IMU Experiment in IR4QA at NTCIR-8

Xiangdong Su, Xueliang Yan, Guanglai Gao, Hongxi Wei
School of Computer Science
Inner Mongolia University
Hohhot, China 010021
Email：csggl@imu.edu.cn

ABSTRACT
This paper describes our work in the subtask IR4QA. Our IR system designed for this task consists of two modules: (1) query processing; (2) indexing, retrieval and re-rank. We first study the method of question classification, and the strategies of weighting based on the result of question classification. Baidu and Wanfang resources are exploited to help query expansion. Through studying the specialty of each index formats and each index unit, we create three indexes of different types: KeyFile-Unigram-Index, KeyFile-Word-Index and Indri-Word-Index. Then we use an interpolating method to re-rank the documents returned from the above three indexes. Our system achieved 0.4266 mean AP, 0.4628 mean Q and 0.6761 mean nDCG in the final evaluation, giving a strong proof of the effectiveness of our approach.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Query formulation, Retrieval models, Selection process.

General Terms
Performance, Experimentation.

Keywords
NTCIR, Query Expansion, Model Combination, Weighting, Information Retrieval.

1. INTRODUCTION
Advanced Cross-lingual Information Access(ACLIA) at NTCIR-8 aims to find satisfactory answers to simple questions as well as complex questions. Information Retrieval for Question Answering (IR4QA) as a sub-task of ACLIA, mainly concentrates on identifying the documents in the corpus which may contain answers to questions. We chooses the IR4QA task and submit four runs, including English-Simplified Chinese (EN-CS) and Simplified Chinese-Simplified Chinese (CS-CS).

This paper describes our IR system. In the query processing module, we obtain the Key-String by using questions resources in NTCIR-7, Baidu Zhidao (*http://zhidao.baidu.com*) and Sina iAsk (*http://www.iask.com*), which can be used for question classification. We implement the question classification through the method combining Key-String and a few of rules of conflict resolution. After that, we tune weights of key terms based on the results of question classification. According to our observation, the Open Category ("开放分类") in Baidu Baike (*http://baike.baidu.com*) is very suitable for query expansion to questions of BIOGRAPHY, DEFINITION, and EVENT. We also find that Related Searches returned from Wanfang (*http://www.wanfangdata.com.cn*) provides a query expansion method in semantic-level and brings the benefit of solving the synonymy problem effectively. For same query, the same format indexes using different index unit will return quite different results. It's also true to the different formats indexes using common index unit. By comparing the performances of different indexes, we create three indexes of different types: KeyFile-Unigram-Index[1], KeyFile-Word-Index[2] and Indri-Word-Index[3] for the documents. We use an interpolating algorithm to re-rank the documents returned from the above three indexes to improve the retrieval performance. The official evaluation results show that our system achieves a good performance [1].

The remainder of this paper is organized as follows. Section 2 describes query processing module in our IR system. Section 3 describes the processes of indexing, retrieval and re-rank, and analyzes the reasons why index combination could improve the retrieval performance. Section 4 presents our experiments. Finally, we give the conclusions and future work in section 5.

2. QUERY PROCESSING
Before the process of retrieval, we need to convert the questions to a suitable format. Figure 1 describes the workflow of query processing. In EN-CS Run, we use the Google Translation Services to translate English questions into Chinese ones. Other treatments to English topics are same as the Chinese ones.

2.1 QUESTION CLASSIFICATION
The topics released by NTCIR Working Group in sub-tasks IR4QA include factoid questions and complex questions. The question types reach nine kinds: PERSON, BIOGRAPHY, DEFINITION, DATE, LOCATION, EVENT, RELATIONSHIP, ORGANIZATION and WHY. As we know, different types of questions have specific types of answers generated from specific documents. In order to improve the IR performance, we need to do specific treatment for questions of specific type. Therefore, the

[1] It was created by lemur toolkit, index format is KeyFile. Index unit is unigram.

[2] It was created by lemur toolkit, index format is KeyFile. Index unit is word.

[3] It was created by lemur toolkit, index format is Indri. Index unit is word.

