On a Combination of Probabilistic and Boolean IR Models for GeoTime Task

Masaharu Yoshioka
Graduate School of Information Science and Technology, Hokkaido University
N14 W9, Kita-ku, Sapporo-shi
Hokkaido Japan
yoshioka@ist.hokudai.ac.jp

ABSTRACT

NTCIR-GeoTime task is a task to search documents with Geographic and Temporal constraints and almost all topic can be regarded as question and answering (QA) for particular named entities. To make a good information retrieval (IR) system for QA for particular named entities, it is better to use Boolean IR model by using appropriate Boolean query with named entity information. In this paper, we propose to use ABRIR (Appropriate Boolean query Reformulation for Information Retrieval) for this problem. In this system, appropriate list of synonyms and variation of Japanese katakana description of given query are used for constructing Boolean query. Evaluation results shows that ABRIR works effectively for the task of IR for QA.

Categories and Subject Descriptors

H3.3 [**Information Systems**]: Information Search and Retrieval

General Terms

Information Retrieval

Keywords

Probabilistic IR model, Boolean IR model, Query formation, Question and Answering

1. INTRODUCTION

Focus of the NTCIR-GeoTime task is on search with Geographic and Temporal constraints[1]. Since detailed analysis on these constraints requires highly computational costs, it is better to have an information retrieval system that can retrieve good initial candidate documents for further analysis.

One of the significant differences between the document retrieval in general and ones for question and answering for particular named entities is that documents that do not contain any information about given named entities must be irrelevant. Therefore, it is better to use Boolean IR model. However, due to the variation of the description about named entities and synonyms of other related terms, it is not so easy to make appropriate Boolean query at the initial retrieval stage.

ABRIR (Appropriate Boolean query Reformulation for Information Retrieval)[6] is an IR system that combines probabilistic and Boolean IR models for handling this type of problem. This system constructs appropriate Boolean query based on the comparison between initial query and pseudo relevance documents and calculates penalty for retrieved documents that do not satisfy the Boolean query.

In this paper, we briefly review ABRIR and discuss how to modify ABRIR for Web documents into one for the task. Experimental results shows that our approach is better than the system with probabilistic IR model only.

2. ABRIR(APPROPRIATE BOOLEAN QUERY REFORMULATION FOR INFORMATION RETRIEVAL)

ABRIR is an IR system that have following features for combination of probabilistic and Boolean IR model.

1. Reformulation of a Boolean query
   The system compares initial Boolean query and pseudo-relevant documents and modify it that satisfies most of these documents.

2. Calculate score based on the results of probabilistic and IR model
   Basic documents scores are calculated by using probabilistic IR model. Penalty is applied for score of documents that do not satisfies given Boolean query.

2.1 Reformulation of a Boolean query

The following procedure is used to reformulate a Boolean query. Figure 1 shows an example of this process.

1. Selection of Boolean candidate words
   We select all terms used in the original query that also exist in all relevant documents. We reformulate a Boolean query by using the selected words with the AND operator. In this example, “A” and “C” exist in all relevant documents, so “A and C” is selected as a candidate query.

2. Reformulation of the Boolean query based on the initial query
   When we have created an original Boolean query, we relax it. When there are one or more words in the

initial query that are used within an OR operator, we expand the generated query by using this OR operator information. In this example, because “C or D” exists in the original query, we modify the generated query to “A and (C or D).”

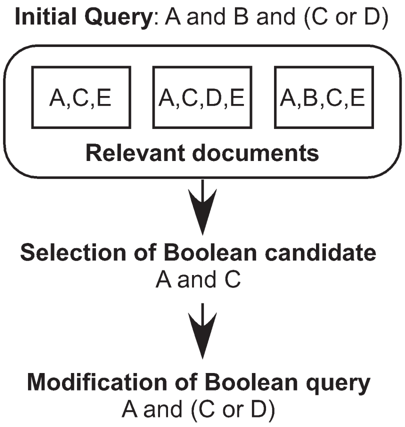
Initial Query: A and B and (C or D)
A,C,E
A,C,D,E
A,B,C,E
Relevant documents
Selection of Boolean candidate
A and C
Modification of Boolean query
A and (C or D)

Figure 1: Boolean Query Construction [6]

2.2 Modification of the Score Based on the Boolean Query

Probabilistic IR model of ABRIR is almost equivalent to Okapi BM25 with pseudo-relevance feedback and query expansion and implemented by using the Generic Engine for Transposable Association (GETA) tool [1].

