Query Expansion and Machine Translation for Robust Cross-Lingual Information Retrieval

Ni Lao, Hideki Shima, Teruko Mitamura, and Eric Nyberg
Language Technologies Institute
School of Computer Science
Carnegie Mellon University

Abstract

In this paper, we describe the Information Retrieval subsystem of JAVELIN IV, a question-answering system that answers complex questions from multilingual sources. Our research focus is on different strategies for query term extraction, translation, filtering, expansion and weighting, including a novel alias expansion technique using lexico-syntactic patterns learned with weakly-supervised algorithm. In the NTCIR7 IR4QA evaluation, our retrieval system achieved 59% and 59% MAP in the Chinese-to-Chinese and Japanese-to-Japanese subtasks, respectively. We provide a rationale for the retrieval system design, and present a detailed error analysis for our formal run results.

Keywords: *Query Expansion, Key Term Filtering, Translation Strategy*

1. Introduction

This paper describes the insights we gained from our participation in the English-Japanese (EN-JA), Japanese-Japanese (JA-JA), English-Simplified Chinese (EN-CS) and Simplified Chinese-Simplified Chinese (CS-CS) tracks in the NTCIR-7 IR4QA evaluation. We begin by briefly describing our system architecture; we then describe each module in the retrieval subsytem, including pattern-based query type classification, a combined translation strategy, key term expansion and filtering, and learning-based key term weighting. The analysis includes training and evaluation on development data as well as evaluation on formal run data.

We present the JAVELIN IV architecture and its modules in Section 2. Sections 3 and 4 describe the question analysis and retrieval strategist components, respectively. For each of these components we conducted experiments using the training set developed for the ACLIA task at NTCIR-7. This preliminary evaluation helped us to decide on the best overall system design for the formal run. Section 5 presents results from the formal run, and discusses issues we discovered in analyzing the results. Section 6 presents conclusions and future work.

We designed our retrieval algorithms and tuned the integrated retrieval subsystem using examples from the ACLIA training set, which has 101 and 88 topics for Japanese and Simplified Chinese, respectively. The topics in the training set do not include relevance judgments provided by assessors; the ACLIA topic developers provided a list of answer-bearing passages, along with the identifier of the document containing each passage. During training, we assumed that the documents provided for each topic were relevant to the topic.

For each type of Chinese question, 7 topics were randomly picked for testing from the ACLIA training set, and the remaining topics were used for training our system. For evaluation, we used the following metrics: P/R, MRR, and P@X for document retrieval.

For Japanese, we focused on maximizing an F1 measure during training, although F1 is not usually used for measuring document retrieval performance. Nevertheless, It is an appropriate measure when maximizing the accuracy of the subsequent phase of answer extraction, which takes only the top 10 documents and ignores document rank.

2. Javelin IV Architecture

Javelin IV is a Question Answering (QA) system for complex questions. Javelin IV has a pipeline architecture consists of four main modules:

- **Question Analyzer:** Responsible for analyzing the question to determine the information need (question type, answer type, key terms, etc.).
- **Retrieval Strategist (RS):** Responsible for extracting a ranked list of answer-bearing documents, using a query formulated using information provided by the Question Analyzer.
- **Information eXtractor (IX):** Responsible for extracting and scoring/ranking answer candidates from the answer bearing documents.
- **Answer Generator (AG):** Responsible for removing duplicates and selecting/filtering answers

All the modules are designed to be language independent, and utilize uniform interfaces to MT and NLP services to support run-time loading of language-specific resources. This paper focuses on the Question Analyzer and RS modules which comprise the retrieval subsystem evaluated in the IR4QA task. More details regarding the other parts of Javelin IV can be found in [9].

3. Question Analysis Module

The Question Analyzer is responsible for two main tasks: predicting the answer type and identifying key terms from the question. (See Figure 1)

The answer type is predicted using manually-created surface patterns based on TREC complex questions and the ACLIA training set. We found that, unlike most factoid questions, complex questions typically have simpler sentence structures, which makes creating broad-coverage surface patterns more feasible. For monolingual retrieval, patterns are written in Chinese or

Japanese; for cross-lingual retrieval the patterns are written in English.

Our Chinese and Japanese configurations use slightly different algorithms to identify key terms from the question, based on the NLP tools available. For Chinese, the question is first parsed into a syntactic tree. Then, for each NP node in the tree, the corresponding words from the sentence are concatenated as a key term. For Japanese, we extract NP chunks and then filter out noisy terms based on document frequency.

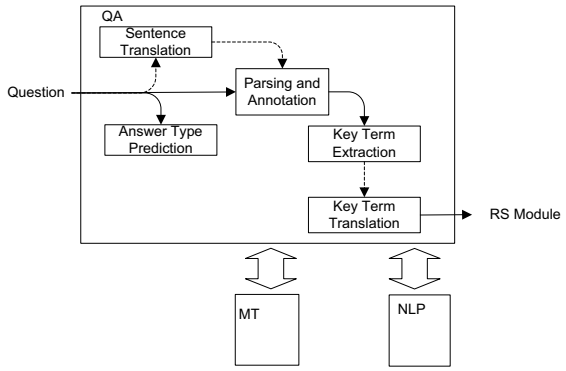

Figure 1 : Question Analyzer

For cross-lingual retrieval, we can either translate the English question and extract key terms from the Chinese and/or Japanese translation, or directly extract key terms from the English question and then translate them into Chinese and/or Japanese. From evaluation on the training set, we found that combining key terms from both approaches gave the best performance.

