ICT-Crossn: The System of Cross-lingual Information Retrieval of ICT in NTCIR-7

Weihua Luo[1,2] Tian Xia[1,2] Ji Guo[2,3] Qun Liu[2]
1 Graduate University of Chinese Academy of Sciences, Beijing 100190, China
2 Key Laboratory of Intelligent Information Processing, Institute of Computing Technology (ICT), Chinese Academy of Sciences, Beijing 100190, China
3 School of Software and Microelectronics, Peking University, Beijing 100871, China
{luoweihua, xiatian, guoji, liuqun}@ict.ac.cn

Abstract

IR4QA is a new task in NTCIR-7, which intends to evaluate which IR techniques are more helpful to a QA system. This paper describes in detail the implementation of our IR4QA system, ICT-Crossn. The system consists of a query translation component that integrates the methods of phrase based statistical machine translation and OOV translation methods based on search engine, and a document retrieval component which combines outputs of multiple IR models with a linear model. We tune the parameters on the development set constructed on the dry run set. The official evaluation results show our method achieves a good performance.

Keywords cross-lingual information retrieval, question answering, statistical machine translation, model combination

1 Introduction

In 2008, Multilingual Interactive Technology Laboratory of Institute of Computing Technology (MITEL as the team name) takes part in the NTCIR-7 evaluation for the first time. Compared with the previous sessions, the setting of evaluation tasks in NTCIR-7 has changed greatly. Since the organizers of NTCIR think the research in CLIR (Cross-Lingual Information Retrieval) has matured and wonder how IR techniques help the QA research, CLIR is no longer taken as an individual task to be evaluated. Instead, IR4QA (Information Retrieval for Question Answering) replaces the traditional CLIR evaluation and becomes a subtask of ACLIA (Advanced Cross-Lingual Information Access) task cluster. IR4QA evaluates the ranked document lists generated by IR systems using the traditional metrics such Mean Average Precision (AP), but the retrieval task is treated as a component of the cross-lingual QA system[1]. That is, the test set for evaluation consists of various kinds of questions instead of ad hoc queries, and the evaluation metrics are devised in favor of the effectiveness of IR systems as a module in QA systems.

We implement a CLIR system titled as "ICT-Crossn". Because of time limit, we have not developed a QA system yet and only participate in the EN-CS (English-Simplified Chinese) and CT-CT (Traditional Chinese-Traditional Chinese) subtasks of the IR4QA task.

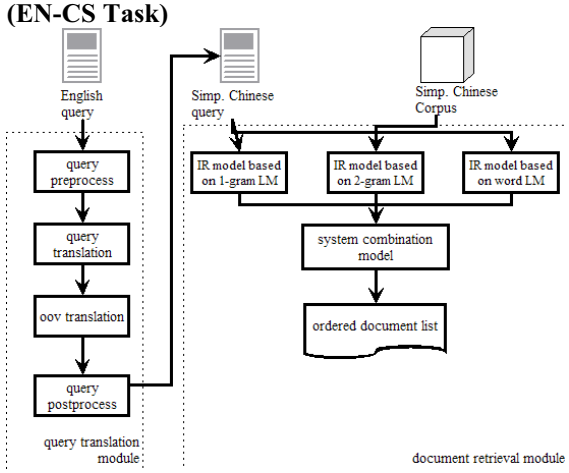
ICT-Crossn consists of a query translation component and a document retrieval component. When performing a CLIR task, e.g. reading the English queries and searching in the Chinese corpus, ICT-Crossn firstly translates the English queries into Chinese, and then searches for the relevant documents in the Chinese corpus. In the translation process, we use the output of phrase-based statistical machine translation (SMT) and accomplish the translation of OOV (out-of-vocabulary) words with search engines[2,3,4]; while in the retrieval process, we combine the results generated by multiple IR models. To tune the parameters of the system, we construct manually the development set based on the dry run set. The official evaluation results show our method achieves good performance.

The remainder of the paper is organized as follows. Section 2 describes the overall architecture of the system. Section 3 and 4 explain respectively our method in query translation and document retrieval. In section 5, the experiments on the development set and the results are introduced. The official results in NTCIR-7 evaluation of our system are provided in section 6. Finally, section 7 presents our conclusion and the future work.