Probabilistic IR model for ABRIR used the BM25 weighting formula to calculate the score of each document:

$$\sum_{T \in Q} w^{(1)} \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} \qquad (1)$$

$w^{(1)}$ is the weight of a (phrasal) term T, which is a term or a phrasal term in query Q, and is calculated using Robertson-Sparck Jones weights:

$$w^{(1)} = log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \qquad (2)$$

where N is the count of all documents in the database, n is the count of all documents containing T, R is the given number of relevant documents, and r is the count of all relevant documents containing T. In addition, tf and qtf are the number of occurrences of T in a document and in a query, respectively, and k_1, k_3 and K are control parameters.

For handling phrasal terms, we introduced a parameter $c(0 \leq c \leq 1)$ that is used for counting the phrasal terms in a query, where qtf is incremented by c rather than one when a phrasal term is found.

For the query expansion, we used Rocchio-type feedback [5]:

$$qtf = \alpha qtf_0 + (1-\alpha) \frac{\sum_{i=1}^{R} qtf_i}{R} \qquad (3)$$

where qtf_0 and qtf_i are the number of times T appears in the query and in relevant document i, respectively.

This system uses following procedures to extract word and phrase indexes from the text.

1. Morphological analysis
   We converted ASCII text characters into two-byte EUC codes by using KAKASI [2] as a code converter, and ChaSen [4] as a morphological analyzer.

2. Extraction of index terms
   We extracted noun words (nouns, unknowns, and symbols) as index terms. We excluded numbers, prefixes, postfixes, and pronouns from the index terms. We removed “ー” from the end of a term when the length of the term was longer than two katakana characters. All alphabets were then normalized to one-byte ASCII codes and stored in lower case.

3. Extraction of phrasal terms
   We aimed to use compound nouns as phrasal terms, so we extracted phrasal terms from pairs of adjacent noun terms. We also used prefixes, postfixes, and numbers for extracting phrasal terms.

ABRIR used the five top-ranked documents for pseudo-relevance feedback and selected the 300 different terms with the highest mutual information content between a relevant document set and a term.

Because we assume that documents that do not satisfy the Boolean query may be less appropriate than documents that do satisfy the query, we subtract a penalty score from documents that do not satisfy the Boolean query.

We apply the penalty based on the importance of the word. For a probabilistic IR model, we used the BM25 weighting formula to calculate the score of each document (Equation 1). In this equation, $w^{(1)} \frac{(k_3+1)qtf}{k_3+qtf}$ shows the importance of the word in the query. We use a control parameter β to calculate the penalty score.

$$Penalty(T) = \beta * w^{(1)} \frac{(k_3+1)qtf}{k_3+qtf} \qquad (4)$$

For the OR operator, we use the highest penalty from all the OR terms as the overall penalty.

We now describe how to calculate the penalty, using the Boolean query (“A” and (“C” or “D”)) discussed in Figure 1 as an example. First, we calculate the penalty score for all words (“A,” “C,” and “D”). We assume $Penalty(C) \geq Penalty(D)$ in this case. Documents not possessing terms “A,” “C,” or “D” receive the penalty $Penalty(A)+Penalty(C)$. Documents possessing only the “C” term receive $Penalty(A)$.

3. ABRIR FOR GEOTIME

3.1 Difference between WWW documents retrieval and GeoTime retrieval

ABRIR, discussed in previous section, was developed for WWW documents retrieval. Since the characteristics of the document retrieval in WWW documents and ones for question and answering for particular named entities is different, it is necessary to modify some parameters for adopting this problem.

Followings are significant difference to consider.

[1] http://geta.ex.nii.ac.jp/

[2] http://kakasi.namazu.org/

1. Usage of verb as index terms
   It is necessary to include verbs as index terms for handling queries with verbs. In addition, since verbs have varieties of synonyms, it is better to have a mechanism to deal with synonyms.

2. Handling named entities
   Since keywords about named entities are important for this type of query, it is better to identify named entity information. In addition, since there are varieties of named entity representation exist especially for Japanese Katakana named entity (mostly from the named entity of foreign countries), it is better to have a mechanism to deal with such variations.