3.1. NLP Preprocessing

We developed an NLP preprocessing module to integrate all the NLP tools used to process question and answer texts: text segmenters, POS taggers, syntactic parsers, etc. Different sets of tools are invoked depending on the setting of a text language parameter. The tools we used for Japanese are MeCab (morphological analyzer) and CaboCha (internally uses named entity recognizer & NP chunker). The tools we used for Chinese are MSR Segmentor (word segmentation and named entity recognition) and a POS tagger and syntactic parser developed at CMU. The tools we used for English are the POS tagging and syntactic parser capabilities provided by the Charniak parser.

Table 1 Sample answer type patterns for DEFINITION and BIOGRAPHY questions

Variables

person = "(人|方|アーティスト|人生|人物|氏|さん|博士|先生|社長|大統領|代表|女優|作家|俳優|教授)"

how = "(どのような|どんな|どういった|どういう)"

DEFINITION		**BIOGRAPHY**	
(何|なん|こと)ですか	0.8	誰	1.0
(何|なに)？	0.8	何者	1.0
how,	0.7	どなた	1.0
とは(何|なに|なん)	0.7	人生|経歴	1.0
について	0.4	how+person,	0.9
のことを	0.3	person,	0.3
特徴は	0.8	person+"について	0.9
何	0.1	@Person@	0.2
@Person@	-0.2		

3.2. Answer Type Classification

Since the questions in training set do not vary much in their surface structure, we used hand-crafted lists of weighted cue words (see Table 1 for a sample) to identify answer types. For each target language, we repeatedly adjusted terms and weights until accuracy on the training set reached 100%.

3.3. Key Term Extraction

The key term extractor is responsible for creating a list of terms that will be useful for both retrieving potentially relevant answer-bearing documents and subsequently extracting answers from those documents. Using the NLP tools described in Section 3.1, the key term extractor identifies a set of noun phrases, which is also extended with any named entities that were recognized.

3.4. Key Term Translation

We used a previously-developed meta-translation engine [4] to combine translation results from multiple dictionaries and online translation systems, in the following steps: 1) retrieve translations from multiple translators; 2) remove any failed translations; 3) assign probability to each possible translation based on a voting model. Failed translations are detected by the existence of English words in the translated text. Weights are assigned to translator outputs based on observed accuracy of the translators on the training set. For example, assume that we have three translators, Amikai, Google and WorldLingo, with weights of 1.0, 1.1 and 1.2 and the following translations for"Bin Laden", respectively: 宾拉登, 本・拉丹, 宾拉登 ; the final scores assigned to the translations would be: 宾拉登 = (1.0+1.2) / (1.0+1.1+1.2) = 0.67, 本・拉丹= 1.1 / (1.0+1.1+1.2) = 0.33.

3.4.1. Resources

The dictionaries and online translation systems we used for both Chinese and Japanese include:

- the Wikipedia inter-language title link dictionary;
- Google (http://translate.google.com);
- Amikai (http://standard.beta.amikai.com) ;
- WorldLingo (http://www.worldlingo.com) ;
- Systran (http://babelfish.altavista.com); and
- CrossLang (http://honyaku.yahoo.co.jp) .

The resources used only for Chinese include the LDC Chinese English Lexicon and Chinese Year Dictionary. The resources used only for Japanese include:

- Honyaku (http://mt.fresheye.com) ;
- BizLingo (http://www.excite.co.jp/world) ;
- Chasen Noun Dictionary;
- Japanese Year Dictionary.

3.4.2. Key Term vs. Sentence Translation

To compare different question translation strategies we performed experiments using CS training data (Figure 2). We found that key term translation works better than sentence translation for Biography and Relation questions.

One reason is that sentence translations generally have lower accuracy in key term translation. However, sentence translation works better than key term translation for Event and Definition questions.