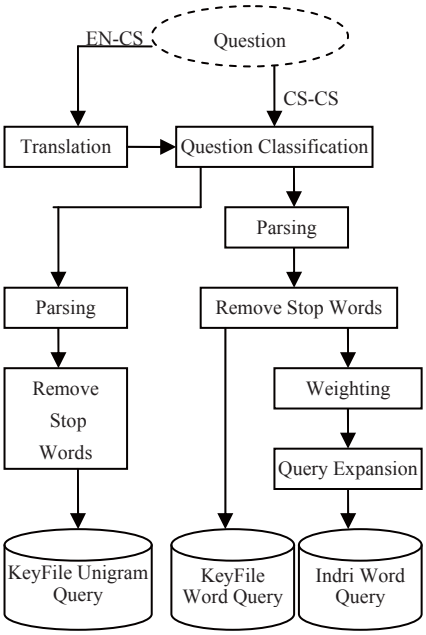

Figure 1. Workflow of query processing

first step of query processing is question classification.

In a common sense, there are specific interrogative or key words that can indicate the question type in Chinese. These words can be used as classification features. Here we use the "Key-String" to represent all of them. In order to find the Key-String of each question type, we collect 800 questions from Sina iAsk (*http://iask.com/*) and Baidu Zhidao (*http://zhidao.baidu.com/*), and classify them manually. Then we use a statistical approach to extract the Key-Strings of each type questions. If Key-String of specific question type appears in one question, we classify the question into corresponding type. Table 1 lists the Key-String of DATE type questions.

Table 1. Key-String of DATE type questions

Key-String: 哪年，哪月，哪天，何时，何年, 何月，何日，什么时候，什么时间，什么时辰，多会儿，具体时间，生日，诞辰，纪念日

To some types of questions, the intersection of their Key-String is not empty. That is, the questions can not be classified by only using Key-String. For these questions, we work out a few rules relying on the POS tagging result to solve the conflict. For Key-String "谁" ("Who"), "谁是" ("Who is"), "是谁" ("Who is"), they belong to Key-String of PERSON and BIOGRAPHY. To solve the conflict, we use the rule R1 which is listed in Table 2.

Table 2.Conflict resolution rule for BIOGRAPHY and PERSON type questions

R1: If except "谁" ("who"), "谁是"("Who is"), "是谁" ("Who is") and punctuation, there is only NR tag in the question after been tagged, the question is BIOGRAPHY type, and otherwise it belongs to PERSON type.

For example, "李宇春是谁？"("Who is LI Yuchun?") is a question in NTCIR-8 Formal Run. Because of Key-String "是谁"("who is") appearing in this question, we conclude that it belongs to BIOGRAPHY or PERSON either. We remove "是谁" and "？", and then tag the remnant. There is only the label "NR". So we assign it to BIOGRAPHY. For question "2005 年超级女声的冠军是谁？"("Who is the champion of Super Girl 2005?"), we assign it to PERSON.

To solve the conflict, we develop five rules in our experiments. The results show that the classification method by combining Key-String and a few rules is effective.

2.2 WEIGHTING

The importance of terms in a query is not the same. Term weight reflects the discriminative power of the term in query. A proper weighting scheme could enhance the retrieval effectiveness. For Lemur does not support weighting operation, there's no weighting in the process of KeyFile-Unigram-Query and KeyFile-Word-Query generation. In experiments, we only use term weighting in Indri-Word-Query.

Term-weighting schemes assign weights to terms relying on how useful they are likely to be in determining the relevance of a document [2]. We use the following scheme for term weighting:

(1) For questions of BIOGRAPHY and DEFINITION, there is no term weighting.

(2) For questions of RELATIONSHIP, we do weighting according to the collection frequency of two objects in the question. When the frequencies of the two objects are not in the same scale, we increase the weight of low frequency object. Otherwise, there is no weighting.

(3) For questions of other types, we increase the weights of the most important two key words.

The reason we use the scheme (1) is: For each question of BIOGRAPHY and DEFINITION, there is only one key word except interrogative and punctuation. So there is no need to do weighting. The reason of scheme (2) is: Since the frequencies of two objects in question are not in the same scale, the retrieval result prefers the high frequency object to the low frequency object. To avoid query drift, we use scheme (2). We tune the term weight according to statistical result on the two objects. For questions of other types, we appropriately scale up the weights of the most important two words in the question. In the absence of guidance on how to precisely tune weights of key terms in a question, we experimentally select the two most important keywords, and adjust their weights two times of other words in the question.