2 The system architecture

In the CLIR tasks of the previous sessions of NTCIR, e.g. NTCIR-5 and NTCIR-6, each query consists of a title which contains the keywords describing the information need briefly, a description which explains the information need in detail, and a narrative which supplies more details about the query, such as the background of the topic and the judgement metric of document relevance. However, for the test set of IR4QA in NTCIR-7, each query is a complex question, such as "What are stem cells", "List major events in Saddam Hussein's life".

Figure 1. the overall architecture of ICT-Crossn (EN-CS Task)



For the IR4QA task, there are two important problems we should pay attention to. The first is that the requirement of translation is more stringent. Generally the questions are very concise, it leads to wrong understanding if 1 or 2 words in a question are translated improperly. The second is that the judgement metric of document relevance in QA differs from the one in IR. We think, for QA, the document containing the answer to the question is more important than the one just related to the queries derived from the question. For example, for the question "who is Junichiro Koizumi", a document containing the personal history of Koizumi is more important than the one about a meeting he attended. Thus, the goal of our system is to implement an effective method to translate the short wh-questions or assertive sentences more accurately, and to modify the current IR models to make them fit for QA metrics.

We think the traditional CLIR framework is still suitable for IR4QA because the input and output are similar with a CLIR one. However, some successful algorithms for CLIR should be re-tested and modified for the new problems in IR4QA. Based on the considerations above, we follow the traditional CLIR methods that split the whole process into 2 steps. One is responsible for translating the queries into the language of the corpus. The second step performs ad hoc retrieval according to the queries. Figure 1 shows the overall architecture of ICT-Crossn. As we see, the architecture is fit for both monolingual and multilingual IR tasks because the retrieval module doesn't care how the queries are made.

However, the specific techniques used in our system are different from many CLIR systems. We don't use dictionary-based translation methods that are widely used in many CLIR systems. Our system borrows some techniques of SMT to improve the translation performance, especially the translation of phrases. While in retrieval, we are inspired by the idea of system combination. Results from different IR systems are usually complementary, and achieve more steady but better performance than a standalone system for various tasks.

Before translation and retrieval, some preprocessing operations are executed, including text segmentation, encoding conversion, etc. We use ICTCLAS[1] to perform word segmentation on the simplified Chinese documents. Unigram and bigram segmentation are also performed on the corpus to build the indices. If the query or the corpus is traditional Chinese, we convert all texts to simplified Chinese at first, and then perform word segmentation with ICTCLAS.

In the query translation module, we run the word alignment toolkit — Giza++[2] on the Chinese-English parallel corpus and then run our phrase extraction tool to extract bilingual phrase pairs. The phrase table generated on a large-scale corpus (e.g. Gigaword released by LDC) covers most common phrases. Take the EN-CS subtask of IR4QA as an example. ICT-Crossn searches for the available Chinese translations corresponding to phrases in an English question in the phrase table, sorts the candidates by probabilities and chooses the top ones. If some have no translations in the table, ICT-Crossn will seek for the help of search engines. It constructs the queries based on the English phrases, submits them to a search engine to obtain the Chinese web pages, and then extracts the possible translations by statistic information.

In the document retrieval module, we use Lemur[3] to build our IR system. Furthermore, we build 3 kinds of IR systems by altering the index unit, i.e. unigram, bigram and word. Each system uses pseudo relevance feedback based language model and provides a different ranked document list in most cases. Then we combine all lists into the final result with a linear function.

The following sections will describe the algorithms and techniques of our system in detail respectively.

3 Module of question translation

3.1 Overview

We don't intend to perform a full text translation on the questions with a SMT system, because the question is usually short and it is difficult for the SMT system to translate it well without context. Additionally, we think IR4QA prefers more relevant key words to the best translation, and is not sensitive to order of the words.

[1] ICTCLAS is an open source tool for simplified Chinese word segmentation, which is developed by ICT and available in http://www.nlp.org.cn/project/project.php?proj_id=6.

[2] Giza++ is a training toolkit of statistical translation models and word alignment. It is available in http://www.fjoch.com/GIZA++.html.

[3] Lemur is a famous open-source toolkit of language modeling and information retrieval. Its official web site is http://www.lemurproject.org/.

Therefore, our goal is to translate the key words in the question instead of the whole sentence. Our work focuses on phrase translation because we think words in sentences are ambiguous but phrases are more clearly defined. And translating multiple words as a phrase introduces much less candidates than translating these words separately. For IR4QA, the critical information in the question is often phrases. If not translated well, we will retrieve the irrelevant documents. Therefore, how to recognize a phrase and how to translate them correctly are the crucial problems we face.