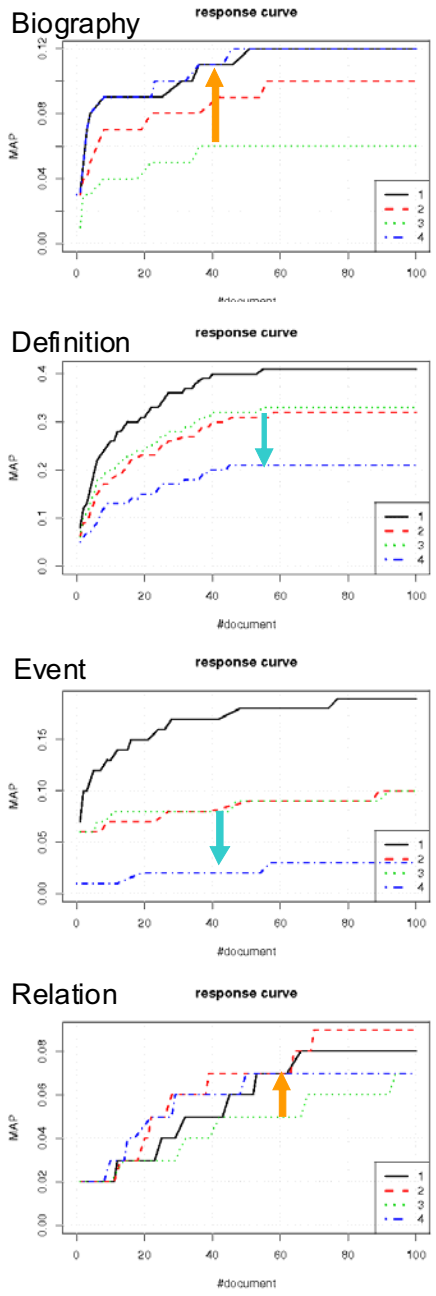

Figure 2 Question Translation Strategies. Horizontal axis is the number of top documents drawn from the system responses; Vertical axis is the mean average precision (MAP). Curves are 1) Monolingual result, 2) Translate both, 3) Translate sentence only, 4) Translate key terms only

The reason for this is that translated questions will have a tree structure which is similar to the structure of answer sentences. But this principle does not seem to apply to Biography questions (which have very simple structures), or Relation questions (which have overly complex structure, such that translators are much less likely to output sentences with correct syntax).

Based on these observations from the training data, we combined the terms from both sentence translation and key term translation for the formal run. However, a smarter strategy would be to automatically choose a translation strategy based on the predicted topic type.

We also compared translation strategies for an EN-JA training data set. The result in Table 2 shows the F1 value measured for three settings: (A) Question translation + same key term extractor used in JA-JA, (B) key term extractor for English + term translation, and a combination of both approaches A and B. Results show that the combination worked better than either approach alone on the training data set.

Table 2 Translation Strategies Comparison for EN-JA

Filtering Strategy	F1
(A) EN-JA Question Translation	0.202
(B) EN-JA Key Term Translation	0.240
Combination of (A) and (B)	0.265

4. Information Retrieval Module (RS)

The RS module uses the Indri search engine to index and search the target corpus. We implemented three possible retrieval units: document, block, and sentence (see Figure 3). From our experience, document retrieval plus sentence/clause extraction gives the best results. The result of just the document retrieval step was submitted to to IR4QA track as our retrieval system output.

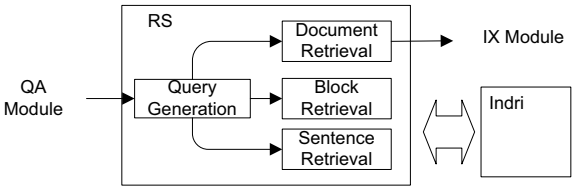

Figure 3 Retrieval Strategist (RS)

Our Japanese and Chinese systems extend the base retrievla system in different ways to explore different aspects of the retrieval problem; these are described below.

4.1. Extension for Chinese: Query Formulation

For Chinese, three types of information are used to create the Indri query for document retrieval:

- The key terms from the Question Analyzer module (extracted from the question);
- Cue terms indicating features such as family, past tense, birth, death, causal relations, etc. 20 word groups were manually selected from the Dictionary of Synonymous Words [3]. For example, birth-related terms include 出世(born), 出生(birth), 祖籍 (home town), etc.;
- Named Entity (NE) types.

Key terms are given uniform weight in the monolingual configuration and weights from the MT

module (normalized for a total weight of 10.0) in the cross-lingual configuration. The weights for cue terms and named entity types are trained automatically as described in our NTCIR-7 CLQA paper [9]. This is done by normalizing the weight vector of the answer extraction models in the IX module to have total weight (1-norm) of 100. Because the IX module trains extraction models separately for each type of query, our IR4QA system will produce different weight vectors according to the predicted query type.

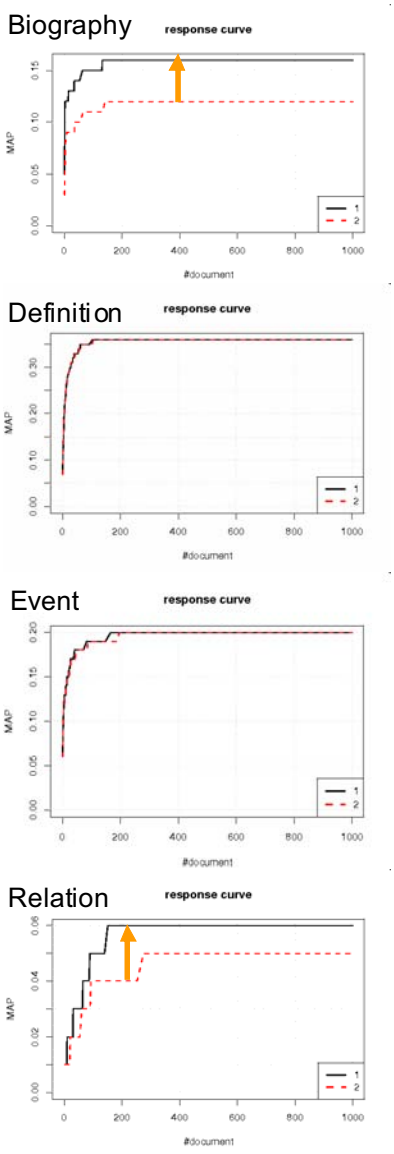

Figure 4 Retrieval Strategies. Horizontal axis is the number of top documents drawn from each system response. Vertical axis is the mean average precision (MAP). Curves are 1) using key term + NE + Cue Term, 2) using key term only.