Table 3. Performance comparison on weighting in the training experiments with the NTCIR-7 IR4QA CS test collection

Weighting	**Indri Word Query**		
	mean AP	**Q-measure**	**nDCG**
No	0.5435	0.5396	0.7438
Yes	0.5542	0.5484	0.7494
Legend: Yes represents result using term weighting, No represents result without term weighting			

Table 3 shows the results on the 97 topics in training experiments with the NTCIR-7 IR4QA CS test collection. All of the topics in the experiments are weighted according to the scheme above. We draw a comparison between the result without weighting and the result with weighting. In the experiments, only the operators "combine" and "weight" are used in Indri-Word-Query. From Table 3, we can see the performance improvement after weighting is about 2%. In query expansion, we experimentally assign a comparative small weight to the expansion word.

2.3 QUERY EXPANSION

Vocabulary mismatch between words of relevant documents and user's query decline the accuracy in information retrieval. Query expansion can be used to solve this problem [3]. In order to avoid query drift, the weights of expanding words should be smaller than the weights of the words in original question in general. Because Lemur toolkit doesn't support weighting operation, there is no explicit query expansion to KeyFile-Unigram-Query and KeyFile-Word-Query. We use Pseudo Relevance Feedback to do expansion for them. The index of Indri format supports structured query [4]. In our Experiment, we use query expansion in Indri-Word-Query. The weight of expansion word is tuned according to training experiments with the NTCIR-7 IR4QA CS test collection. We show our expansion methods as follows.

Intuitively, the document about BIOGRAPHY always describes the person's family, educational experience, occupation, deed, etc. The occupation and deed in real world, especially, take the important place, because people always pay more attention to what this person does. So we want to find some related words which could provide such information. Many studies use external resources to help IR and QA [5] [6] [7]. Wikipedia has been used to do query expansion [8]. We found that the Open Category in Baidu Baike (*http://baike.baidu.com*) provides the information we need. Similarly, for the questions of DEFINITION and EVENT, the Open Category in Baidu Baike also provides good expansion words in semantic level. Therefore, we use it to do query expansion for questions of BIOGRAPHY, DEFINITION and EVENT. Table 4 lists the Open Category of entry "何大一" ("David Ho(Da-i Ho)"), "李宇春" ("LI Yuchun"), "老龄化社会" ("Aging Society") and "319 枪击案" ("3-19 shooting incident").

Table 4. Open Category in Baidu Baike

entry	Open Category
何大一	台湾，人物，博士，科学院院士
李宇春	明星，华人明星，歌手，超级女声，影响力人物
老龄化社会	社会保障，社会保险，社会问题，老人，人口研究
319 枪击案	台湾

We draw a comparison between the performances of the result with query expansion which used Open Category and the result without query expansion on 37 CS topics in the training experiments with the NTCIR-7 IR4QA CS test collection. The results are listed in Table 5. In the experiments, index unit is Word and index format is Indri. Only the operators "combine" and "weight" are used in query.

Table 5.Performance comparison between results with and without Open Category expansion in the training experiments with the NTCIR-7 IR4QA CS test collection

Expansion	Indri Word Query		
	MAP	Q-measure	nDCG
No	0.5837	0.5579	0.6820
Yes	0.6060	0.5800	0.7180
Legend: No represents result without expansion, Yes represents result with expansion			

As mentioned above, we expect to find synonyms or related words for terms in question which can be used as expansion words to solve the problem of vocabulary mismatch. We find that Related Searches in Wanfang provide such convenience. When we search specific content in Wanfang database, there are several entries of Related Searches on the returned page. The entries of Related Searches are synonyms or semantic related words. So they are suitable to be used as expansion words. They include three kinds: ty_max, ty_mid, ty_min. In our Experiment, we select the entries of ty_max as the expansion words. And only if the entries of ty_max do not exist, we select the entries of ty_mid as the expansion words. The rest can be done in the same manner. So we use the Wanfang as a synonyms dictionary to expand the key words in question. Let's give an example to illustrate this. There is a question "藏历新年和春节为什么会重合？"("Why does Tibetan New Year and Spring Festival coincide?") in NTCIR-8 Formal Run. The most important two words calculated by Hailiang API(*www.hylanda.com*) are "藏历" and "重合". Table 6 lists the entries of Related Searches about the former two words.