3. Number of relevant documents
   Since there are not so many articles reporting same events, it is better to modify the size of pseudo-relevant documents.

4. Number of query expansion terms
   For question and answering, precision is more important than recall, it is better to reduce the number of query expansion terms.

3.2 Query Construction by using Synonyms and Variation

In order to make a good Boolean query, it is better to have appropriate list of synonyms and variation of Japanese katakana description.

For the verbs, EDR electronic dictionary, developed by Japan Electronic Dictionary Research Institute, Ltd. [2] is used for finding out synonyms. In this dictionary, each verb has one or more semantic id(s). All verbs that shares one or more semantic id(s) with the original verb are candidates of the synonyms.

For the named entities written in Japanese katakana, following rules are used for generating varieties of description.

1. Remove “ー” from the original keyword

2. Remove small katakana (e.g., “ァィゥェォャュョヮヵヶッ”) from the original keyword

3. Replace small katakana (e.g., “ァィゥェォャュョヮヵヶッ”) to large katakana (e.g., “アイウエオヤユヨワカケツ”)

By applying this generation rule to the keyword “ヘップバーン” (Hepburn), three candidates (“ヘップバン”, “ヘプバーン” “ヘツプバーン”) are generated.

Figure 2 shows procedures for constructing query and retrieval in ABRIR.

1. Remove question part of the query
   Question part of the query (e.g., “のはいつですか？” (when)) is trimmed from the original query.

2. Morphological analysis and NE tagging
   Almost same index terms extraction system is used for extract initial keywords. There are two difference in this extraction process.

   - Extraction of verb
   - Identification of named entities
     Cabocha[3] is used for identify named entities.

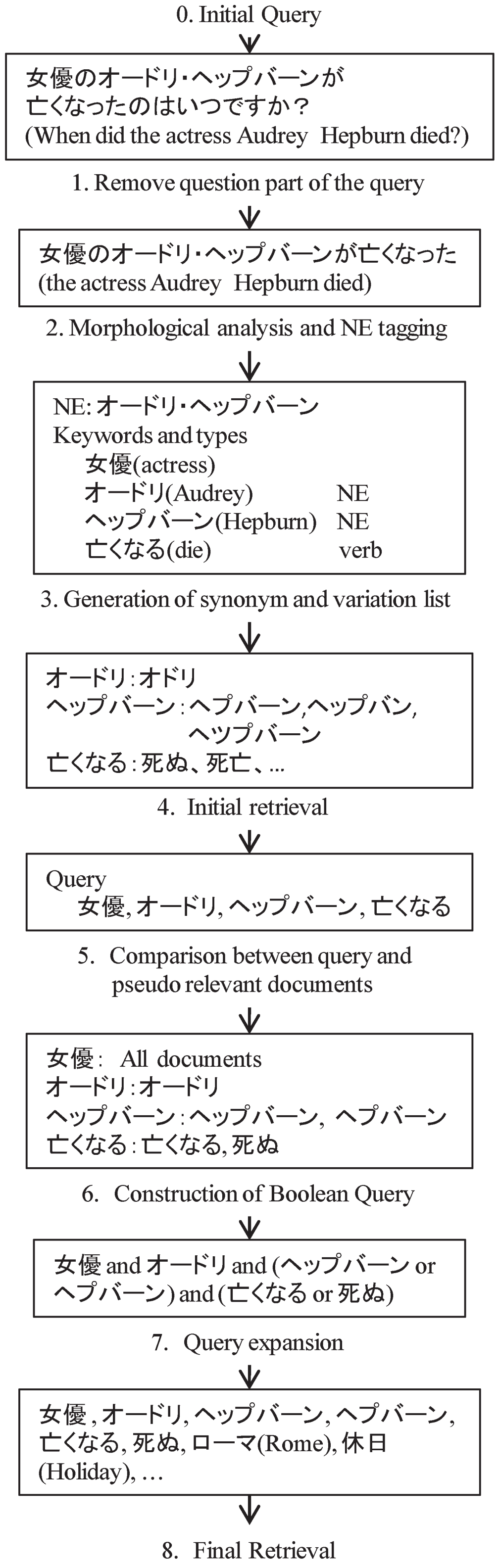
0. Initial Query
女優のオードリ・ヘップバーンが
亡くなったのはいつですか？
(When did the actress Audrey Hepburn died?)
1. Remove question part of the query
女優のオードリ・ヘップバーンが亡くなった
(the actress Audrey Hepburn died)
2. Morphological analysis and NE tagging
NE: オードリ・ヘップバーン
Keywords and types
女優(actress)
オードリ(Audrey) NE
ヘップバーン(Hepburn) NE
亡くなる(die) verb
3. Generation of synonym and variation list
オードリ:オドリ
ヘップバーン:ヘプバーン,ヘップバン,
ヘツプバーン
亡くなる:死ぬ、死亡、...
4. Initial retrieval
Query
女優, オードリ, ヘップバーン, 亡くなる
5. Comparison between query and
pseudo relevant documents
女優: All documents
オードリ:オードリ
ヘップバーン:ヘップバーン, ヘプバーン
亡くなる:亡くなる, 死ぬ
6. Construction of Boolean Query
女優 and オードリ and (ヘップバーン or
ヘプバーン) and (亡くなる or 死ぬ)
7. Query expansion
女優 , オードリ, ヘップバーン, ヘプバーン,
亡くなる, 死ぬ, ローマ(Rome), 休日
(Holiday), …
8. Final Retrieval