Example:

#weight(10 #2(周恩来) //key term
7.02 #syn(出世 出生 祖籍…) … //cue term
1.96 #any:CARDINAL 1.61 #any:DATE …) //NE

As we see in Figure 4, we found that the use of NE and cue terms benefit some types of topics (Biography and Relation); however, the cue terms were not used in our formal run, because they produce Indri queries with hundreds of terms that execute very slowly. In order to make this technique applicable to real applications, we will need to improve retrieval speed by indexing the cue terms.

4.2. Extensions for Japanese Run

We analyzed the baseline IR system results on training data and found two problems: noisy key terms and vocabulary mismatches. To solve these issues, we implemented two solutions that might also work for the formal run; these are described below.

4.2.1. Key Term Filtering

Let us define a "noisy term" to be a term that matches too many irrelevant documents and does not contribute many relevant documents to a specific retrieval task. Through analysis of our baseline system, we found a proportion of the extracted key terms to be noisy. Lack of context in word-by-word translation often results in ambiguous, incorrect translations which introduce terms that match many irrelevant documents. Even in monolingual retrieval, terms extracted from the question can be very noisy given this definition. These observations motivate us to ask the following research question: How can we mitigate the negative effect of noisy terms in retrieval? We hypothesized that filtering common terms contributes to retrieval performance, and implemented the following term filtering methods:

DF Filter removes a term based on Document Frequency, assuming terms that appear in more than *N* documents in the corpus are common terms that should be filtered. We set *N* to be 7000, based on experiments on the training dataset. For example, assume that "Moyamoya Disease (もやもや病) " and "disease (病気) " are terms extracted given the question "What kind of disease is Moyamoya Disease?" (もやもや病とはどんな病気？)". Intuitively, the former term (which appears 7 times in the corpus) is much more precise than the latter (which appears 7864 times) when matching documents to satisfy the information need.

Stopword Filter removes a term that appears on the stop word list. We hand-crafted a stop word list consisting of 13 words observed in the training dataset. For example, "event (出来事) " from "Tell me major events occurred in Tuvalu (ツバルで起きた主要な出来事について教えてください。)" is an example of a stop word that is filtered.

Overlap Filter removes a term if there is a longer term which contains it. For example, suppose "bomb (爆弾)" and "dirty bomb (汚い爆弾)" are extracted from the question "What kind of bomb is dirty bomb? (いわゆる「汚い爆弾」とはどんな爆弾のことですか。)". The former term will be filtered out because it is contained within the latter term.

Greedy Filter removes key terms until the number of terms is just one for DEFINITION and BIOGRAPHY topics, and two for RELATIONSHIP and EVENT topics.

We performed an experiment on the training data to compare different filtering methods to both a baseline with no filtering and a combination of all four filtering methods. The Greedy Filter was applied last when combined with other filters. The result is shown in Table 3, where the combination of filters is seen to outperform the other approaches.

Table 3 Key term filtering affect on retrieval

Filtering Strategy	F1
No filtering (baseline)	0.420
(A) Document Frequency Filter	0.422
(B) Stopword Filter	0.425
(C) Overlap Filter	0.439
(D) Greedy Filter	0.440
Combination of (A)+(B)+(C) +(D)	0.445

4.2.2. Query Expansion using Alias Patterns from Bootstrapping Learning

If we assume that we have solved the noisy key term filtering problem, and possess a method for generating a noise-free list of key terms, our next focus becomes how to solve *vocabulary mismatch* – a problem that occurs when a query and relevant answer-bearing documents don't match because of surface variations in the text (for example, morphological variants of a verb or noun, or multiple spellings of one person's name). Regular variation can be addressed with basic NLP tools (e.g. word segmentation, stemming), but resolving all the variation in named entity references is a difficult, open problem.

Following examination of the corpus documents, we formulated the hypothesis that the alternate forms of a proper noun can be captured from a corpus using Lexico-Syntactic Patterns (LSP). To test this hypothesis, we obtained alternate forms for proper nouns using LSP learned with a weakly-supervised general-purpose learning framework called Espresso[6]. Figure 5 presents a visual overview of the steps in the Espresso algorithm:

1. For each instance {x, y}, retrieve all sentences containing the two terms x and y.
2. Pattern Induction: all substrings linking terms x and y are then extracted from sentences S{x,y}, and overall frequencies are computed to form P. We adopted the Longest Common Substring algorithm to find patterns, as it is used in a similar pattern acquisition task [7].
3. After a set of patterns are obtained, score all patterns in P according to reliability scores based on approximated PMI statistics.
4. Select only "reliable" patterns to generate new instances.
5. Likewise in step 3, calculate reliability scores for all instances.
6. Select only "reliable" instances and go back to step 1 unless the algorithm has converged.