Table 6. Related Search in Wanfang

KeyTerm	藏历	重合
ty_max	历法,时令节气,时轮历,藏历	不动点,匹配,相交,重合
ty_mid		主位,主语,平移,旋转角度,移动变换,超凸度量空间,非紧性测度
ty_min		g-凸空间,双务合同,定性对策,履行,截口,抽象经济

We see that Related Searches provides the expansion words in the lexical representations. In the NTCIR-8 Formal Run, we use the method mentioned above to do query expansion for other questions except BIOGRAPHY, DEFINITON and EVENT.

3. INDEXING, RETRIEVAL AND RE-RANK

In this module, we carry out three tasks: indexing, retrieval and re-rank. Figure 2 shows the concrete workflow. At first, we need to segment the corpus according to the index units used and then create index. Secondly, we input the query of each type into corresponding index and get the returned document list. Finally, we generate the result through re-ranking the lists returned from three different indexes.

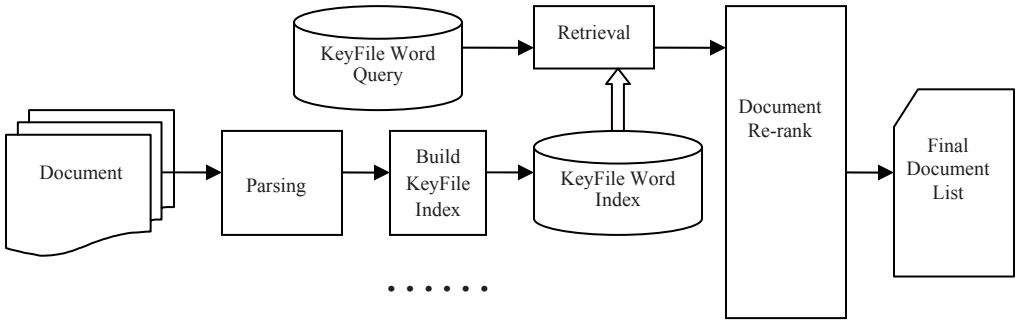

Figure 2. Workflow of indexing, retrieval and re-rank

Table 7. Performance of single index and their combination in the CS-CS training experiments with the NTCIR-7 IR4QA CS test collection

Result / Measure	CS-CS						
	①	②	③	①+②	①+③	②+③	①+②+③
AP	0.4354	0.5424	0.5435	0.5685	0.5600	0.5483	0.5735
Q-measure	0.4445	0.5389	0.5396	0.5693	0.5564	0.5441	0.5738
nDCG	0.6736	0.7378	0.7438	0.7710	0.7579	0.7458	0.7733

3.1 Three Indexes

There is no clear boundary in Chinese. So we need parsing before indexing [9]. In our experiments, there are two types of index units: unigram and word. We use regular expressions to parse the documents into unigram, and use ICTCLAS [4] to parse the documents into words.

In Our IR system, we use Lemur API to create index and retrieval. We totally create three indexes of different types: KeyFile-Unigram-Index, KeyFile-Word-Index and Indri-Word-Index. An interpolating algorithm is executed on the returned document lists from the above three indexes to re-rank the result.

Many index units, including unigram, bi-gram, trigram, etc, can be used in Chinese IR system. Here we give the reasons why we use unigram and word as index units. Except unigram and word, n-gram including bi-gram, trigram would generate meaningless word-pair, leading to a decline in retrieval accuracy. At the same time, these N-Grams do not facilitate using external resources for query expansion. For unigram and word, there are no such problems. Word is the comparable better semantic unit in Chinese. In addition, the use of unigram as index units can improve the recall. For some problems, the number of returned documents from KeyFile-Word-Index and Indri-Word-Index is small. The high recall of KeyFile-Unigram-Index will help IR system to increase the accuracy of these problems. Although its retrieval accuracy is low, the negative impact is relatively small as its relatively small interpolation coefficient. And thus it helps the overall retrieval accuracy. We therefore adopted the unigram and word as index units.

