TREC-9 Experiments at KAIST: QA, CLIR and Batch Filtering

Kyung-Soon Lee, Jong-Hoon Oh, JinXia Huang, Jae-Ho Kim and Key-Sun Choi

Division of Computer Science,

Department of Electrical Engineering & Computer Science,

Korea Advanced Institute of Science and Technology, KORTERM,

373-1 Kusung Yusong Taejon 305-701 Korea

Tel: +82-42-869-5565

Fax: +82-42-867-3565

{kslee, rovellia, hgh, jjaeh, kschoi}@world.kaist.ac.kr

1. Introduction

In TREC-9, we participated in three tasks: question answering task, cross-language retrieval task, and batch filtering task in the filtering task.

Our question answering system consists of following basic components - query analyzer, Named entity tagger, Answer Extractor. First, question analyzer analyzes the given question. Question analyzer generates question type and keywords of the given question. Then retrieved documents are analyzed for extracting relevant answer. POS tagger and Named entity tagger are used for the purpose. Finally, Answer Extractor generates relevant answer.

There are four runs in our CLIR, two runs follow the dictionary and MI information based translation approach (KAIST9xlqm, KAIST9xlqt), another one using the mixture result of two commercial Machine Translation systems (KAIST9xlmt), and the final one is monolingual run (KAIST9xlch). We translated only query and description fields in all four runs.

In batching filtering task, we submitted results for OHSU topics and MSH-SMP topics. For OHSU topics, we have been exploring a filtering technique which combines query zone, support vector machine, and Rocchio's algorithm. For MSH-SMP topics, we use support vector machine simply.

2. Question Answering track

2.1 System Description

Our TREC-9 question answering system consists of three basic components - query analyzer, named entity tagger, and answer extractor. Our system operates on a set of documents retrieved by information retrieval system. For convenience, we worked with the top-ranked document set generated by NIST.

First, a question analyzer analyzes the given question. It generates question type and extracts keywords of the given question. Then top 50 documents retrieved by information retrieval system are analyzed for extracting relevant answer. A POS tagger and a named entity tagger are used for the purpose. Finally, an answer extractor generates relevant answers from named entity tagged documents using question types and keywords analyzed by the question analyzer.

2.1.1 Question Analyzer

A question analyzer parses the given question to identify question types and extract keywords. We define six kinds of question type for the expected answer.

<Person>, <Location>, <Organization>, <Time>, <Currency>, <Measure>

There are five steps for analyzing questions. First, a question is tagged by POS tagger – We use the Brill tagger (Brill, 1995). Second, keywords are extracted from tagged question. The POS tag, which we extract as keywords, is noun, adjective, countable numeric, and verb. However, we exclude category of "be-verb (is, are, was, were)" and "do-verb (do, does, did)". Third, we check an acronym in the given question. If there is a word with capital letters, we assume that it is an acronym and we search an expanded form in the acronym dictionary. Once, there is an expanded form, we add it to keyword lists. For example, "Cable News Network", which is expanded form of CNN, is added into keyword list for the question containing word 'CNN'. Fourth, a question type is determined by the pattern that we define. There are list of patterns in the table 2.1.

Table 2.1 Patterns for each question type

Question Type	Patterns
Person	Who, Who's, Whom~, ~man's name, ~woman's name
Location	Where ~, What + location (city, country,…..) ~ In what + location (city, country….) ~, What nationality~
Organization	What company~, What institution~
Time	When~, What time~, How many + Time (years, months, days)~
Currency	How much ~ spend, rent, cost, money, price~
Measure	How much, How many, How+ Adjective

If there is no matched pattern in the question, we estimate its question type using WordNet (Miller *et al*. 1991). We extract a noun phrase, which contains a head noun of the given question and estimate its question type based on synsets and a hypernyms of the head noun. Table 2.2 shows synsets and hypernyms lists for

each category. If the synsets and hyperyms of the head noun are not matched with a list in the table 2.2, we generate a noun phrase, which contains a head noun of question, as question type.