3.2 Phrase translation

In the beginning, we compile LDC dictionaries, electronic version of traditional dictionaries and some terminology dictionaries into a huge dictionary. The phrase translation module enumerates the phrases in a question and searches for their translations in the dictionary. However, quite a lot of words and phrases cannot be identified and translated properly for the dictionary has poor coverage. In addition, dictionaries don't provide possibilities to rank the candidates. We also tried a Stanford parser to recognize the constituents we need in a question. Although it works well for long sentences, it introduces many errors for short ones. As a result, we try another way. At first, we extract the phrases and words by the specific structures in the sentences. Then we translate them with a phrase table.

The phrase table we use is trained on 5 million parallel sentence pairs which is constructed on the English-Chinese corpus released by LDC for NIST MT evaluation. The training process is to use Giza++ to align words on the corpus, and then to use our phrase extraction tool to extract the parallel phrase pairs and to estimate the alignment probabilities[5]. The phrase table occupies up to 17GB in disk. In evaluation, to reduce the size of the table, we filter it with the test set. So only the phrases existing in the test set are kept in the table.

In the test set of the dry run and the formal run, it is easily observed that most questions follow a few fixed structures. For instance, "Who is Osama Bin Laden?", "Who is Ronaldo?", "List the hazards of global warming.", "List the relationship between inflation and the economy.", etc. Roughly, four types of question are defined in four kinds of structure. Therefore, we predefine some structural templates to capture the constituents we need in the questions. For example, a template we define is "what is X" and X is the constituent we want. However, this kind of template has a rough granularity. E.g. for a question like "what is Three Gorges Dam in China", we can extract "Three Gorges Dam in China", but it's a compound phrase consisting of shorter phrases and words and has no translations in the phrase table. Thus, we divide the compound phrase into "basic" ones with the help of the phrase table.

Algorithm 1. The algorithm of phrase identification and translation

Input: an English question in the test set, the structural template set T, the phrase table P, the constituent set D

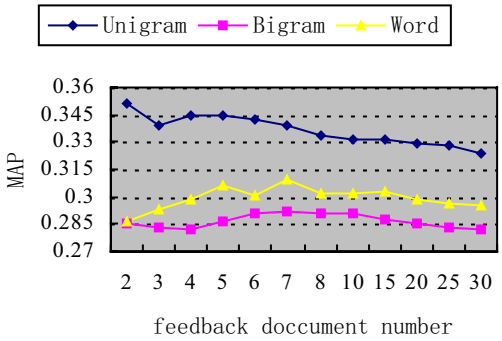
Output: the Chinese key words of the question

1) Initialize: $D = \Phi$
2) For each rule in T:
   Extract the constituents c_i from the question with each template of T;
   Add c_i into D
   End for
3) Remove the repeated items in D
4) For each item c_i in D
   If $| c_i | \leq 3$ then
   Take c_i as a basic phrase and search for the translation in the phrase table
   If not found, pass on c_i to the OOV translation
   Else
   Enumerate the sub-phrases in c_i, which occur in P or are left after others are enumerated
   Choose the combination C of sub-phrases which make up c_i and minimize the joining times
   End If
   For each sub-phrase s_i in C
   If s_i is in P then
   choose the top 3 candidate translations by the probability p(c|e)
   Else
   Pass on s_i to the OOV translation
   End if
   End for
   End for

Algorithm 1 illustrates the algorithm of phrase identification and translation. Here we think a phrase consisting of less 4 words is "basic". Our idea is to find the most spanning sub-phrases both in the question and in the phrase table. For example, for the query "what is human genetic sequence determination", "human genetic sequence determination" is extracted with the template. And then we find 6 sub-phrases in the phrase table, i.e. "human", "genetic", "sequence", "determination", "human genetic sequence" and "genetic sequence". However, "human genetic sequence" and "determination" make up the whole constituent and have the minimal joining times. So the combination is what we want. In order to provide more relevant key words, the system is designed to output multiple candidates. We rank the candidates by p(c|e), the probability to translate an English phrase e into a Chinese one c.

If the phrase/word is a particle after enumeration or it occurs in many questions, we will remove it from the

Figure 2. The results of different index units and feedback document number



result because we think they are useless for retrieval. In addition, we find if the first letter of each word in a word sequence is capitalized, the word sequence is likely to be a proper noun, e.g., "Yasukuni Shrine", "Edmund Ho Hau Wah", "Balkan Syndrome", etc. Thus, we take the phrase matching the rule as a whole sub-phrase and search for its translations.