We do experiments with the NTCIR-7 IR4QA CS test collection. The performances of experimental results are listed in Table 7. In the experiments, we use three type indexes: KeyFile-Unigram-Index, KeyFile-Word-Index and Indri-Word-Index. We use TFIDF retrieval model with PRF in KeyFile Index, and use the "combine" operation in Indri Query. In Table 7, ①, ② and ③ represent the results for KeyFile-Unigram-Index, KeyFile-Word-Index and Indri-Word-Inde, and ①+②, ①+③, ②+③ and ①+②+③ represent their combined results respectively. There are no weighting and query expansion for these three kinds of queries. We empirically set the parameters of combination.

Combination of different index units could improve retrieval precision [10]. Our experimental results in Table 7 also confirmed this. We also see that the combination of indexes using different formats can improve the retrieval performance.

In Table 7, the performances of index combination are higher than the performance of each single index. The improvement in performance is caused by the following reasons.

At first we notice that, for the same problem, a considerable portion of the documents in the three returned lists from the above three indexes are different. Let's use an example to illustrate this.

For the problem "全球气候变暖"("List the hazards of global warming"), the number of returned documents from each index is 1000. 503 documents appeared in the three returned lists at the same time. 523 documents appeared in the returned lists from KeyFile-Unigram-Index and KeyFie-Word-Index at the same time. 509 documents appeared in the returned lists from KeyFile-Unigram-Index and Indri-Word-Index. 934 documents appeared in the returned lists from KeyFile-Word-Index and Indri-Word-Index. 577 documents only appeared in one returned list. The total number of returned documents in the three returned lists is 1537. How to select 1000 documents from the 1537 documents in the three returned lists as the final result. Although the performance of result from single Indri-Word-Index is the highest in the results from the above three indexes, using the returned document list from Indri-Word-Index as the final result is not optimal. We naturally think as follows:

[4] http://www.ictclas.org.

(1) If document A appears in three returned lists from above indexes at the same time, document B appears only in one returned list from index K (K is one of the three indexes), and the score of document A in the returned list from index K equal to the score of document B, we believe that document A is more relevant than document B.

(2) Similarly, if document A only appears in the returned list from Indri Word Index, document B only appears in the returned list from KeyFile-Unigram-Index or KeyFile-Word-Index, and the score of document A is equal to the score of document B in the corresponding returned list, we believe that document A is more relevant than document B as the performance of Indri Word Index is higher than KeyFile-Unigram-Index and KeyFile-Word-Index.

(3) Although the performances of KeyFile-Unigram-Index and Key-Word-Index are lower than Indri-Word-Index, we believe document A is more relevant than document B, if document A only appears in the returned list from KeyFile-Unigram-Index or KeyFile-Word-Index and its score is much higher than document B which only appears in the returned list from Indri-Word-Index.

(4)The other situation is similar.

We designed the interpolation algorithm to re-rank the documents based on the assumption above. From Table 7, we see performances of combined results are better than any result without combination. Similarly, this assumption can be adapted to any indexes combination. We use the same method to re-rank the documents in the NTCIR-8 Formal Run. The final evaluation results of our runs show that the method we used is effective.

3.2 Re-rank

To improve the retrieval performance, we use an interpolating algorithm to re-rank the documents which is listed in Table 8.

Table 8. Re-rank algorithm

Description: This algorithm is designed to implement the document re-rank task according to the returned document lists from three indexes mentioned above.
(1)score normalize:
For each returned document list from above three indexes
For each D_i in the returned document list
Score normalize(D_i);
(2)compute score:
For each D_i appeared in one of the returned document list
Score interpolating(D_i);
(3)sort documents:
Sort(score list);

The score of each document in returned document list from index of KeyFile format is positive, and the score from index of Indri format is negative. Before index combination, normalizing the score of each document in returned lists is needed.