Figure 2: Procedures for Constructing Query and Retrieval in ABRIR.

3. Generation of synonym and variation list
   The system generates synonym list for verbs and variation list for named entity.

4. Initial retrieval
   Probabilistic IR model is used for finding out pseudo relevant documents. Based on the discussion of section 3.1, we only use top 3 ranked documents as pseudo relevant ones.

5. Construction of Boolean query
   There are three types of keywords in query; e.g., NE, verb, and other. The system compares query keywords and pseudo relevant documents in following manner.

   - Named entity

     Since the system generates variation list of given NE automatically, most of the keywords are meaningless. Therefore the system compares variational description list and keywords in the documents and remove keywords that do not exist in the documents. For example, when there is two documents that contain “ヘップバーン” and one document that contains “ヘプバーン”, the system constructs OR description (“ヘップバーン” or “ヘプバーン”) for “ヘップバーン”.

   - Verb

     When all pseudo relevant documents contains one or more synonyms of the verb, these documents are sufficient enough for generating synonym list for final Boolean query. In this case, synonyms that exist in the documents are used for Boolean query. For example, when there is two documents that contain “亡くなる” (die) and one document that contains “死ぬ” (die), AND elements are modified as (“亡くなる” or “死ぬ”).

     When there is one or more document(s) that do not contain any synonyms, the system generates new query by replacing the verb with synonym list and conducts secondary retrieval. By using new three pseudo relevant documents, the system selects synonyms that exist in the documents are used for Boolean query.

   - Other keywords in initial query

     When other keywords in initial query exist in all pseudo relevant documents, These keywords are used as AND elements of the final query.

6. Construction of Boolean query
   A set of synonyms, named entity variation lists, and keywords in all pseudo relevant documents are joined by AND operator for constructing Boolean query.

7. Query expansion by using pseudo relevant documents
   The system selected the 5 different terms with the highest mutual information content between a relevant document set and a term. The system also add keywords in Boolean query as expansion terms.

8. Final retrieval
   Based on the final query, final retrieval is conducted by using probabilistic IR model. We apply the penalty based on the importance of the word by using equation 4. In this formalization, we assume the Boolean query element for named entity is more important than others, we set higher value to the β_n instead of β for named entity.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Experimental Set Up

Followings are parameters for the submitted results. Most of the parameters are common ones for WWW retrieval. We use $k_1 = 1, k_3 = 7, K = \frac{dl}{avdl}, c = 0.3, \alpha = 0.7$ for probabilistic IR model. Here, dl is the length of a document (the number of terms and phrasal terms) and $avdl$ is the average length of all documents.

We also use $\beta = 3, \beta_n = 1000000$ for penalty calculation. By using this formalization, there are many documents with minus scores. Therefore we just recalculate the score values that retains the order of all document scores.

Followings are description of the submitted runs.

HU-KB-JA-JA-01-D, HU-KB-JA-JA-02-DN Boolean operators on named entities are used for filtering the results instead of penalty calculation. Boolean operators on verbs are used for penalty calculation. “-D” uses description only and “-DN” uses description and narrative.