Through experiments, we find that the CLIR system achieves the optimal performance when 3 translations with highest probability for each sub-phrase are given. In order to make each key word equal, we reproduce the translation of a phrase three times if it has less than three candidates. We also conduct a experiment on the NTCIR-5 test set, which only outputs the best translation of the whole phrase, but find it works badly. We think it is because the SMT system aims to translate phrase more precise while CLIR need more relevant translations to indicate the information need.

3.3 OOV words translation

In our system, an OOV word is defined as the one not appearing in the phrase table. Our OOV translation algorithm is based on search engines combining with artificial rules and statistics measures.

At first, we construct an English query for Chinese pages based on the OOV word and submit it to a search engine, e.g. BAIDU (http://www.baidu.com). To make trade-off between the relevance and the coverage, we collect the top 200 snippets returned by the search engine. A snippet is the abstract of a page extracted by the search engine, which often includes a title and a short text near the key words. After duplicated snippets deletion and word segmentation, we extract the possible translations within a predefined window. The words occurring around the keywords in less than 20 words distance are considered as the candidate translations.

We think many factors determine if a Chinese word is the translation of an English key word. Therefore, we use a linear feature function to capture the idea:

$$score(c,e) = \sum_i \omega_i f_i(c,e) \qquad (1)$$

in which $f_i(c,e)$ is the feature function and ω_i is the weight of the function.

We find the candidates appearing directly behind the key words with a pair of parenthesis are more likely to be the right translation. This feature is considered with a high priority and the feature function $f(c,e)$ is an indicator one equal to 0 or 1. Another important feature is the similarity between the transliteration of a candidate and the keyword. Each word is transliterated according to Chinese Pinyin and compared with the English word. We take the longest common sub-string length as feature value. And of course, the co-occurrence probability of the key words and the Chinese word is an important feature, too. All weights are tuned manually in a heuristic and greedy way. Initially, we assign greater weights to some important features. Then we fix them and tune the weights for other features. Following the way in the phrase translation, we choose the top 3 candidates to increase the coverage.

4 Module of document retrieval

4.1 Overview

In NTCIR-7, we investigate the effect of traditional ad hoc IR methods in the IR4QA task. And then we propose a linear combination method based on different

Table 1. The results of different IR models and different index units for Chinese monolingual IR on the NTCIR-5 test set

IR model	Mean Average Precision								
	T run			D run			N run		
	Unigram	Bigram	Word	Unigram	Bigram	Word	Unigram	Bigram	Word
tf_idf	0.2497	0.2664	0.2515	0.2164	0.2465	0.2052	0.2716	0.3032	0.2729
fb_tf_idf	0.2724	**0.3162**	0.2950	0.2569	0.3088	**0.2608**	0.2844	0.3362	0.2837
okapi	0.2609	0.2646	0.2582	0.2065	0.2393	0.1769	0.2835	0.3172	0.2737
fb_okapi	0.2931	0.3005	**0.3277**	0.1946	**0.3130**	0.2345	0.1678	**0.3390**	0.2773
kl_jm	0.2369	0.2366	0.2499	0.2218	0.2151	0.1851	0.2916	0.2970	0.2685
kl_dir	0.3031	0.2615	0.2706	0.2588	0.2270	0.2013	0.3154	0.3019	0.2790
kl_abs	0.3045	0.2519	0.2578	0.2302	0.2217	0.1823	0.3183	0.3013	0.2624
mixfb_kl_dir	**0.3443**	0.2860	0.3057	**0.2933**	0.2983	0.2513	**0.3343**	0.3378	**0.2992**

Algorithm 2. Parameter tuning algorithm

Input : model set M={M_1, M_2, ... M_n}
Output : parameter set $\Lambda = \{\alpha_1, \alpha_2, ..., \alpha_n\}$

Let D is the model set with fixed parameter in the combination.
Let maxMAP to be the best MAP score.