For each document, score normalization uses the formula (1):

$$Score(D_i) = \frac{Score(D_i) - Min(Q)}{Max(Q) - Min(Q)} \qquad (1)$$

where

$Max(Q)$: The highest score in returned document list of Query Q

$Min(Q)$: The lowest score in returned document list of Query Q

$Score(D_i)$: The score of document D_i in returned list of Query Q

After normalization, the score of each document in returned lists ranges from 0 to 1.

We use the formula (2) to refine the score of each document.

$$Score(D_i) = \alpha Score_{KU}(D_i) + \beta Score_{KW}(D_i) + \gamma Score_I(D_i) \qquad (2)$$

where

α, β, γ : The interpolating coefficients of the scores of D_i in the corresponding returned document lists.

$Score_{KU}(D_i)$: The score of D_i in the returned list from KeyFile-Unigram-Index. If D_i is not in the returned list from KeyFile-Unigram-Index, it equals to zero.

$Score_{KW}(D_i)$: The score of D_i in the returned list from KeyFile-Word-Index. If D_i is not in the returned list from KeyFile-Word-Index, it equals to zero.

$Score_I(D_i)$: The score of D_i in the returned list from Indri-Word-Index. If D_i is not in the returned list from Indri-Word-Index, it equals to zero.

There are three parameters: α , β , γ in formula (2) which represent the weights of the document scores in each returned document list respectively. Now we have the following expressions.

$$\begin{cases} \alpha + \beta + \gamma = 1 \\ \gamma > \alpha \\ \gamma > \beta \end{cases} \qquad (3)$$

The first expression in (3) is obvious. There are two inequalities in (3). Because the retrieval accuracy of Indri-Word-Index is higher than KeyFile-Unigram-Index and KeyFile-Word-Index, we prefer the result returned from Indri-Word-Index to the results returned from other indexes. The comparison above is made on such basis: at first, there are no query expansion in Indri and KeyFile; second, we use TFIDF retrieval model with PRF in KeyFile retrieval, and "combine" operator in Indri retrieval.

The experiment of ①+②+③ listed in Table 7 uses the parameters as follows:

$$\alpha = 0.25, \quad \beta = 0.25, \quad \gamma = 0.5$$

The parameters here are not optimal. After we use the weighting and query expansion in Indri-Word-Query, the accuracy of Indri-Word-Index will be higher than before. So it should increase the value of γ. In experiment section, we will show the value of α, β and γ in Formal Run.

4. EXPERIMENT

In the IR4QA task at NTCIR-8, our group submitted four runs: IMU-CS-CS-01-T, IMU-CS-CS-02-T, IMU-CS-CS-03-T, and IMU-EN-CS-01-T.

The total number of the topics released by NTCIR Working Group in IR4QA is 100. 73 topics are used in the final evaluation. There are nine kinds of questions: PERSON, BIOGRAPHY, DEFINITION, DATE, LOCATION, EVENT, RELATIONSHIP, ORGANIZATION and WHY. We list our experiments as follows. The corpus we used is in Table 9.

Table 9. Corpus in NTCIR-8 formal run

Lang	Name	Year	#Doc
Chinese (Simplified)	Xinhua	2002-2005	308,845

Table 10 lists the official evaluation results of our submitted runs. We generate the three CS-CS runs based on different interpolating parameter settings. IMU-EN-CS-01-T, which integrates the Google Translation Service into the query translation stage, used the same parameters as IMU-CS-CS-01-T. The interpolating parameter setting of each run is listed in Table 11. According to the training experiment, we empirically select the interpolating parameters, which fulfill the ratio: $\alpha:\beta:\gamma=(1-\gamma)^2:(\gamma(1-\gamma)):\gamma$. More experiments are needed to find the optimal interpolating parameters. In the submitted runs, IMU-CS-CS-01-T achieves the highest score among all runs, and confirms that index combination with a reasonable parameter setting could obtain a better result than any individual index. IMU-CS-CS-02-T is close to the result returned from Indri-Word-Index. The latter reaches the highest score among the single indexes. The performance of IMU-CS-CS-03-T is slightly lower than the result returned from Indri-Word-Index. This is caused by the unreasonable interpolating parameters.

