshop*, 2011.

[4] D. Xiao, Z. Han, H. Qi, M. Yang, J. Gao, S. Li. HIT2 Joint NLP Lab at the NTCIR-9 Intent Task In *Proceedings of NTCIR-09 workshop*, 2009.

[5] J. Han, Q. Wang, N. Orii, Z. Dou, T. Sakai, R. Song. Microsoft Research Asia at the NTCIR-9 Intent Task In *Proceedings of NTCIR-09 workshop*, 2009.

[6] V. Dang and W. B. Croft. Diversity by proportionality: An election-based approach to search result diversification. In *SIGIR '12*, pages 65–74, Portland, Oregon, USA, 2012.

[7] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Proc. the 3rd TREC 1994*, pages 109–126, 1995.