# On a Combination of Probabilistic and Boolean IR Models for Question Answering

Masaharu Yoshioka

Graduate School of Information Science and Technology, Hokkaido University
N-14 W-9, Kita-ku, Sapporo 060-0814, Japan
yoshioka@ist.hokudai.ac.jp

**Abstract.** To make a good question answering (QA) system based on a text database, it is preferable to have a good information retrieval (IR) system that can find appropriately relevant document sets for a given query. To make a good IR system for QA about particular named entities (NEs), it is preferable to use a Boolean IR model that uses appropriate Boolean queries with the NE information. In this paper, we propose to use appropriate Boolean query reformulation for information retrieval (ABRIR) for this problem. In this system, an appropriate list of synonyms and variations of Japanese katakana descriptions of a given query are used to construct the Boolean query. Evaluation results show that ABRIR works effectively for the IR task in QA.

**Keywords:** Information Retrieval, Boolean IR model, Probabilistic IR model, Question Answering.

## 1 Introduction

QA is a task concerned with finding answers to natural language (NL) questions (e.g., "How tall is the Tokyo Tower?" and "Who is George Bush?") from large text collections. To construct a good QA system, it is preferable to have an appropriate IR system for retrieving documents that have descriptions relevant to providing answers. IR for QA (IR4QA) is a task that evaluates IR modules from the viewpoint of QA system construction [1–3].

There are several approaches to IR4QA, but most of the systems discussed use partial-match IR models, such as the probabilistic IR model, the language model, and the vector-space model [1–3].

In contrast, we assume that one of the significant differences between document retrieval in general and QA about particular NEs is that documents that do not contain any information about the given NEs must be irrelevant. Therefore, it is preferable to use a Boolean IR model. However, because of variations in the description of NEs and synonyms of other related terms, it is not straightforward to make an appropriate Boolean query at the initial retrieval stage.

ABRIR [4] is an IR system that combines probabilistic and Boolean IR models for handling this type of problem. The system constructs an appropriate

P.-J. Cheng et al. (Eds.): AIRS 2010, LNCS 6458, pp. 588–598, 2010.
© Springer-Verlag Berlin Heidelberg 2010

Boolean query based on the comparison between the initial query and pseudo-relevant documents. It then calculates a penalty for retrieved documents that do not satisfy the Boolean query.

In this paper, we briefly review ABRIR and then discuss how to adapt ABRIR for Web documents to become suited to the QA task. Experimental results show that our approach is better than one that uses a probabilistic IR system model alone.

## 2 The ABRIR Approach

ABRIR is an IR system that has the following combination of features from the probabilistic and Boolean IR models.

1. Reformulation of a Boolean query
   The system compares an initial Boolean query and pseudo-relevant documents and modifies the query, aiming to satisfy most of these documents.
2. Calculating a score based on the results of the probabilistic and IR model
   Basic document scores are calculated by using the probabilistic IR model. A penalty is used in scoring documents that do not satisfy the given Boolean query.

### 2.1 Reformulation of the 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 the 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 with 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 extend the generated query to "A and (C or D)."

### 2.2 Modification of the Score Based on the Boolean Query

The probabilistic IR model in ABRIR is almost equivalent to Okapi BM25 [5] with pseudo-relevance feedback and query expansion, and is implemented by using the generic engine for transposable association (GETA) tool [1].

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

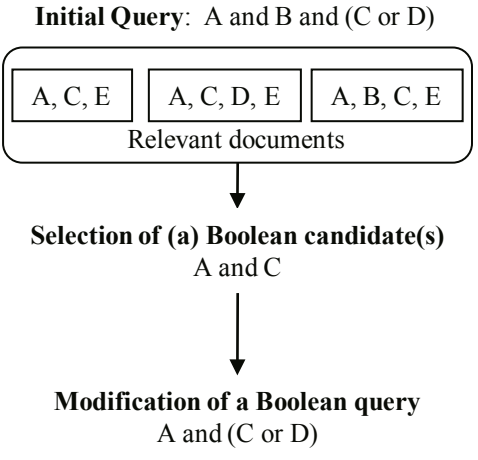


**Fig. 1.** Boolean query construction. [4]

The probabilistic IR model in ABRIR used the BM25 weighting formula to calculate the score for each document:

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

Here, $w^{(1)}$ 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 [5]:

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

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 $c$ $(0 \leq c \leq 1)$, which is used to count the phrasal terms in a query, such that $qtf$ is incremented by $c$ rather than 1 when a phrasal term is found.