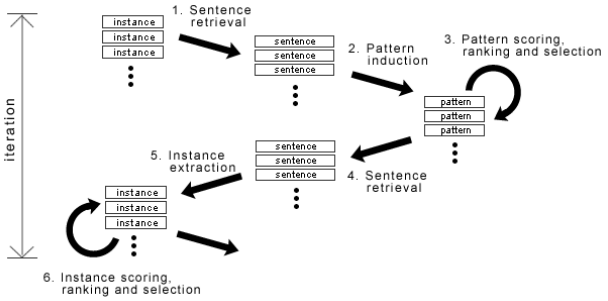

Figure 5 Espresso overview

We hand-crafted 10 to 20 seed instances where binary-argument pairs are alternative forms of each other. As a result of batch time training, we obtained LSPs such as the examples shown in Table 4.

Table 4 Sample of ALIAS LSPs learnt by Espresso

「 <NP2> 」 と さ れる <NP1> が
「 <NP2> 」 と いわ れる <NP1>
「 <NP2> 」 と も 呼ば れる <NP1>
「 <NP2> 」 と も 言わ れる <NP1> が
<NP2> は 「 <NP1> 」
<NP2> （ 日本 版 <NP1> ）
、 <NP2> （ <NP1> ）
<NP2> ・ <NP1> が
を 実効 支配 する <NP2> 、 <NP1>
発売 予定 の 次期 <NP2> 「 <NP1> 」

Using LSPs learned by Espresso, we implemented a novel query expansion technique we refer to as LSP-PRF, where the idea is to add possible alias terms found in pseudo-relevant documents. To illustrate the idea, consider this question from the training dataset: "What is yen-dominated foreign bond? (円建て外債とは何ですか？)". The following steps show how LSP-PRF performs query expansion on "yen-denominated foreign bond (円建て外債)" in order add its alternative form, "Samurai bond (サムライ債)":

1. Retrieve documents with a query made from the key term "yen-denominated foreign bond"
2. Instantiate all LSP patterns with the key term. For this example, let's focus specifically on the "、<NP1>（<NP2>）" pattern. As a result, we have two instantiations, "、 yen-denominated foreign bond （ <NP2> ） " and "、 <NP1> （ yen-denominated foreign bond）".
3. Assuming that the top N documents from the query are relevant (the pseudo-relevance assumption), we apply the patterns to the top N documents. In this example, the instantiated pattern matches "、yen-dominated foreign bond (Samurai bond)". As a result, "Samurai bond" is captured as an alternative form of the original key term.
4. The original query is expanded with the alternative forms found in step 3.

5. IR4QA Formal Run Results and Analysis

In the formal NTCIR IR4QA evaluation [8], our Japanese and Chinese systems achieved the following performance: MAP = 0.59 and rank 3rd for Chinese, MAP = 0.59 and rank 4th for Japanese. For cross-lingual retrieval, our English-Japanese run achieved MAP = 0.43 which was best among all English-Japanese runs.

5.1. Error Analysis for Chinese

Our Chinese runs combined two strategies in three configurations, with mono- and cross-lingual variation:

- Use only query key terms, and use combined translation: CMUJAV-CS-CS-01, CMUJAV-EN-CS-01;
- Use key terms and named entities in query, and use combined translation: CMUJAV-CS-CS-02;
- Use key terms and named entities in query, but only sentence translation: CMUJAV-EN-CS-02.

The performance of these runs are shown in Table 5 and Table 6 below.

Table 5 Performance based on pseudo-qrels (top 10 responses): CS runs; 97 topics.

Run Name	MAP	Rank	Q	Rank	nDCG	Rank
CS-CS-02	0.519	2	0.565	2	0.752	2
CS-CS-01	0.508	3	0.555	3	0.746	3
EN-CS-02	0.430	13	0.473	13	0.671	14
EN-CS-01	0.426	14	0.471	14	0.670	15

Table 6 Performance based on real qrels (top 30 responses): CS runs; 97 topics.

Run Name	MAP	Rank	Q	Rank	nDCG	Rank
CS-CS-02	0.593	4	0.606	5	0.795	4
CS-CS-01	0.590	6	0.603	6	0.794	7
EN-CS-01	0.546	14	0.556	14	0.740	14
EN-CS-02	0.527	15	0.537	15	0.725	16

First, we can see that our system performs relatively better at top 10 responses compared top 30. This is consistent with the observation that our system gets a slightly higher ranking with MAP vs. nDCG, because the former has a faster rate of decline (1/rank) than the later (1/log(rank)).

Second, using named entities improves the result for top-10 by about 1% MAP, which is evident by comparing CS-CS-01 and CS-CS-01 in Table 5.