Step 1: Initialize $D = \varnothing$, max $MAP = 0$
Step 2:
(1) for $\forall < M_i, M_j >$ in model set M where i ≠ j
tune their parameter $<\alpha_i, \alpha_j>$ to get the best MAP score $MAP_{i,j}$
(2) Fix the parameter of the model pair $< M_f, M_s >$ with the best MAP score. Remove M_f, M_s from model set M and append them into D. Add the parameters of M_f, M_s into Λ
Step 3 :
while $M \neq \varnothing$
(1) for each model M_k in M
tune its parameter along with the models in D and their fixed parameter to get the best MAP score MAP_k
(2) Fix the parameter of the model M_l with the best MAP score of the combination model. Remove M_l from M and append it into D.
Add its parameter into Λ.

index models. A greedy search algorithm is used to tune the parameter setting to achieve the locally optimal performance. A lot of experiments on different data set are conducted to adjust the parameters of the system.

There are two categories of models in information retrieval. One is retrieval model and the other is index model. Retrieval model includes Boolean model, vector space model, probabilistic model, language model etc[6, 7, 8, 9]. Index model is specified by the index term, e.g. unigram, bigram, word, etc. Previous studies show different models have different performances on different IR tasks[10]. However, we think results of various models are complementary for each other. If we combine all results into one in a proper way, the final result should have a better recall and precision.

Thus, motivated by the idea of system combination in SMT, we design a combination model for a robust and better IR performance. The final score of a document for a query is a summed interpolation value of different models as follows:

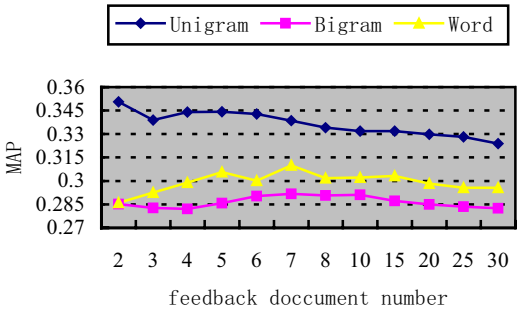
$$score(d,q) = \sum_j \alpha_j score_j(d,q) \qquad (2)$$

where $score_j$(d, q) is the original score assigned by an IR model and α_j is the weight of the model.

In ICT-Crossn, we prefer index models to retrieval models to construct our combination model. There are two reasons for our choice:

- Various retrieval models often generate very different document lists which are hard to be combined. On the contrary, the similar models with various index units often generate similar lists with minor difference, which makes the combination easy to carry out.
- Index model combination is simpler than retrieval model. Various retrieval models have many parameters which influence the performance. But not all parameters are in the same level, so it is hard to adjust them in a linear function. However, for index model combination, we can adopt the same IR model but different index units. Thus, the parameters for each model can be tuned within a linear formula.

Figure 3. The results of different index units and feedback term count



To tune the parameters, we define the object function in formula 2 as follows:

$$\delta(\bigcup_{i=1}^{n} Model_i) = MAP(\bigcup_{i=1}^{n} Model_i) \qquad (2)$$

here $\bigcup_{i=1}^{n} Model_i$ is the set of different models. The goal is to find the parameter setting maximizing the MAP score of the combination model. The algorithm is used as Algorithm 2.

4.2 IR model selection

In order to choose the optimal IR model that fits for the NTCIR data set, we conduct a series of experiments on the NTCIR-5 test set. Table 1 illustrates our experiments on the NTCIR-5 C-C tasks. The IR system we use is Lemur with default settings. We choose 8 candidates: tf_idf (simple tfidf model), fb_tfidf (tfidf model with feedback), okapi (simple okapi model), fb_okapi (okapi model with feedback), kl_jm (simple language model with JM smoothing), kl_dir(simple language model with Dirrichlet smoothing), kl_abs (simple language model with Absolute Discounting smoothing), mixfb_kl_dir (language model, Dirrichlet smoothing, with feedback). We conduct three group experiments. The first group uses the title of each topic (T run). The second group only uses the description (D run) and the third uses the narrative (N run). Table 1 shows that the simple language modeling, with Dirichlet smoothing and feedback, outperforms other models. Based on the results, we think the last model (mixfb_kl_dir) is the best for our task.

Table 2. Development set in NTCIR-7

Task	Question Type	Question number
EN-CS	Event	14
	Relationship	10
	Biography	21
	Definition	33
CT-CT	Event	8
	Relationship	10
	Biography	24
	Definition	29

4.3 Parameter optimization

The mixfb_kl_dir model has parameters of itself and the default setting may not be the best one. We adjust the number of feedback document and the count of feedback term in the update stage of the model to get the locally optimized performance for different indexing strategies. Figure 2 and Figure 3 illustrate the IR performance with different feedback document number and term count for each indexing strategy on the NTCIR-5 C-C task using only the title of each topic. In Figure 2, the count of feedback term is set to 100.