Table 2.2 Lists of WordNet synsets and hypernyms to assign questions to question types

Question type	Synsets and hypernyms
Person	Person
Location	district, territory, region
Organization	Commercial
Time	time period
Currency	cost, price
Measure	measure, magnitude

2.1.2 Named Entity Tagger

For the given POS tagged text, a named entity tagger generates named-entity tagged texts. It identifies six kinds of question type, which we define in the question analyzer step: Person = <PER>, location, region = <LOC>, organization = <ORG>, date or time expression = <TIME>, expression containing currency = <CUR>, and measures expression = <MEA>. For detecting question type of <PER>, <LOC>, and <ORG> - which are proper nouns, we use dictionaries for them. There are about 50,000 entries for person, 1,300 entries for location and 4,000 entries for organization. If we can not determine the type of a named entity, we tag it as <NPP>. And we use patterns and dictionaries for identifying question type of <TIME>, <CUR>, and <MEA>. For applying patterns, we extract phrase using regular expression – DT JJ* CD*. In the regular expression, DT, JJ and CD represent determiner, adjective, and cardinal number respectively. Table 2.3 shows patterns for <TIME>, <CUR>, and <MEA>.

Table 2.3 Patterns for “Time”, “Currency”, and “Measure” Named Entity

Question type (Named Entity)	Pattern
<TIME>	Four sequential digit e.g. 1942, Four sequential digit + ‘s’ Four sequential digit + punctuation mark, Four sequential digit + ‘s’ + punctuation mark mid- or late- , in + sequential
<CUR>	$+digit
<MEA>	digit +(m, km, cm), digit +(kg,g), digit+ l(liter)

2.1.3 Answer Extractor

Our answer extractor generates top-5 ranked phrases with two steps. First we extract three sentences for each document using keywords and question types. Second, sentences are partitioned into fixed length phrases – under 50 bytes and under 250 bytes. Third, the partitioned phrases are scored by keywords and question types.

Sentence Selection

In sentence selection step, top-3 sentences are selected for each document. Since, we believe that the context is very important for selecting relevant sentence, we consider the previous sentence, the current sentence and the next sentence for selecting relevant sentence. Each sentence is scored using a keyword and a question type of the given question. We use W_{sent} in the formula (3-1) for scoring sentences. It is based on the fact that how many keywords are matched and how many question types (named entity) are relevant to the question type of the given question in the current sentence, the previous sentence, and the next sentence. We believe that more keywords appear in a sentence and a context of the sentence and there are more relevant question types (named entities) to the question, the sentence has higher probability to contain a relevant answer to the question.

$$W_{key}(S_{Qij}) = \frac{matched \quad keywords \quad in \quad S_{Qij}}{\#of \quad keywords \quad in \quad Q_i}$$

$$W_{NE}(S_{Qij}) = \frac{\#of \quad NE \quad matched \quad with \quad Qtype_{Qi}}{\#of \quad NE \quad in \quad S_{Qij}}$$

$$W_{context}(S_{Q_{ij}}) = (W_{key}(S_{Qij}) + 0.5 \times W_{NE} \times (S_{Qij}))$$

$$W_{Sent}(S_{Qij}) = \alpha \times W_{context}(S_{Q_{ij-1}}) + \beta \times W_{context}(S_{Q_{ij}}) + \gamma \times W_{context}(S_{Q_{ij+1}}) \qquad (3-1)$$

where, S_{Qij} is jth sentence for the question i, and Qi is ith question.

Phrase Selection

In this step, we partition the extracted sentences into phrases with fixed length (under 50 bytes and under 250 bytes). Then each phrase is scored using keywords and question types. And top-5 ranked phrases are extracted as the relevant answer to the given question. We score phrases using W_{pass} in the formula (3-2). In the formula, we use nine window contexts for calculating scores of the phrases. It means that the contexts of the current phrase are considered – the previous four phrases and the next four phrases.

$$W_{key}(P_{Qij}) = \frac{matched \quad keywords \quad in \quad P_{Qij}}{\#of \quad keywords \quad in \quad Q_i}$$

$$W_{NE}(P_{Qij}) = \frac{\#of \quad NE \quad matched \quad with \quad Qtype_{Qi}}{\#of \quad NE \quad in \quad P_{Qij}}$$

$$W_{context}(P_{Qij}) = W_{key}(P_{Qij}) + 0.5 \times W_{NE}(P_{Qij})$$

$$W_{Pass}(P_{Qij}) = \theta_1 \times \sum_{P=P_{Qij-4}}^{P_{Qij-1}} W_{context}(P) + \theta_2 \times Wcontext(P_{Qij}) + \theta_3 \times \sum_{P=P_{Qij+1}}^{P_{Qij+4}} W_{context}(P) \qquad (3-2)$$

where, P_{Qij} is jth phrase for the question i, and Qi is ith question.