Third, because the effect of using named entities is small, the difference between EN-CS-01 and EN-CS-02 mainly reflects the difference between the combined translation and sentence translation approaches. Figure 6 shows that the combined approach almost always performs better. There are only two salient outliers, topics T58 and T79. A closer investigation showed that in both cases, the translation has unlawful character for Indri query, which cause Indri returned an empty result. Adding a proper character filter would solve this.

We did a full analysis of all the topics, and grouped them according to the factors that hurt/help system performance (Table 7). Table 8 gives a detailed example for the translation-related issues.

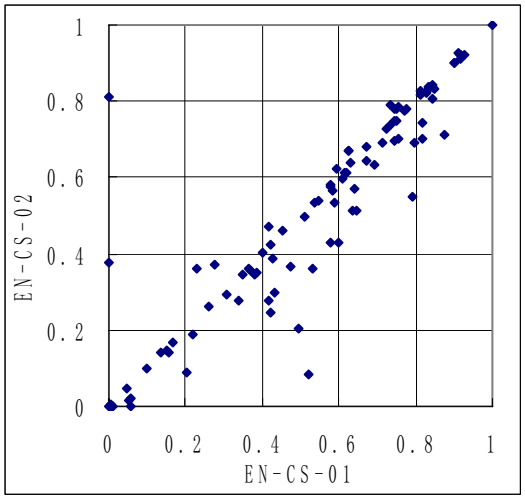

Figure 6 MAP of combined translation (horizontal) and sentence translation (vertical) results

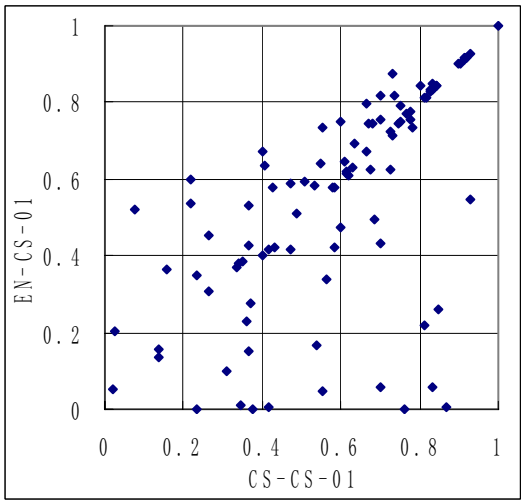

Figure 7 MAP for monolingual (horizontal) and cross lingual (vertical) results

Table 7 Factors that affect performance and topics affected by each factor

Type	Factors	Topics
	Mono Lingual	
bad	Need more weight on title terms	T41, T46, T47, T54
bad	Key word in Chinese query cannot match to corpus	T42, T102, T337
bad	Cannot match acronym	T366
bad	Incomplete stop word in Chinese	T92, T93
	Cross Lingual	
bad	Translation has unlawful character for Indri query	T58, T79
good	MT improves CC result by correctly translating the named entity	T102, T366
good	MT provide variation of word segmentation	T60, T99, T379
good	Parsing error in Chinese question, but not English question or the translation of Chinese question	T81, T379
good	English question have term that can directly match to corpus, while Chinese question does not have	T49
bad	MT bring in bad related terms	T54, T67
bad	Translated term is correct, but cannot be matched to corpus	T42, T62, T71
bad	General term translation failed	T76, T355
bad	Proper noun translation failed	T75, T77, T89, T93, T94, T95,T326, T359, T385
bad	Acronym (proper noun)translation failed	T56, T359

In the corpus, each document has a metadata field called *title*. We found that many titles of relevant documents match well with retrieval queries. If we add more weight to the title field in our Indri queries, query performance should improve.

In many cases our monolingual system failed because terms in the given topic cannot be directly matched with the terms used in the corpus (e.g. T42, T102, T337, T366). Some of these topics (T102, T366) get a performance boost in cross lingual run, because the translation output can be matched with the corpus.

This observation reflects the main benefit gained from term translation: improved robustness through the combination of multiple translators. By using a set of terms instead of a single term to express the relevant meanings in the information need, we improve the likelihood that the system will avoid parsing errors, avoid segmentation discrepancies, and match some answer-bearing document in the corpus. This issue is especially prominent in the corpora used for NTCIR-7 where spellings are very uniform; e.g. *Bin Laden* is expressed uniformly as "本・拉丹" in the Xinhua corpus. Failing to find the correct translation of the named entity will result in total failure for this topic.

Despite the advantages of using multiple translators, translation is still a significant source of error. Most failures are associated with incorrectly translated proper nouns, and the overall effects are usually fatal, resulting in near pessimal performance in many cases (Figure 7). Therefore, there is still significant room for improvement in overall performance by improving the translator. For example, we may consider translating terms collectively instead of independently (e.g., *Jordan* and *basketball*). Another common source of translation error occurs when the system translates Chinese names which include English given names; for example, "Charles Zhang" and "Jerry Yang" should be translated using their original Chinese given name, instead of a translation of their English given name.

Overall, most of the failures in retrieval can be attributed to robustness problems: variation in spelling, errors in segmentation and parsing, and finding related terms for query expansion. Translation can help to alleviate some robustness issues in cross-lingual retrieval, but future work should also improve the robustness of monolingual retrieval, perhaps through the use of language resources for term expansion and a combination of multiple parse tree and segmentation results.

