Study on the Combination of Probabilistic and Boolean IR Models for WWW Documents Retrieval

Masaharu YOSHIOKA Makoto HARAGUCHI Hokkaido University
{yoshioka,makoto}@db-ei.eng.hokudai.ac.jp

Background

- Requirement for IR system with large scale text data
- Different IR models
  - A probabilistic model
    - The user may not select query term appropriately.
  - A Boolean model
    - The user must select query term appropriately.
    - A Boolean query formula is expressive but is very difficult to construct appropriate one.

Objective

- Evaluate following IR systems.
  - our IR system, which is based on the probabilistic IR model.
  - our method for combining probabilistic and Boolean IR models for clarifying queries.

IR System (Probabilistic IR Model)

- Modified version of OKAPI
  - Use BM25 formula to calculate each document score

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

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

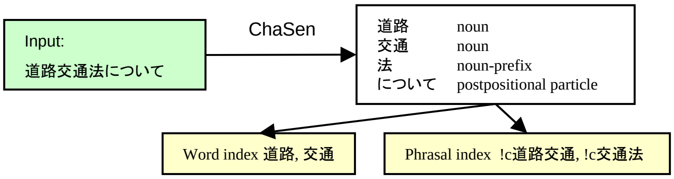
  - Term weighting for phrasal terms
  - Document score may differ according to the dictionary entry

情報処理→ Word 情報処理
情報科学→ Word 情報, 科学 Phrase !c情報科学

  - Discount score for phrasal index $qtf = c * qtf_c$

Index

- Word and phrasal index
  - Use ChaSen as morphological analyzer and select noun (noun, unknown, symbol) for word index
  - Phrasal index: a pair of adjacent noun terms
    - We use prefixes, postfixes, and numbers in addition to words that are used for word index
- Database engine: Generic Engine for Transposable Association (GETA)



Input:
道路交通法について

ChaSen

道路	noun
交通	noun
法	noun-prefix
について	postpositional particle

Word index 道路, 交通

Phrasal index !c道路交通, !c交通法

An Example of Index Extraction

Relevance Feedback

- Pseudo-relevance feedback
  - Use top 5 ranked documents of initial retrieval are used as relevant documents.
  - Reject documents with small number of terms in it.
- Query expansion
  - Use terms in relevant documents as query terms
  - Max: 300 terms
  - Rocchio-type feedback

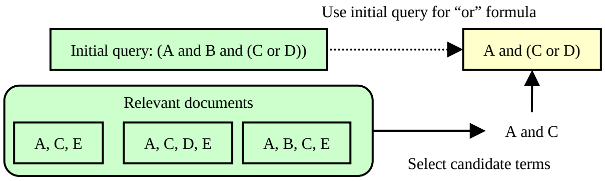
$$qtf = \alpha * qtf_0 + (1-\alpha) * \frac{\sum_{i=1}^{R} qtf_i}{R}$$

Characteristics of Two IR Models

	Assumption of user	Selected Documents	Readability
A probabilistic model	The user may have difficulties to select appropriate query terms,	Documents that do not contain a part of query terms may select as higher relevant ones.	Difficulties to understand appropriateness of query
A Boolean model	The user can select appropriate query terms.	Documents that do not satisfy a Boolean formula is not selected	The user can easily understand why the IR system selects the documents

Reconstruction of a Boolean Query Formula

- Relax an initial Boolean query formula to include given relevant documents as relevant one
  - Use terms that exists in all relevant documents and also exists in an initial query as a candidate to construct a relaxed Boolean query formula
  - Use an initial query for “or” formula



Use initial query for “or” formula

Initial query: (A and B and (C or D))

A and (C or D)

Relevant documents

A, C, E

A, C, D, E

A, B, C, E

A and C

Select candidate terms

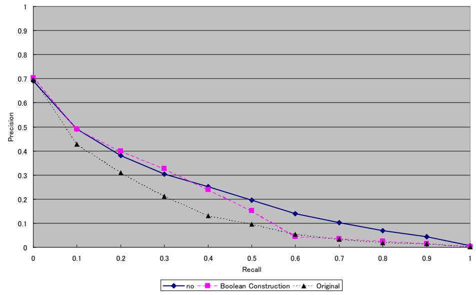
Combination of Two IR Models

- Two approach
  - Use a Boolean IR model first and calculate score of each retrieved document by using a probabilistic model
  - Use a probabilistic IR model first and apply penalty for documents that do not satisfy a Boolean query formula
    - Penalty is calculated by using term importance in BM25

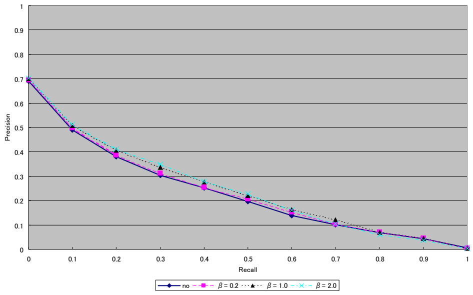
$$\beta \times w^{(1)} \times \frac{(k_3+1)qtf}{k_3+qtf}$$

β: parameter

    - Penalty is calculated for each “and” element
    - For “or” formula, use penalty of a term that has highest one among them.



R-P Graph for Different Boolean Query



R-P Graph for Different β

Conclusion

- A proposal of our IR system based on a probabilistic IR model
  - We confirm the system has better performance in NTCIR-4 submission.
  - This system may be good enough to use as a benchmark system.
- A proposal of a combination of two IR models
  - User defined Boolean query is not precise enough to retrieve all relevant documents
  - Relaxing an initial Boolean query formula by using relevant documents improve quality of a Boolean query formula
  - Penalty calculation by using a Boolean query formula improves retrieval performance