HU-KB-JA-JA-03-D All Boolean operators are used for penalty calculation.

HU-KB-JA-JA-04-D Boolean operators on Named Entity are used for penalty calculation. Boolean operators on verb list are not used.

HU-KB-JA-JA-05-D No Boolean operators are used. This system is equivalent to the baseline Okapi BM25 system.

4.2 Discussion about Experimental Results

Table 1 shows the evaluation measure for each submitted run.

Table 1: Evaluation measure for each submitted run

	01-D	02-DN	03-D	04-D	05-D
AP	0.3697	0.3867	0.3719	0.3627	0.2881
nDCG	0.4117	0.4268	0.4162	0.4078	0.3282
Q	0.5710	0.5685	0.5881	0.5717	0.4993

From the comparison between 01-D and 03-D, we can discuss the effectiveness of the Boolean query for the filter. For the 14 topics (4,5,7,8,10,11,12,15,16,18,21,22,23,24), the system can not make appropriate Boolean query and the result of 01-D is same as that of 03-D. For the 7 topics (Topic:Boolean_matched_documents; 1:772, 3:1, 6:275, 9:6, 13:105, 19:329, 20:945), the system can make appropriate Boolean query and it retrieves all relevant documents with smaller number of documents. For other 3 topics (Topic:filter_out/total_rel 2:26/48, 14: 2/2, 25: 1/3), constructed Boolean query is not appropriate enough and some relevant documents are filtered out. Due to this problem, the system performance of 01-D is worse than 03-D.

Since 03-D is better than 01-D, we use comparison between 03-D and 05-D (base line) for analyzing the effectiveness about the usage of Boolean query. The t-test and Wilcoxon Signed Rank test was used to compare Average Precision (AP), normalized Discounted Cumulative Gain (nDCG), and Q. From a result of the t-Test with significance level of 0.05 for a two-sided tests, difference about nDCG(0.018) and Q(0.040) are statistically significant and one about AP(0.055) is not significant. From a result of Wilcoxon Signed Rank test with significance level of 0.01 for a two-sided tests, AP(0.0015), nDCG(0.0006) and Q(0.0024) are statistically significant.

There are 3 topics (2:AP,nDCG,Q 11:AP,nDCG,Q, 21:AP) where the result of 03-D is worse than 05-D.

For the topic 2, "ハリケーン"(hurricane) is recognized as named entity and articles about "ハリケーン" (hurricane) without "カトリーナ" (Katrina) get similar score "カトリーナ" (Katrina) without "ハリケーン" (hurricane).

For the topic 11 and 21, those topics don't contain the named entity information. In such a case, it is difficult to assure the quality of generated query.

The topic 14 is also a difficult topic for our system. The topic 14 includes named entity keyword "アフリカ" (Africa). However, the relevant documents has name of the African country "コンゴ民主共和国" (Democratic Republic of the Congo) instead of "アフリカ". In order to deal with such relation, it is necessary to have a good query analyzer and mechanism to deal with part whole relationship to generate related keyword list for Boolean query.

5. CONCLUSION

In this paper, we propose to use ABRIR as an IR system for question and answering for particular named entities. From the evaluation experiment, we confirm that ABRIR can make appropriate Boolean query and penalty based system outperform the baseline system (probabilistic IR model: Okapi BM25).

6. REFERENCES

[1] F. Gey, R. Larson, N. Kando, J. Machado-Fisher, and T. Sakai. NTCIR-GeoTime overview: Evaluating geographic and temporal search. In *Proceedings of the Eighth NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, 2010. (to appear).

[2] Japan Electronic Dictionary Research Institute, Ltd. (EDR). *EDR ELECTRONIC DICTIONARY VERSION 2.0 TECHNICAL GUIDE TR2-007*, 1998.

[3] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.

[4] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, K. Takaoka, and M. Asahara. *Morphological Analysis System ChaSen version 2.2.1 Manual.* Nara Institute of Science and Technology, 2000.

[5] M. Uchiyama and H. Isahara. Implementation of an IR package. In *IPSJ SIGNotes, 2001-FI-63*, pages 57–64, 2001. (in Japanese).

[6] M. Yoshioka and M. Haraguchi. On a combination of probabilistic and boolean ir models for www document retrieval. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4:340–356, 2005.