For the query expansion, we used Rocchio-type feedback [6]:

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

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

The system used the following procedures to extract word and phrase indexes from the text.

1. Morphological analysis
   We converted ASCII text characters into two-byte extended Unix codes (EUC) by using KAKASI [2] as a code converter, and ChaSen [7] 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 "—" 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 format.
3. Extraction of phrasal terms
   Aiming to use compound nouns as phrasal terms, we extracted phrasal terms from pairs of adjacent noun terms. We also used prefixes, postfixes, and numbers in extracting phrasal terms.

ABRIR used the five top-ranked documents for pseudo-relevance feedback and selected 300 different terms having 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 give a penalty score to documents that do not satisfy the Boolean query.

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

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

For the OR operator, we used the highest penalty among the OR terms as the overall penalty.

We now describe how to calculate the penalty, using the example Boolean query ("A" and ("C" or "D")) shown in Figure 1. First, we calculate the penalty score for all individual words ("A," "C," and "D"). We assume $Penalty(C) \geq Penalty(D)$ in this case. Documents possessing none of "A," "C," or "D" receive the penalty $Penalty(A)+Penalty(C)$. Documents possessing only the "C" term receive $Penalty(A)$.

## 3 ABRIR for QA

### 3.1 Differences between WWW Document Retrieval and QA Retrieval

ABRIR, as discussed in the previous section, was developed for WWW document retrieval. Because the characteristics of document retrieval in WWW documents

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

and those of QA for particular NEs are different, it is necessary to modify some parameters when applying ABRIR to QA.

The following significant differences should be considered.

1. Use of verbs as index terms
   It is necessary to include verbs as index terms for handling queries containing verbs. In addition, because verbs have various synonyms, it is preferable to have a mechanism for dealing with synonyms.
2. Handling NEs
   Because keywords about NEs are important for this type of query, it is preferable to identify the NE information. In addition, because various NE representations exist, particularly for Japanese katakana NEs (mostly derived from foreign NEs), it is preferable to have a mechanism for dealing with such variations.
3. Number of relevant documents
   Because there are relatively few articles reporting the same events, it is preferable to modify the number of pseudo-relevant documents.
4. Number of query expansion terms
   For QA, precision is more important than recall, and it is therefore preferable to reduce the number of query expansion terms.

### 3.2 Query Construction Using Synonyms and Variation Lists

To make a good Boolean query, it is preferable to have an appropriate list of synonyms and variations of Japanese katakana descriptions.

For the verbs, the EDR electronic dictionary, developed by Japan Electronic Dictionary Research (EDR) Institute, Ltd., [8] is used for finding synonyms. In this dictionary, each verb has one or more semantic ids. All verbs that share a semantic id with the original verb are candidate synonyms.

For NEs written in Japanese katakana, the following rules are used for generating varieties of description[3].

1. Remove “ー” from the original keyword.
2. Remove small katakana (e.g., “アイウエオヤユヨワカケツ”) from the original keyword.
3. Replace small katakana (e.g., “アイウエオヤユヨワカケツ”) by 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”, and “ヘツプバーン”) are generated.

Figure 2 shows the procedures for query construction and retrieval in ABRIR, which can be described as follows.

[3] Our NE Variation generation rule is simple. It is better to use more sophisticated method [9, 10] for our future works