Yet another observation is that Biography and Definition questions are relatively simple, in the sense that by using just the named entities from the question as the Indri query, the system can already achieve the MAP of the top-performing systems (**Table 9**). This is not likely to be the case Event and Relation questions.

Table 8 Examples of translation's effect on performance. The middle three columns show how a term is expressed as a monolingual query, a cross-lingual query, and in the corpus. The last column shows the change in MAP with translation.

Topic	Query	Translation	Corpus	dMAP
T379	邓, 肯	邓, 肯, 邓肯	邓肯	+0.45
T366	澳大利亚, 印度尼西亚	澳大利亚, 澳洲, 印度尼西亚, 印尼	澳, 印尼	+0.36
T102	瓷器, 画画	陶瓷, 绘画	陶瓷 绘画	+0.27
T337	和平 谈判	和平 进程	和谈 进程	+0.23
T99	李宁	李 宁, 李宁	李 宁	+0.19
T74	中 俄	中国, 俄罗斯, 俄国, 苏联	中国, 俄罗斯	+0.18
T60	洛克比	洛克比, 洛 克 比, 洛 克	洛克比, 洛 克 比	+0.18
T49	申奥	申奥, 奥运, 奥林匹克	申奥, 奥运 奥林匹克	+0.15
T98	伏明霞 和 跳水	伏明霞 和 潜水	伏明霞	+0.14
T44	911	九一一	9 · 11 九一一(rare)	+0.04
T42	本拉登	宾拉登, 拉登, 奥萨玛, 欧萨玛	本・拉丹	+0.02
T64	比尔盖茨	比尔・盖茨, 比尔・格茨, 比尔盖茨	比尔・盖茨, 比尔盖茨, 盖茨	0.0
T62	Windows 2000	Windows 2000 窗口, 视窗, 美国微软公司	Windows 2000, 视窗 2000	-0.13
T355	登上	上升	登上	-0.19
T67	捐款	捐赠	捐款, 捐赠	-0.21
T322	安南	安 南, 安南 科菲安南, 科非・安南	安南, 安 南	-0.22
T385	百年 校庆	一百周年 庆祝	100 周年 百年 校庆	-0.26
T359	巴 以	以色列, 以色列人 以色列共和国民, 巴勒斯坦, 巴勒斯坦人	巴 以	-0.37
T71	张朝阳	查尔斯・张, 查尔斯・张, 查尔斯张	张朝阳	-0.38
T77	世纪大阅兵	世纪, 巡游, 游行	世纪大阅兵	-0.41
T333	核试验	核 测试	核试验	-0.5
T89	曾溢滔	N/A	曾溢滔	-0.59
T326	杨致远	杰里・杨	杨致远	-0.59
T95	乔丹	约旦	乔丹	-0.86

Table 9 Topics that perform well with a simple query consisting only of named entities. Numbers in parenthesis are difference in MAP when compared to the top performing system for each topic. Topics with a difference smaller than 5% are shown here.

Topic Type	Topics (MAP differences in percentages)
BIO	T43(-2.1), T55(0), T69(-1.6), T339(-3.2), T340(-2.4), T370(-0.9)
DEF	T80(0), T369(-0.7), T378(-1), T381(-0)
REL	T61(-2.2)

5.2. Error Analysis for Japanese

In the formal evaluation, we used the following configuration settings for our JA-JA and EN-JA runs:

- Run 01: expand query with LSP-PRF algorithm.
- Run 02: expand query with PRF assuming top 5 initially retrieved documents are relevant
- Run 03: expand query with PRF assuming top 30 initially retrieved sentences are relevant

- Run 04: expand query with alias dictionary constructed from inter-page redirection information on Wikipedia
- Run 05: No query expansion is performed

Results are shown in Table 10 and Table 11, where we see that Run 01 with our novel query expansion algorithm LSP-PRF achieved the best performance of all our runs on the real qrels (Table 11).

Table 10 Performances based on the pseudo-qrels (top 10 responses): JA runs; 98 topics.

Run Name	MAP	Rank	Q	Rank	nDCG	Rank
JA-JA-04	0.674	1	0.715	1	0.854	1
JA-JA-01	0.673	2	0.713	2	0.852	2
JA-JA-05	0.670	3	0.711	3	0.851	3
JA-JA-03	0.660	5	0.701	5	0.844	4
JA-JA-02	0.657	6	0.699	6	0.842	6
EN-JA-01	0.441	12	0.474	13	0.621	15
EN-JA-04	0.438	13	0.472	14	0.620	16
EN-JA-05	0.437	14	0.470	15	0.620	17
EN-JA-03	0.429	16	0.462	16	0.612	18
EN-JA-02	0.420	17	0.454	17	0.607	19

Table 11 Performances based on the real qrels (top 30 responses): JA runs; 98 topics.