2.2 Results

2.2.1 Performance of Our QA system

We submitted one run in the 50byte category and one run in the 250byte category. The results are presented in Table 2.4. Table 2.4 breaks down the results by question type. Our system answers more correctly for the "Organization", "Currency", and "Person" question type than others. Not surprisingly, most of question types produce better result for the 250byte run than the 50byte run. However, for the question types, "Measure", "Time", and "Currency", there is not significant performance increase in the 250 bytes run result. We believe that the named entity tagger of our system can not identify the question types – "Measure", "Time", and "Currency" - very well and it makes difficult for answer extractor to identify relevant answers in the texts.

Table 2.4 Performance of our system for each question type. ARR means "Average Reciprocal Rank"

Question Type	# of Question	50 bytes run type			250 bytes run type		
		Correct #	Correct %	ARR	Correct #	Correct %	ARR
Person	147	52	35.37%	0.2355	**80**	**52.98%**	**0.3496**
Location	137	*41*	*29.93%*	*0.1915*	59	43.07%	0.3018
Organization	14	**7**	**50%**	**0.4071**	**8**	**57.14%**	**0.4524**
Time	77	25	32.47%	0.2353	*31*	*39.24%*	*0.2753*
Currency	6	**5**	**83.33%**	**0.4861**	**4**	**66.7%**	**0.45**
Measure	62	22	35.48%	0.2788	*19*	*30.16%*	*0.2193*
Other	239	*62*	*25.94%*	*0.1651*	**119**	**48.97%**	**0.3466**
Total	682	214	31.4%	0.212	320	46.9%	0.327

2.2.2 Error Analysis

We perform error analysis on the first 100 questions. It focuses on 250bytes run results. We divide errors into four types according to the component of our system where it causes errors. Table 2.5 shows errors and error types in the 250 bytes run results on the first 100 questions.

Table 2.5. Error analysis on the first 100 questions.

Error Type	# of Error (% of Error)
IR (Information retrieval) error	17 (34%)
QA (Query Analyzer) error	5 (10%)
NE (Named Entity Tagging) error	6 (12%)
AE (Answer Extraction) error	22 (44%)
Total	50

The first one is an IR type error. If there are no relevant answers in the retrieved documents, we determine the error as the IR error. For example, for "*Q222: What is Anubis?",* there is no relevant answer in the retrieved document. There are many errors with IR error type. We believe that since, the retrieved documents

are very small, – they are only 50 documents for each question –, and there are many question, which are very short, – there is only one content word such as "*Q236: Who is Coronado?*", and "*Q241: What is a caldera?*" –, IR system can not retrieve relevant documents very well.

The second one is a QA type error. If query analyzer mis-analyses the given question, we call it a QA error. For example, *"Q288: How fast can a Corvette go?"* is analyzed as "Other" by the question analyzer, although it should be "Measure" question type. It is caused by a POS tagger error – "fast" is tagged as a noun. Therefore, the question analyzer produces a wrong result, although there is the pattern, [How + adjective => "Measure" question type].

The third one is a NE type error. We treat errors as the NE error when named entity tagger can not detect the relevant named entity to the question type in the sentence or phrase where answer appears. We exclude the case that named entity tagger mis-analyses the named-entity boundaries or can not identify the precise named entity in the sentence or phrase where answer does not appear. It is because there are too many named entities to check them.

For example, for "*Q209: Who invented the paper clip?"*, the relevant answer is located in the following sentence.

> The paper clip, weighing <CUR>a desk-crushing 1,320 pounds,</CUR> is a faithful copy of **<NPP>Norwegian Johan Vaaler's</NPP>** <TIME>1899</TIME> invention, said Per <NPP>Langaker</NPP> of <NPP>the Norwegian School</NPP> of <NPP>Management.</NPP>

The question is analyzed as the "Person" question type. Therefore, answer will be "Person" named entity. However, "Norwegian Johan Vaaler", which can be relevant answer, is tagged as "<NPP>"- it means that the type of the named entity can not be determined.

There is another kind of NE type error. It is caused by roughly categorized question type.