4.4 Index unit combination

Previous studies have shown that index unit combination can improve IR performance using unigram and bigram[11]. We further investigate possible combination of different index units including unigram (U), bigram (B) and word (W) because we think words in Chinese are important meaning units.

Two questions arise when combining different index units. The first is that the scores of a document assigned by various index models are not theoretically comparable and therefore should not be summed directly. We normalize the score of each document in the returned list with the highest score, i.e. the score of the top document. Then all scores fall into the range [0,1]. The second is that when the combination should be performed. There are three strategies: (1) Each index model completes its initial retrieval and the system combines their scores before feedback; (2) Each index model finishes retrieval and feedback respectively and then the system combines their final score; (3) Each index model finishes its initial retrieval, and then the system combines their scores before feedback. After feedback, the system combines all scores again to get the final results. Through the experiments, we find the second works better than the others.

5 Experiments on the dry-run set

5.1 Development set construction

Figure 4. The results of index model combination for the EN-CS task on the dry run set

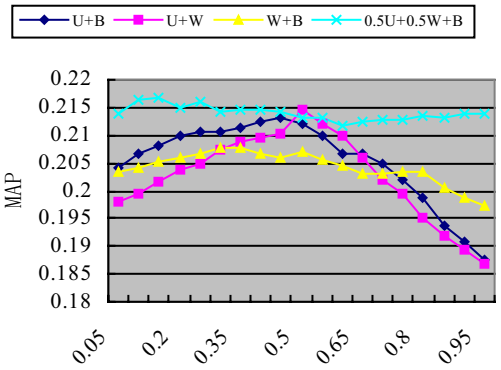
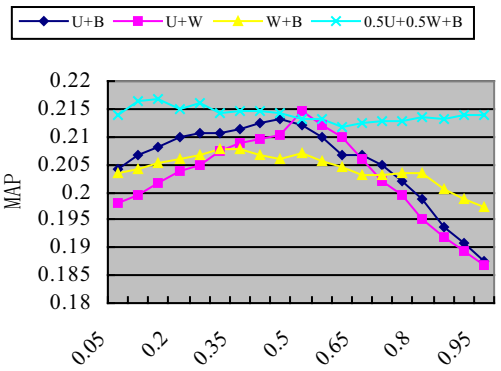


Figure 5. The results of index models combination in EN-CS task for dry run



The test set in IR4QA differs from that in previous collections. Thereby, the metric of performance of IR systems has changed. So we don't expect the parameter setting that works well in NTCIR-5 is still useful in NTCIR-7. Parameters should be adjusted on the test set similar with the official set. Fortunately, the organizers released the dry run set before the formal run. We construct our development set both for EN-CS and CT-CT tasks based on the dry run set. We download the dry run set and the answers through the EPAN[4] interface. For each question, documents that contain an answer are marked as relevant. Our EN-CS development set contains 78 topics and CT-CT set contains 71 topics. Table 2 gives a detailed illustration about our development set.

5.2 EN-CS cross-language information retrieval

We choose language model with Dirrichlet smoothing and feedback, as our primary retrieval model. Unigram, bigram and word indices are created separately. The parameter setting is tuned using the methods introduced in section 4.1. The MAP score is 0.1847 on the unigram index, with 2 feedback documents and 180 feedback

[4] http://aclia.lti.cs.cmu.edu:8080/epan/index.jsp

terms. The MAP score reaches 0.2018 on bigram index, with 3 feedback documents and 160 feedback terms, and the MAP score gets 0.1955 on word index, with 9 feedback documents and 80 feedback terms.

Figure 5 shows the combination results of two index models. The best performance is shown in table 3.

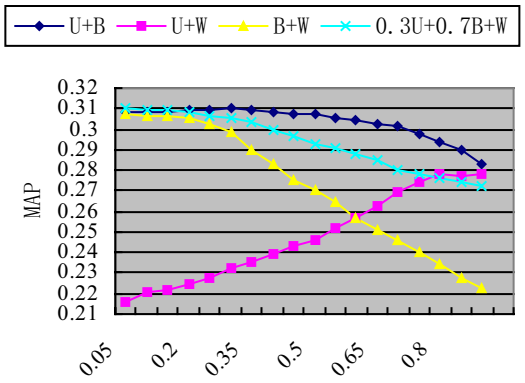
- U+B: the highest MAP 0.2131 when 0.45U+0.55B.
- U+W: the highest MAP 0.2146 when 0.5U+0.5W.
- W+B: the highest MAP 0.2079 when 0.35W+0.65B.