Run Name	MAP	Rank	Q	Rank	nDCG	Rank
JA-JA-01	0.593	4	0.600	4	0.783	4
JA-JA-03	0.589	5	0.595	6	0.780	7
JA-JA-04	0.585	6	0.591	7	0.778	8
JA-JA-02	0.579	8	0.588	8	0.774	10
JA-JA-05	0.579	9	0.585	9	0.772	11
EN-JA-01	0.426	13	0.434	15	0.603	15
EN-JA-03	0.425	15	0.432	16	0.601	16
EN-JA-04	0.423	16	0.431	17	0.600	17
EN-JA-02	0.419	17	0.427	18	0.597	19
EN-JA-05	0.419	18	0.427	19	0.596	18

Table 12 shows an exhaustive list of terms expanded by LSP-PRF, for three example topics. Useful aliases, or sometimes an answer itself, is underlined. We found that query expansion generally helps on average, but sometimes harms performance at the topic level, as also pointed out by Collins-Thompson [1]. Improving the stability of expansion is a topic for ongoing research.

Table 12 Examples expanded terms (formal run)

Original key terms	Expanded terms (exhaustive)
スーパーカミオカンデ *(Super-Kamiokande)*	ニュートリノ観測装置 *(Neutrino Detector)*, 観測装置 *(detector)*, 素粒子検出装置 *(elementary particle detector)*, 観測施設 *(detection facility)*, 建設には１０４億円 *(construction cost is 10 Billion yen)*
えひめ丸 *(Ehime Maru)*, ＰＴＳＤ	心的外傷後ストレス障害 *(posttraumatic stress disorder)*, 実習船 *(training ship)*, １人が部分的 *(one person partly)*, 主な症状 *(main symptom)*, 対策費 *(expenditures)*, 治療の専門家 *(medical expert)*, 症状 *(symptom)*, 事故 *(accident)*, ２人が部分的 *(two people partly)*,
ＫＦＯＲ, ＮＡＴＯ	コソボ駐留国際治安部隊 (Kosovo-stationed international peacekeeping force), コソボ国際治安部隊 *(Kosovo international peacekeeping force)*, 国際平和維持部隊 (international peacekeeping force), 参加部隊や住民 *(participating troops and residents)*, 任期更新期 *(renewal period)*, フランス人兵士１３人 *(13 French soldiers)*

6. Conclusion and Future Work

This paper described the retrieval system we evaluated in the IR4QA evaluation. The basic findings of this study include the following:

- Translation failure causes the most retrieval failure;
- Although translation helps to improve robustness, monolingual retrieval must be improved;
- The optimal retrieval strategy may differ by topic type;
- Pre-annotation and indexing improve strategies based on additional text features (e.g. cue terms).

Our planned future work includes:

- Improving the robustness of monolingual retrieval through the use of multiple segmenters, parsers, and language resources for term expansion;
- Dynamically selecting a translation strategy based on topic type;
- Adding cue terms to the Indri repository index (e.g. adding / indexing the annotation BIRTH in the same location as occurrences of birth-related words appearing in the corpus).

Acknowledgments

This work was supported in part by IARPA's Advanced Question Answering for Intelligence (AQUAINT) Program. We thank Eric Riebling for his assistance and we also thank the corpus provider for the Japanese and Chinese corpora used in the NTCIR-7 evaluation.

References

[1] Collins-Thompson, K. Robust Model Estimation Methods for Information Retrieval, *Ph.D. Thesis*, Carnegie Mellon University, 2008.

[2] Goldstein, J., V. Mittal, J. Carbonell, and J. Callan. Creating and evaluating multi-document sentence extract summaries. *In Proceedings of CIKM 2000*, 2000

[3] Mei, J., Y. Zhu and Y.Q. Gao, Y. Hongxiang. Edited. 1983. Tongyici Cilin (Dictionary of Synonymous Words), Shanghai Cishu Publisher, 1983.

[4] Mitamura, T., F. Lin, H. Shima, M. Wang, J. Ko, J. Betteridge, M. Bilotti, A. Schlaikjer and E. Nyberg. JAVELIN III: Cross-Lingual Question Answering from Japanese and Chinese Documents, *In Proceedings of NTICIR-6 Workshop*, Japan, 2007.

[5] Nyberg, E., R. Frederking, T. Mitamura, M. Bilotti, K. Hannan, L. Hiyakumoto, J. Ko, F. Lin, L. Lita, V. Pedro, and A. Schlaikjer. JAVELIN I and II systems at TREC 2005, *In Proceedings of TREC'05*, 2005.

[6] Pantel, P. and M. Pennacchiotti. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations, *In Proceedings of ACL 2006*, 2006

[7] Ravichandran, D., E. Hovy. Learning Surface Text Patterns for a Question Answering System. *In Proceedings of ACL 2002*, 2002

[8] Sakai, T., N. Kando, C.-J. Lin, T. Mitamura, H.Shima, D. Ji, K.-H. Chen, E. Nyberg. Overview of the NTCIR-7 ACLIA IR4QA Subtask, *In Proceedings of NTICIR-7 Workshop*, Japan.

[9] Shima, H., N. Lao, E. Nyberg, and T. Mitamura, Complex Cross-lingual Question Answering as Sequential Classification and Multi-Document Summarization Task, In Proceedings of NTICIR-7 Workshop, Japan, 2008.