For example, for "*Q245: Where can you find the Venus flytrap?*", the question can be treated as "Location" question type. And following sentence can be its answer.

> "**Whole savannas where flytraps were abundant** have been cleaned out," says <PER>Cecil Frost,</PER> coordinator of <PER>the North Carolina Plant Conservation Program.</PER>

Since, named entity tagger identifies "Location" named entities when it tagged as proper noun, the words, which contains meaning of location and is not proper noun, can not be detected as a "Location" named entity. Therefore, "savannas" is not tagged as the "Location" named entity and we can not extract it as answer.

The fourth one is an AE type error. When answer extractor can not identify the relevant answers, we define it as the AE error. Since, our answer extractor system extracts three sentences for each document, we can not extract the answer, which appear in the multi-sentence in the document. For example, for the question "*Q203: How much folic acid should an expectant mother get daily?*", following sentences can be relevant answer.

> Here are some good sources of folic acid according to <NPP>the USDA.</NPP>
> Raw forms of some vegetables are not included, because in their raw state they don't contain enough folic acid. For example, a 1/2-cup cooked serving of beets contains more than the same amount of the vegetable raw. <NPP>Also,</NPP> the term "good source" is based on **<NPP>the RDA</NPP> of <MEA>400</MEA> micrograms daily for a pregnant woman.**

In the sentence, which contains relevant answer, there is no words that matched with keyword analyzed by the question analyzer – *folic, acid, expectant, mother, get*. However, through four sentences, there are words matched with the keywords.

2.3. Discussion

Since our team did not have experience in the development of Question-Answering system before participating in the QA-track this year, our system is open to further improvement. Among the research issue for improving performance of our system, we will focus on following aspects:

- More detailed question type – we will divide each question type into detailed question type.
- Different weighting schemes for extracting sentence and phrase in the documents.
- Sorting criteria for the equally scored phrase and sentence.
- Sophisticated named entity tagger – using machine-learning technique.
- Coreference resolution
- Multi-sentential answer extraction

3. Cross-Language Information Retrieval track

There are four runs in our CLIR, two runs follow the dictionary and MI information based translation approach (KAIST9xlqm, KAIST9xlqt), another one using the mixture result of two commercial Machine Translation systems (KAIST9xlmt), and the final one is monolingual run (KAIST9xlch). We translated only query and description fields in all four runs.

We used SMART system (Salton, 1983) in our IR part after query translation. And because most of our resources are in GB code, we converted all BIG5 documents and the Chinese topics that given by TREC9 to GB. We used Universal Code Converter of shareware NJStar Communicator 2.0 (http://www.njstar.com/).

3.1 Translation Approach

3.1.1 Dictionary and MI information based query translation

The first two cross-lingual runs - KAIST9xlqm and KAIST9xlqt follow next steps:

1. Preprocessing.

   Paring the English topics and descriptions using the parser of Brill tagger (Brill, 1995), remain only noun and noun phrases.

2. Translation.

   Translate the remaining noun and noun phrases to Chinese using dictionary, do segmentation after translation. For example of CH55, we got "Word Trade Organization/f/世界贸易组织","membership/n/会员资格, 成员资格" from dictionary, after segmentation, the probable translations of "Word Trade Organization member ship" will be "Word Trade Organization membership : 世界 贸易 组织 会员 资格, 世界 贸易 组织 成员 资格".

   Proper noun translation was quite a problem in TREC9 query. In our translation, the proper noun recognition and translation followed next steps:

   1) If a capitalized word cannot be found in our bilingual dictionary, and it satisfies a Chinese Pinyin spelling, it will be regard as a Chinese proper noun (ex, Wan). If a Chinese proper noun is a Pinyin sequence contains more than two characters, it will be separated (ex, Daya → Da Ya), but after translation, them will be considered as one Chinese proper noun again. If a Chinese proper noun is followed by another Chinese proper noun, regard them as one word after translating (ex, "Da Ya"+"Wan" → "Da Ya Wan").

   2) Getting all probable Chinese characters of the Pinyin sequence by using Chinese Pinyin–character table, and select the Chinese character associations by using the character co-occurrence information that can be gotten from Chinese corpus. Chinese dictionary will be used in this step to delete the common words from the probable Chinese character associations.

      If the Chinese Proper noun contains only two characters, it will be a one-stop process: get all probable associations, delete the associations that can be found in Chinese dictionary – because they will be common words, and then get the occurrence times of the associations from Chinese corpus, remain the association that has the most frequent occurrence.