Table 10. Performance of our results in NTCIR-8 formal run

Measure	mean AP	mean Q	mean nDCG
IMU-CS-CS-01-T	0.4266	0.4628	0.6761
IMU-CS-CS-02-T	0.4114	0.4480	0.6580
IMU-CS-CS-03-T	0.4032	0.4394	0.6575
IMU-EN-CS-01-T	0.3184	0.3540	0.5720

Table 11. Parameters setting of results combination in NTCIR-8 formal run

Measure	α	β	γ
IMU-CS-CS-01-T	0.04	0.16	0.80
IMU-CS-CS-02-T	0.16	0.24	0.60
IMU-CS-CS-03-T	0.25	0.25	0.50
IMU-EN-CS-01-T	0.04	0.16	0.80

Table 12 lists the classification result of the 100 released topics. The overall accuracy of question classification is 0.87.

Table 12. Question classification result

Type	BIOGRAPHY	PERSON	EVENT
Number	10	5	18
Type	**ORGANIZATION**	**LOCATION**	**DATE**
Number	4	7	5
Type	**RELATIONSHIP**	**DEFINITION**	**WHY**
Number	19	10	22

In IR4QA task, CS runs were evaluated using 73 topics [1]. In these 73 topics, we classified 63 correctly. Table 13 lists the mean AP of each question type of these 63 topics. The relative order of each question type about the mean AP is in Table 14.

From Table 13 and Table 14, we can clearly see that the mean AP of ORGANIZATION is much higher than that of WHY. The mean AP of BIOGRAPHY, RELATIONSHIP, and EVENT are close. The mean AP of WHY is the lowest in all the question types. According to our analysis, the reasons are listed as follows:

- The mean AP of ORGANIZITION is the highest in all question types. But because there are only two topics belonging to ORGANIZATION, including ACLIA2-CS-0028 and ACLIA2-CS-0061, it is not a sufficient proof which indicates the AP of ORGANIZATION will be always higher than others. The AP of the topic ACLIA2-CS-0061 reaches up to 0.9226. This maybe due to that people are used to introducing a person by adding his or her title and affiliation. The AP of topic ACLIA2-CS-0028 is only 0.3415.
- The mean AP of EVENT, RELATIONSHIP and BIOGRAPHY are very close and comparatively higher than other question types. This partially due to that we do query expansion for questions of EVENT and BIOGRAPHY. And also the term weighting scheme for RELATIONSHIP is efficient. Another reason is that the corpus is newswire, which means that the documents in which the keywords of one question appear almost are relevant to that question.
- The mean AP of DEFINITION is lower than EVENT and BIOGRAPHY, although we do query expansion for all of them. Our post submission experiments show that the Open Category in Baidu Baike gives little help for DEFINITION overall.
- The mean AP of WHY is the lowest in all question types. This maybe due to the fact that the documents which contain the keywords of a WHY question may do not include the real reasons of the question at all.

Table 13. The mean AP of Each Question Types

Type	BIOGRAPHY	PERSON	EVENT
Number	0.535733	0.427233	0.557311
Type	**ORGANIZATION**	**LOCATION**	**DATE**
Number	0.63205	0.42242	0.3473
Type	**RELATIONSHIP**	**DEFINITION**	**WHY**
Number	0.538154	0.305363	0.2096

Table 14. Mean AP comparison of each question types

ORGANIZATION>EVENT>RELATIONSHIP>BIOGRAPHY >PERSON> LOCATION>DATE>DEFINITION>WHY

In NTCIR-8 formal run, we combine the returned document lists from KeyFile-Unigram-Index, KeyFile-Word-Index and Indri-Word-Index to generate the final result. The returned document list from Indri-Word-Index is the result with weighting and query expansion (**Indri+W+E)**. To show the efficiency of the query expansion method and term weighting scheme, we make a

comparison between the performances from Indri with and without weighting and expansion after submission. These experiments are all performed on the NTCIR-8 formal collection and evaluated using the NTCIR-8 official relevant judgment. The concrete results are listed in Table 15, which indicates that the weighting and expansion method enhance the performance greatly.