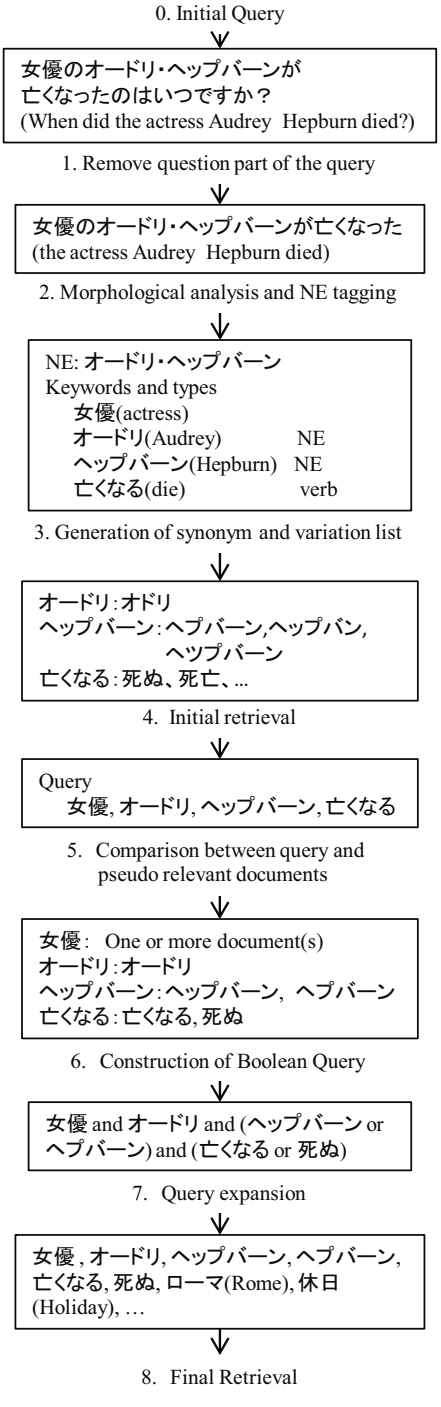


**Fig. 2.** Procedures for query construction and retrieval in ABRIR

1. Remove the question element from the query
   The 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 the same index-term-extraction system is used to extract initial keywords, except for two differences:
   - extraction of verbs, and
   - identification of NEs, where Cabocha [11] is used to identify NEs.
3. Generation of the synonym and variation lists
   The system generates a synonym list for verbs and a variation list for NEs.
4. Initial retrieval
   The probabilistic IR model is used to find pseudo-relevant documents. Based on the discussion in section 3.1, we use only the top-three-ranked documents for this purpose.
5. Construction of a Boolean query
   There are three types of keyword in the query, namely, NEs, verbs, and other keywords. The system compares query keywords and pseudo-relevant documents in the following manner.
   - NEs.
     Because the system generates a variation list for a given NE automatically, most of the keywords are meaningless. Therefore, the system compares the variation list and keywords in the documents, removing keywords that do not exist in the documents. For example, when there are two documents containing “ヘップバーン” and one document containing “ヘプバーン”, the system constructs an OR description (“ヘップバーン” or “ヘプバーン”) for “ヘップバーン”.
   - Verbs.
     When all pseudo-relevant documents contain one or more synonyms of the verb, these documents are sufficient to generate a synonym list for the final Boolean query. In this case, synonyms that exist in the documents are used for the Boolean query. For example, when there are two documents that contain “亡くなる” (die) and one document that contains “死ぬ” (die), the AND elements are modified to give (“亡くなる” or “死ぬ”).
     When at least one document does not contain any synonyms, the system generates a new query by replacing the verb by the synonym list and conducting a secondary retrieval. By using three new pseudo-relevant documents, the system selects synonyms that exist in the documents for the Boolean query.
   - Other keywords.
     When other keywords in the initial query exist in one or more pseudo-relevant documents, these keywords are used as AND elements of the final query.
6. Construction of the Boolean query
   The set of synonyms, NE variation lists, and keywords from all pseudo-relevant documents are joined by the AND operator to construct the Boolean query.

7. Query expansion using pseudo-relevant documents
   The system selects the five different terms with the highest mutual-information content between a relevant document set and a term. The system also adds keywords in the Boolean query as expansion terms.
8. Final retrieval
   Based on the final query, final retrieval is conducted using the probabilistic IR model. We apply a penalty based on the importance of the word by using equation 4. In this formalization, we assume that a Boolean query element for NE is more important than the others, and we therefore give a higher value to $\beta_n$ than to $\beta$ for NE.

## 4 Experimental Results and Discussion

### 4.1 Experimental Setup

The NTCIR-8 GeoTime Japanese monolingual task data [12] was used to evaluate the proposed system. There are 24 QA topics about geographic and temporal information in Japanese for Mainichi newspapers over the period 2002-2005, which comprises 377,941 documents. Submitted results are evaluated based on the view point of document based relevance judgement. They uses the same techniques used for analyzing IR4QA runs [3].

The parameter values used for the experiments were as follows. Most of the values are common in WWW retrieval. We used $k_1 = 1, k_3 = 7, K = \frac{dl}{avdl}, c = 0.3, and \alpha = 0.7$ for the 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.

In addition, we used $\beta = 3, and \beta_n = 1000000$ for the penalty calculations. Using this formalization, many documents had minus scores. Therefore, we simply recalculated the score values to retain the ordering of all document scores.

These are descriptions of the versions of the system tested for comparison:

**NE-filter-Verb-penalty (NfVp).** Boolean operators on NEs are used for filtering the results, instead of using penalty calculations. Boolean operators on verbs are used for penalty calculations.