      If it contains more than three characters, we will get the occurrence time of first two character's first, delete common words from them, and remain only the associations that the occurrence times are ranking in top 5%. Then combine the remain associations to third probable characters, delete common words again, remain top 5% associations, and so on. In the final step, only the character association that owns the highest occurrence time will remain.

For example of “Da Ya Wan”, we can get 19 Chinese probable characters with pronunciation “Da”, 26 Chinese character with pronunciation “Ya”, and 29 character with “Wan”. In first step, get the occurrence times of all probable associations that pronounced “Da-Ya” (19*26 probable associations), delete the common words, remain the top 5% association by their occurrence times (ex, “打压，打牙，打亚，达雅，达亚，大丫，大压，大鸦，大亚，大呀，塔亚… ”). And in second step, get all of the Chinese character sequences of “Da-Ya-Wan” by using the remaining “Da-Ya” associations, get the occurrence times and delete the common words again, remain the best one.

In our experiments, we used the Peoples-Daily corpus of TREC5, because we have not finished our BIG5→GB converter about TREC9 documents when we do this work.

3. Word sense disambiguation.

   We used MI information of two nearby words in queries to do word sense disambiguation. The window is 5 words, and we got the MI information from the segmented Chinese documents supplied by TREC9. The window was 5 words.

   In one of our CLIR run KAIST9xlqm (maximum strategy), we try to select only the Chinese word association that own maximum MI value, if the MI values of given associations are all 0, remain only the first translation in each word. If there are the same MI values between two translation results, remain both of them.

   In KAIST9xlqt run (threshold strategy), we remain all of the Chinese word pairs that own MI values bigger than given threshold 0.

   For example of title CH57, “human right violation” (“human right/f/人权”, “violation/n/违犯，侵犯，妨碍，不敬”) will be translated as “人权 侵犯” in maximum strategy, and translated to “人权 侵犯，人权 妨碍” in threshold strategy. And in both strategies, “Chinese press”(“Chinese/n/中国 人,中国 话”, “press/n/新闻界，压力 机，压，按”) will be translated to “中国 人 按” (the correct translation is “中国 新闻界”).

4. Double the title field.

   Because the title field includes the most import words or phrases, to improve the IR performance, we double the title field. Our test result shows that this heuristic is quite helpful.

5. Using SMART system to do information retrieving.

3.1.2 Query translation using machine translation system

In our machine translation run KAIST9xlmt, we used two commercial systems by using the combination of the two machine translation results. As the above two runs, we translate the title and description fields of the

topics, do segmentation and POS tagging on the translation result, remain only content words (nouns, verbs, adverbs and adjectives), delete some stop words by using our stop word list (this stop word list includes the words that for the description field, like "报告 (report)", "报导 (report)", "文件 (document)"etc.). We can see next section that the result is not so good even after such effect.

3.1.3 Monolingual run

In monolingual run KAIST9xlch, we do nothing except do segmentation to the given Chinese titles and descriptions.

3.1.4 Resources and tools

Resources:

1) Chinese word dictionary with POS information (Yu, 1998), over 50,000 items. Using in Chinese document segmentation, Chinese topic segmentation in two CLIR runs KAIST9xlqm and KAIST9xlqt, and POS Tagging in machine translation run KAIST9xlmt and monolingual run KAIST9xlch.
2) English-Chinese bilingual dictionary with POS information, over 15,000 items. Using in English-Chinese word/phrase translation in two CLIR runs KAIST9xlqm and KAIST9xlqt.
3) Chinese Pinyin-Character table. Using in English-Chinese proper noun translation.
4) Chinese corpus: Peoples-Daily Newspaper that supplied in TREC5 and Chinese documents of TREC9.
5) All of the Chinese resources are in GB code, or converted to GB code from BIG5.

Tools:

1) English Parser of Brill tagger (Brill, 1995): Shareware. Used in our two CLIR run KAIST9xlqm and KAIST9xlqt.
2) SMART Information Retrieval System (Salton, 1983): in all four runs.
3) NJStar Communicator 2.0 (http://www.njstar.com): Shareware, can be download from web. Used as a BIG5→GB code converter.
4) Chinese segmentator and POS Tagger: A part of model of our Chinese-Korean machine translation system (Zhang & Choi, 1999).