Table 15. Performance comparison of whether use weighting and query expansion in Indri-Word-Query

Measure	Indri	Indri+W+E
mean AP	0.4017	0.4110
mean Q	0.4378	0.4486
mean nDCG	0.6541	0.6655
Legend: indri represents result without weighting and expansion, Indri+W+E represents result with weighting and expansion.		

As we mentioned above, we combine the returned document lists from KeyFile-Unigram-Index, KeyFile-Word-Index and Indri-Word-Index to generate the final result. We make a comparison among each document list returned from the above three indexes and their combination. The results evaluated with the NTCIR-8 official relevant judgment are listed in Table 16. ①+②+③ represents the result after combination, which have been submitted as IMU-CS-CS-01-T. The interpolating parameters of IMU-CS-CS-01-T are:

$$\alpha = 0.04\ ,\quad \beta = 0.16\ ,\quad \gamma = 0.8\ .$$

According to the experimental results, index combination really improves the retrieval performance.

Table 16. Retrieval performance of three indexes and their combination

Measure	①	②	③	①+②+③
MAP	0.3088	0.4016	0.4110	0.4266
Q-Measure	0.3450	0.4407	0.4486	0.4628
nDCG	0.5803	0.6483	0.6655	0.6761
Legend: ①, ②, ③ represents the document lists returned from KeyFile-Unigram-Index, KeyFile-Word-Index, Indri-Word-Index respectively, ①+②+③ is the result after combination				

5. CONCLUSION AND FUTURE WORK

A simple weighting scheme has been used in our IR system, which brings a slight improvement to performance. We exploit Open Category in Baidu and Related Searches in Wanfang to solve the vocabulary mismatch. Our experiments show that combination of indexes using different formats can improve retrieval performance. They also confirm that combination of indexes using different index units helps the retrieval performance. So we believe index combination could obtain a more satisfactory result in general.

Static parameters have been used in our experiment of index combination. We think the dynamic parameters will bring a better result. Next, some effort will be put in the dynamic parameter optimization.

6. ACKNOWLEDGEMENT

This paper is supported by National Science Foundation of China and the project number is 60865003.

7. REFERENCES

[1] Tetsuya Sakai, Hideki Shima, Noriko Kando, Ruihua Song, Chuan-Jie Lin, Teruko Mitamura, Miho Sugimoto, Cheng-Wei Lee. Overview of NTCIR-8 ACLIA IR4QA. Proceedings of the NTCIR-8, 2010.

[2] Cunnins, R. and O'Riordan, C. A Framework for the study of Evolved Term-Weighting Schemes in Information Retrieval. Artificial Intelligence Review, pp. 35 – 47, 2006.

[3] Li, W.-J, Zhao, T.-J., and Wang, X.-G. A new approach to query expansion in information retrieval. HIGH TECHNOLOGY LETTERS, 2008.

[4] Strohman, T., Metzler, D., Turtle, H., and Croft, B. Indri: A Language-model based search engine for complex queries. Intelligent Analysis, 2005.

[5] Hunt, W. A., Lita, L. V., and Nyberg, E. Gazetteers, WordNet, Encyclopedias, and The Web: Analyzing Question Answering Resources. Technical Report CMU-LTI-04-188,2004.

[6] Yang, H. and Chua, T.-S. QUALIFIER: Question Answering by Lexical Fabric and External Resources. ACL, pp.363-370, 2003.

[7] Katz, B., Marton, G., Borchardt, G., Brownell, A., Felshin, S., Loreto, D., Louis-Rosenberg, J., Lu, B., Mora, F., Stiller, S., Uzuner, O., Wilcox, and Angela, W. External Knowledge Sources for Question Answering. TREC, 2005.

[8] Hsu, C.-C., Te, Y., Chen, Y.-W., and Wu, S.-H. Query Expansion via Link Analysis of Wikipedia for CLIR. NTCIR-7, pp.125-131, 2008.

[9] Liu, I.-C., Ku, L.-W., Chen, K.-H., and Chen, H.-H. NTUBROWS System for NTCIR-7 Information Retrieval for Question Answering. NTCIR-7, pp.153-158, 2008.

[10] Shi, L.-X. and Nie, J.-Y. Using Unigram and Bigram Language Models for Monolingual and Cross-Language IR. NTCIR-6, pp.23-25, 2007.