Then we fix the weight of unigram and word in the interpolation, adjust the weight of bigram for some steps and get the best performance with MAP score of 0.2168.

5.3 CT-CT monolingual information retrieval

We convert all the documents and topics in CT to CS because Lemur and ICTCLAS only recognize Chinese characters with GB encoding. The process of CT-CT IR is the same as the retrieval process of EN-CS CLIR task. The MAP score is 0.2722 on unigram index, with 3 feedback documents and 10 feedback terms. The score of bigram is 0.3077 when 6 feedback documents and 10 feedback terms are used. The MAP score is only 0.2057 on word index, with 6 feedback documents and 80 feedback terms. The reason for the poor performance of word index is that the encoding conversion is based on Chinese character but the words for many objects are different in Chinese mainland and Taiwan, especially proper nouns. For example, Bush is always translated into "布什" in simplified Chinese but "布希" is often used in Taiwan and Hong Kong. So there are many word segmentation errors in the word index and thus the performance is degraded. The results of index unit combination are shown in Figure 6.

Figure 6. The results of index model combination in CT-CT task for dry run



6 Results on NTCIR-7 evaluation

We submit 5 runs for EN-CS task and 4 runs for CT-CT task. They are listed as following:

- MITEL-EN-CS-01-T: our primary system for EN-CS task, combined in the scheme of 0.5U+0.5W+0.15B, which outperforms others in the dry run set and uses only the question part of each topic.
- MITEL-EN-CS-02-T: the combination scheme is 0.5U+0.5W, using the question part of each question.
- MITEL-EN-CS-03-T: the combination scheme is 0.45U+0.55B, using the question part of each question.
- MITEL-EN-CS-04-D: the combination scheme is the same as primary system, but only using the narrative part of each topic.
- MITEL-EN-CS-05-TD: the same as the primary system, except using both question and narrative part of each topic.
- MITEL-CT-CT-01-T: our primary system for CT-CT task. The combination scheme is 0.3U+0.7B, using the question part of each topic.
- MITEL-CT-CT-02-T: the combination scheme is 0.3U+0.7B+0.05W, using the question part. We use it as our secondary system because of the poor performance of word index.
- MITEL-CT-CT-03-D: the same as primary system except using the narrative part.
- MITEL-CT-CT-04-T: only use bigram index and the question part.

Table 3 shows the performance of each run of ICT-Crossn on the IR4QA subtasks in the NTCIR-7 formal runs. Our runs achieve a good performance in both subtasks. In the EN-CS CLIR subtask, MITEL-EN-CS-03-T works well even when compared to the monolingual runs. In CT-CT task, our runs rank top 4.

According to the official report of NTCIR-7[12], ICT-Crossn has the highest coverage over relevant documents. Our system also has a high recall on the dry run set, but the precision is not satisfying. We think the rea-

Table 3. The performance based on real qrels in the NTCIR-7 formal result

Task	Run id	Mean AP	Mean Q	Mean nDCG
EN-CS	MITEL-EN-CS-01-T	0.5849	0.6005	0.7949
	MITEL-EN-CS-02-T	0.5693	0.5858	0.7847
	MITEL-EN-CS-03-T	**0.5959**	**0.6124**	0.7947
	MITEL-EN-CS-04-D	0.5789	0.5950	0.7907
	MITEL-EN-CS-05-TD	0.5898	0.6058	**0.8003**
CT-CT	MITEL-CT-CT-01-T	0.5791	0.5963	0.7835
	MITEL-CT-CT-02-T	**0.5839**	**0.6018**	**0.7873**
	MITEL-CT-CT-03-D	**0.5839**	0.6013	0.7869
	MITEL-CT-CT-04-T	0.5645	0.5783	0.7648

son lies in the difference between the metric of performance of traditional IR and IR4QA. For example, given a question like "Who is Hu Jintao", the user wants to know some details about Hu. In the dry run set, a document that introduces the biography of Hu is the best answer. Even the key word "Hu Jintao" only appears once. However, The documents about his visit to Japan are not listed in the answer, but it ranks high in a traditional IR system because "Hu Jintao" appears many times. In other words, the traditional IR methods are not fully suitable in IR for QA tasks.