3.2 Results

The following table shows the experiment results. We can see the comparison result in the individual queries part, and it based on the average precision over all relevant documents.

Table 3.1 The comparison of the precision.

Total performance				Individual performance				
Run	Avg. Prec.	R-Prec.	Avg. of Median	Best	Above	Median	Below	Worst
KAIST9xlqm	0.2231	0.2145	0.1460	1	10	4	10	0
KAIST9xlqt	0.2107	0.2095	0.1460	1	12	4	7	1
KAIST9xlmt	0.1378	0.1546	0.1460	0	11	1	11	2
KAIST9xlch	0.2233	0.2225	0.2522	4	4	0	16	1

The following is the recall-precision figure on CLIR run KAIST9xlqm (dictionary & corpus based query translation) and KAIST9xlmt (query translation by using machine translation system), we can compare it to the monolingual run KAIST9xlch.

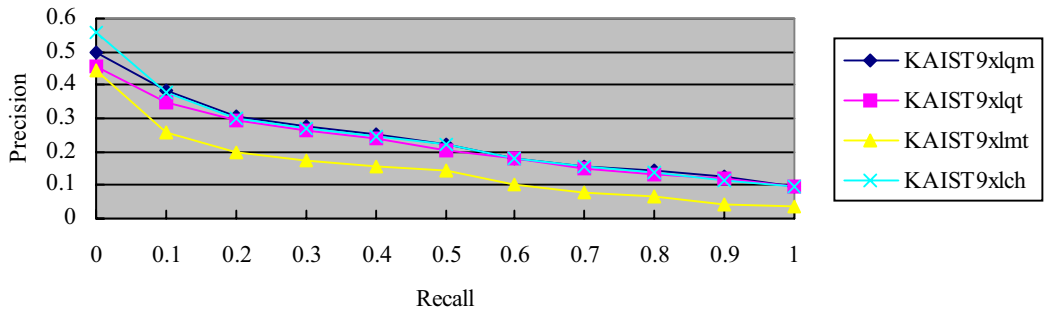

Fig. 3.1 Recall-Precision comparison

In Fig 3.1, we can see that the CLIR run KAIST9xlqm got even high precision than the monolingual run. We think there are several reasons, for example, we only use the noun and noun phrase in our KAIST9xlqm run, but remain all words (even stop words) in monolingual run KAIST9xlch; and we double the title field in our CLIR run KAIST9xlqm while we did not do so in monolingual run, when in the TREC9 topic, the titles reflect the queries very well, and contain only the important words.

The result of the machine translation run KAIST9xlch is not good enough comparing to our exception, especially when it compare to the other CLIR runs. We found that one of the machine translation system generates more noise words and failed to translate almost all of the proper nouns.

Although we pay much attention to the proper noun translation, but the experiment result of the queries that contain proper nouns are under medium yet. We think the reason is that, because the proper nouns not included in our Chinese dictionary, in the Chinese corpus they will be separated to independent characters, and it reflects the information retrieval result directly.

4. Batch Filtering track

We only submitted results for OHSU topics and MSH-SMP topics in batch filtering task. For OHSU topics, we have been exploring a filtering technique which combines query zone (Singhal, 1997), support vector machine (Vapnick, 1995), and Rocchio's algorithm (Rocchio, 1971). For MSH-SMP topics, we use support vector machine simply.

4.1 Experimental Procedure

4.1.1 Profile construction

User profile is created using modified Rocchio's formulation (Rocchio, 1971, Salton, 1990) for each OHSU topic.

$$\vec{P} = \alpha \cdot \vec{Q} + \beta \cdot \frac{1}{R} \sum_{D \in \mathrm{Re}l} \vec{D} - \gamma \cdot \frac{1}{N - R} \sum_{D \notin \mathrm{Re}l} \vec{D} \qquad (4.1)$$

, where $\vec{Q}$ and $\vec{D}$ denote the weighted term vector for query Q and document D, respectively. R = |Rel| is the number of relevant documents, and N is the total number of documents in the collection. The parameters were set to α=0, β=1, and γ=0.

The weights of terms are calculated by product of term frequency and inverse document frequency.

4.1.2 Filtering based on SVM using Query Zone

We reduced the number of negative training documents for learning of support vector machine using a variation of query zone.