**NE-penalty-Verb-penalty (NpVp).** Boolean operators only are used for penalty calculations.

**Baseline-Okapi (BO).** No Boolean operators are used. Query expansion terms and term weighting is same as our proposed system. This system is equivalent to the baseline Okapi BM25 system.

### 4.2 Discussion of the Experimental Results

Table 1 shows evaluation measures for each version of the system and Figure shows normalized Discounted Cumulative Gain (nDCG) of each system per topic. NpVp, using descriptions only in the query, is the best-performed system for the NTCIR-8 GeoTime Japanese monolingual task.

**Table 1.** Evaluation measures for each version of the system

| | NfVp | NpVp | BO |
|---|---|---|---|
| AP | 0.3697 | 0.3719 | 0.2881 |
| nDCG | 0.4117 | 0.4162 | 0.3282 |
| Q | 0.5710 | 0.5881 | 0.4993 |

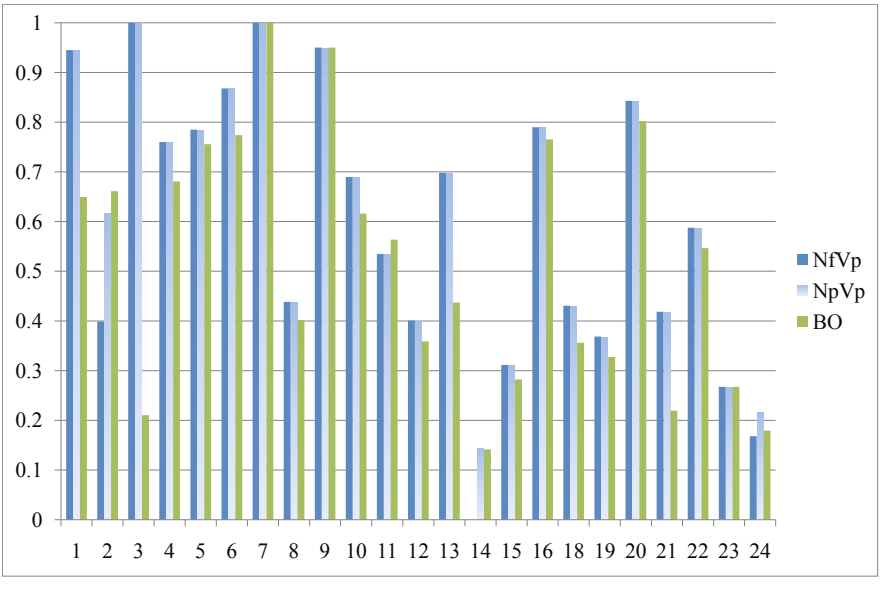


**Fig. 3.** Normalized Discounted Cumulative Gain (nDCG) of each System per Topic

Comparing NfVp and NpVp enables us to discuss the effectiveness of the Boolean query for the filter. For 14 of the topics (4, 5, 7, 8, 10, 11, 12, 15, 16, 18, 21, 22, 23, and 24), the system could not make a strict Boolean query for selecting small numbers of documents , and the results for NfVp were the same as for NpVp. For seven of the topics (Topic:Boolean_matched_documents 1:772, 3:1, 6:275, 9:6, 13:105, 19:329, and 20:945), the system did make an appropriate Boolean query, retrieving all relevant documents using fewer documents. For the remaining three topics (Topic:filter_out/total_rel 2:26/48, 14:2/2, and 25:1/3), the constructed Boolean queries were too strict, and some relevant documents were filtered out. For example, topic 2, “ハリケーン・カトリーナ” (hurricane Katrina) is recognized as an NE. Therefore, articles with “カトリーナ” (Katrina) and without “ハリケーン” (hurricane) were filtered out. The quality of Boolean query filter for NE is highly depends on the one of NE tagging system. The topic 14 includes an NE keyword “アフリカ” (Africa). However, the relevant documents have the name of the African country “コンゴ民主共和国”

(Democratic Republic of the Congo) instead of “アフリカ”. To deal with such relations, it is necessary to have a good query analyzer and a mechanism to deal with the part-whole relationship when generating the related keyword list for the Boolean query. Topic 25 has an NE keyword “スマトラ沖” (off the coast of Smatra) but a relevant document that has exculded with Boolean filter has “スマトラ島沖” (off the coast of Smatra island) and does not have “スマトラ沖” (off the coast of Smatra). Because of this problem, the system performance for NfVp was worse than for NpVp.

Because NpVp performs better than NfVp, we use the comparison between NpVp and BO (base line) to analyze the effectiveness of using a Boolean query. The $t$ test and Wilcoxon Signed Rank test were used to compare the AP, the normalized Discounted Cumulative Gain (nDCG), and the Q measure (Q). From the results of the $t$ test at a significance level of 0.05 for two-sided tests, the differences for nDCG (0.018) and Q (0.040) are statistically significant and that for AP(0.055) is not significant. For the Wilcoxon Signed Rank tests at a significance level of 0.01 for two-sided tests, the AP (0.0015), nDCG (0.0006), and Q (0.0024) results are statistically significant.

There were three topics (2 (AP, nDCG, Q), 11 (AP, nDCG, Q), and 21 (AP)) where the results for NpVp were worse than for BO.

For topic 2, “ハリケーン” (hurricane) is recognized as an NE and articles about “ハリケーン” (hurricane) without “カトリーナ” (Katrina) get a similar score to “カトリーナ” (Katrina) without “ハリケーン” (hurricane).

For topics 11 and 21, these topics do not contain NE information. In such cases, it is difficult to assure the quality of the generated query.

Based on the comparison of the relevant documents and queries generated by our system, we found there are many relevant documents that do not have keywords of each query. It is necessary to have such Boolean query modification mechanism for constructing IR4QA system. In addition, it is very difficult to construct a perfect Boolean query from given query and pseudo-relevant documents, Boolean penalty type system performs well especially for the survey type question.

## 5 Conclusion

In this paper, we propose to use ABRIR as an IR system for QA about particular NEs. From an evaluation experiment using the NTCIR-8 GeoTime Japanese monolingual task, we confirm that ABRIR can be used in a system that uses appropriate Boolean queries and penalties to outperform a baseline system (the probabilistic IR model, Okapi BM25).

## Acknowledgment

This research was partially supported by a Grant-in-Aid for Scientific Research (B) 21300029, from the Japan Society for the Promotion of Science. I would also like to thank organizers of the NTCIR-8 GeoTime task and the reviewer of the paper for their fruitful contribution.

## References

1. Greenwood, M.A. (ed.): Proceedings of the 2nd workshop on Information Retrieval for Question Answering (2008)
2. Sakai, T., Kando, N., Lin, C.J., Mitamura, T., Shima, H., Ji, D., Chen, K.H., Nyberg, E.: Overview of NTCIR-7 ACLIA IR4QA task. In: Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Quesiton Answering, And Cross-Lingual Information Access, pp. 63–93 (2010)
3. Sakai, T., Shima, H., Kando, N., Song, R., Lin, C.J., Mitamura, T., Sugimoto, M., Lee, C.W.: Overview of NTCIR-8 ACLIA IR4QA. In: Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Quesiton Answering, And Cross-Lingual Information Access, pp. 63–93 (2010)
4. Yoshioka, M., Haraguchi, M.: 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)
5. Robertson, S.E., Walker, S.: Okapi/Keenbow at TREC-8. In: Proceedings of TREC-8, pp. 151–162 (2000)
6. Uchiyama, M., Isahara, H.: Implementation of an IR package. In: IPSJ SIGNotes, 2001-FI-63, 57–64 (2001) (in Japanese)
7. Matsumoto, Y., Kitauchi, A., Yamashita, T., Hirano, Y., Matsuda, H., Takaoka, K., Asahara, M.: Morphological Analysis System ChaSen version 2.2.1 Manual. Nara Institute of Science and Technology (2000)
8. Japan Electronic Dictionary Research Institute, Ltd (EDR): EDR Electronic Dictionary Version 2.0 Technical Guide TR2-007 (1998)
9. Masuyama, T., Sekine, S., Nakagawa, H.: Automatic construction of Japanese katakana variant list from large corpus. In: COLING 2004: Proceedings of the 20th international onference on Computational Linguistics, Association for Computational Linguistics,Morristown, NJ, USA, pp. 1214–1219 (2004)
10. Goto, I., Kato, N., Ehara, T., Tanaka, H.: Back transliteration from Japanese to English using target English context. In: COLING 2004: Proceedings of the 20th international Conference on Computational Linguistics, Association for Computational Linguistics, Morristown, NJ, USA, pp. 827–833 (2004)
11. Kudo, T., Matsumoto, Y.: Japanese dependency analysis using cascaded chunking. In: CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops), pp. 63–69 (2002)
12. Gey, F., Larson, R., Kando, N., Machado-Fisher, J., Sakai, T.: NTCIR-GeoTime overview: Evaluating geographic and temporal search. In: Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Quesiton Answering, And Cross-Lingual Information Access, pp. 147–153 (2010)