7 Conclusion and future work

In NTCIR-7, IR4QA is set up as an evaluation task for the first time. Although it is obvious that the traditional CLIR methods are not fully suitable for the new IR4QA test, they are still the most straightforward scheme to deal with the problem.

ICT-Crossn is based on the traditional CLIR framework. First, the system translates the questions into the target language of the corpus with a phrase table generated by a SMT system and an OOV word translation module based on search engines. Second, the system performs information retrieval on the corpus with different index units, and combines the document lists into the final result. To obtain the optimal models and the parameters, we construct the development set based on the dry run set. The results on the official set show our system achieves a good performance.

However, there are still some drawbacks with our methods. Firstly, our translation module hasn't yet solved the translation of OOV words well. E.g., our system cannot identify the phrase "natural gas hydrates" because it neither appears in the phrase table nor has distinct features, which results in the improper translations and the bad performance of retrieval. Secondly, since the judgment metric of document relevance of IR4QA is different from the one of CLIR, the order of the document list should be different from the one of a CLIR system. We attempt to add re-ranking techniques similar with the one of I2R in the document retrieval module to bring forward the documents containing the answers[13]. However, the performance on the development set is not satisfying. Therefore, we plan to improve the method of phrase identification and the method filtering the candidate translations extracted from the web pages returned by search engine. In addition, we try to explore more effective techniques of re-ranking to increase the precision of our system in the IR4QA task.

Acknowledgments

The work is supported by Hi-tech research and development program of China (863 program), Contact No. 2007AA01Z438 and Contact No. 2006AA010108. We would thank to Dr. Zhongjun He for his help.

References

[1]Teruko Mitamura, Eric Nyberg, Hideki Shima, Tsuneaki Kato, et al. Overview of the NTCIR-7 ACLIA: Advanced Cross-Lingual Information Access. To appear in the proceedings of NTCIR-7, 2008.

[2]Ying Zhang, Phil Vines. Using the Web for Automated Translation Extraction in Cross-Language Information Retrieval. In the Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 2004, pp.162-169.

[3]Gaolin Fang, Hao Yu, and Fumihito Nishino. Chinese-English Term Translation Mining Based on Semantic Prediction. In Annual Meeting of the ACL Proceedings of the COLING/ACL on Main conference poster sessions, 2006, pp.199 – 206.

[4]Franz Josef Och, Hermann Ney. The Alignment Template Approach to Statistical Machine Translation. In Computational Linguistics, Volume 30, Issue 4, 2008, pp.417 - 449

[5]Chengye Lu, Yue Xu Shlomo Geva. Translation disambiguation in web-based translation extraction for English-Chinese CLIR. In Symposium on Applied Computing Proceedings of the 2007 ACM symposium on Applied computing, 2007, pp.819- 823.

[6]J. Lafferty and C.X. Zhai. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In the Proceedings of ACM SIGIR, 2001, pp.111-119.

[7]J. Ponte and W.B. Croft. A language modeling approach to information retrieval. In the Proceedings of ACM SIGIR, 1998, pp.275-281.

[8]C.X. Zhai and J. Lafferty. A Study of Smoothing Methods for Language models Applied to Ad Hoc Information Retrieval. In the Proceedings of ACM SIGIR, 2001, pp.334-342.

[9]C.X. Zhai and J. Lafferty. Model-based Feedback in the Language Modeling Approach to Information Retrieval. In the Proceedings of the tenth international conference on Information and knowledge management, 2001, pp.403-410.

[10]J.Y. Nie., J. Gao, J. Zhang and M. Zhou. On the use of words and n-grams for Chinese information retrieval. In the Fifth International Workshop on Information Retrieval with Asian Languages, IRAL-2000, 2000, pp.141-148.

[11]L.X. Shi, J.Y. Nie. Using Unigram and Bigram Language Models for Monolingual and Cross-Language IR. In the Proceedings of NTCIR-6 Workshop Meeting, 2007, Tokyo, Japan, pp.20-25.

[12]Tetsuya Sakai, Noriko Kando, Chuan-Jie Lin, Teruko Mitamura. Overview of the NTCIR-7 ACLIA IR4QA Subtask. To appear in the proceedings of NTCIR-7, 2008.

[13]L.P. Yang and D.H. Ji. I2R at NTCIR5. In the Proceedings of NTCIR-5 Workshop Meeting, 2005, Tokyo, Japan.