Singhal et al. (Singhal, 1996) have proposed that only a selected set of non-relevant documents that have some relationship to a user's interest should be used in Rocchio's method. They proposed sampling of the non-relevant documents to form a query zone. We selected all documents with similarity to the profile greater than some threshold. If some relevant document does not pass the similarity threshold, it is included in the query zone.

Support vector machines are based on the Structural Risk Minimization principle (Vapnik, 1995) from computational learning theory. The method is defined over a vector space where the problem is to find a decision surface that maximizes the margin between the data points in a training set. We test SVM using the SVM^{light} system (Joachims, 1998) which is an implementation of Vapnik's Support Vector Machine (Vapnik, 1995) for the problem of pattern recognition.

4.1.3 Re-filtering using profile-document similarity

We re-filtered the results from SVM classifier by profile-document similarity. The in-class threshold and out-class threshold are used for re-filtering.

If a profile-document similarity is above in-class threshold, re-filter decide the document to be relevant for user's interest without respect of the result of SVM filter. And, if a profile-document similarity is below out-class threshold, re-filter decide the document to be non-relevant.

4.2 Results

There are three sets of topics: OHSU topics, MSH topics, and MSH-SMP topics. We submitted two runs (KAISTbfo1, KAISTbfo2) for the OHSU topics and one run (KAISTbfms) for MSH-SMP topics.

The TREC-9 filtering track use the OHSUMED collection of documents from MEDLINE. In batch filtering, the 1987 OHSUMED documents are used for building the filtering profiles. The 1988-91 OHSUMED documents form the test set. We didn't use the M. field in the documents for the OHSU topics and MSH-SMP topics.

We tested RBF (radial basis function) models offered by SVM^{light} system. SV learning is based on non-relevant documents from query zone and all relevant documents for each topic. The threshold for query zone was set to 0.1. The number of feature is 31,042. For KAISTbfo1, in-class threshold was set to 0.6 and out-class threshold was set to 0.2. For KAISTbfo2, the thresholds are 0.6 and 0.3. The result of KAISTbfo1 differ little from KAISTbfo2.

Table 4.1 shows the results of OHSU topics and MESH-SAMPLE topics.

Table 4.1 TREC-9 Batch Filtering Results

Topic set / Measure	OHSU		MESH-SAMPLE
	KAIST9bfo1	KAIST9bfo2	KAIST9bfms
Total retrieved	1615	1437	62483
Relevant retrieved	794	746	35146
Macro average recall	0.227	0.204	0.245
Macro average precision	0.421	0.485	0.543
Mean T9P	0.200	0.194	0.419
Mean utility	12.175	12.714	85.910
Mean T9U	12.175	12.714	86.424
Mean scaled utility	0.061	0.078	0.153
Zero returns	0	2	0

We expected that combined method using QZ, SVM, and Rocchio's algorithm might perform much better than SVM. However, the result of combined method differ little from SVM. A more in-depth analysis is needed to understand these results.

References

Brill, E. (1995) *Transformation-Based error-driven learning and natural language processing: a case study in part of speech tagging*. Computational Linguistics.

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the European Conference on Machine Learning (ECML)*.

Miller G.A. & Beckwith R.& Fellbaum C. & Gross D. & Miller K. (1991). Five Papers on WordNet" *International Journal of Lexicography.*

Rocchio, J.J. (1971). Relevance feedback in information retrieval. *In THe SMART Retrieval System - Experiments in Automatic Document Processing*, (pp. 313-323). Prentice Hall, Inc.

Salton, G. & McGill, M.J. (1983*). Introduction to Modern Information Retrieval*, McGraw-Hill, Inc.

Salton, Gerard & Buckley, Chris. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288-297.

Singhal, Amit & Mitra, & Mandar & Buckley, Chris. 1996. Learning routing queries in a query zone. In *Proceedings of the Twentieth ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 21-29).

Vapnick, Vladimir N. (1995). *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.

Yu Shi-Wen, Xue-Feng Zhu, Hui Wang, Yun-Yun Zhang (1998). *The Grammatical Knowledge-base of Contemporary Chinese – A Complete Specification*. The Press of Tsinghua University.

Zhang, Min & Choi, Key-Sun (1999). *Pipelined Multi-Engine Machine Translation : Accomplishment of MATES/CK System*, in Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation.