Query by document via a decomposition-based two-level retrieval approach

Linkai Weng, Zhiwei Li, Rui Cai, Yaoxue Zhang, Yuezhi Zhou, Laurence T. Yang, and Lei Zhang. 2011
INTRODUCTION

Document query != short query

- **Document query:**
  can contain many words (e.g. 2000)

- **Short query:**
  contains 2-10 words

Authors name this issue as **Query by document** (QBD)
A QUICK OVERVIEW

Retrieving similar documents from a large-scale text corpus according to a given document

Problems of indexing techniques due to:

- High dimensionality
- Sparse representation
- Semantic issue

Authors propose a **two-level retrieval solution based on a document decomposition**:  
- Document decomposition model (DDM)  
- Effective two-level index and rank scheme
THE PROBLEM OF QUERY BY DOCUMENT (QBD)

- High dimensionality • document query contains many terms
- Sparse representation • A document query is often represented as a sparse vector
- Semantic issue • A document query often has its important inherent semantics
QBD AND HIGH-DIMENSIONAL INDEXING (HDI) PROBLEM

BASIC HDI IDEA:

- Partition a feature space to lots of “bins”.

TYPICAL HDI APPROACHES:

- Tree-based (kd-tree) - only applied to "low" dimensional spaces

- Hash based (locality sensitive hashing or LHS) - effective for images but not good at indexing sparse feature vectors (such as TF-IDF vectors)
DIMENSION REDUCTION (DR)

Examples:
- Matrix factorization (latent semantic indexing (LSI))
- Nonnegative matrix factorization (NMF)
- Topic models (pLSI, LDA)

High dimensional feature vectors are projected to low dimensional

New representations of documents often are dense

Similarity measures (e.g. HDI) can work and synonymy and polysemy issues are less severe

Illustration is taken from the paper
Purpose: find a new representation which is easier to index

Authors' idea is related to dimension reduction approaches

\[ d = \mu + Xw + \epsilon \]

\[ d = \begin{bmatrix} \cdot & \\ \cdot & \phi_{ij} & \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \theta_d \gamma_{d1} \end{bmatrix} + \begin{bmatrix} \cdot \end{bmatrix} \gamma_{d2} + \begin{bmatrix} \cdot \end{bmatrix} \gamma_{d3} \]
Document decomposition model (DDM)

- **background words** (widely appear in documents)
- **topic related words** (commonly appear in documents with similar topic)
- **document specific words** (only appear in few documents)
DOCUMENT DECOMPOSITION (CONT'D)

- Less storage cells are needed to store the new document representation

+ The new representation can increase the storage cost of short documents

Authors suggest implementing document decomposition idea by a probabilistic topic model
INDEXING AND RANKING: INDEXING

Indexing workflow

- Document
- DDM
- Background/stop Words
- Topic Related Words
- Compact Vector
- Document Specific Words
- LSH
- LSH Index
- Forward Files
- <word id, weight ($\varphi_w$)>
INDEXING AND RANKING: RANKING FUNCTION

A document in corpus & query document: \( \langle \theta, \varphi \rangle \)

Similarity of two documents: Linear combination of Compact vector and A few keywords

"Accurate ranking approach"

Purpose: define the upper-bound of retrieval accuracy for evaluation
INDEXING AND RANKING: RANKING FUNCTION (CONT'D)

Similarity of two documents: Linear combination of \( \langle \theta, \varphi \rangle \).

\[
sim(d, q) = \gamma_d \gamma_q \sim(\theta_d, \theta_q) + \gamma_d \gamma_q \sim(\varphi_d, \varphi_q)
\]

Word ratios of topic related (TS) words

Word ratios of document specific (DS) words
TWO-LEVEL INDEX AND RANKING

Procedures:

1. Search the Locality Sensitive Hashing (LSH) index
2. Get a set of candidate documents (should be in the same topics as the query)
3. Re-rank the documents by their document specific keywords

\[ sim(d, q) = \gamma_1 \gamma_q \cdot sim(\theta_d, \theta_q) + \gamma_2 \gamma_q \cdot sim(\varphi_d, \varphi_q) \]

- Word ratios of topic related (TS) words
- Word ratios of document specific (DS) words
- If documents belong to different categories
- If documents are in the same category
EVALUATION SETUP

• Proposed approaches:
  1. Decomposition-Based Approaches (DDM)
  2. Decomposition-Based Approaches + two-level index and rank schema (DDM+Index)

• Data sets:
  • 20 newsgroups (20NG)
  • Wikipedia pages (Wiki)
  • A randomly selected subset of Wikipedia (SWiki)
  • Corpus with 5 million pages (5M) randomly sourced from the TREC'08 corpus

• Baselines:
  1. TF-IDF with cosine similarity of Neto & Yates (1999) (TFIDF)
  2. Direct indexing TF-IDF vectors by LSH of Andoni & Indyk (2008) (LSH)
  3. Similarity computed only by top-N TF-IDF words (TopN) of Yang et al. (2009) (TopN)
  4. Linear combination of TopN & similarity computed by topic mixtures obtained by LDA of Wei & Croft (2007) (TopNLD)

• Evaluation goals:
  1. DDM's performance
  2. Performance of the index and ranking solution
  3. Scalability
RESULTS - DDM

Figure 4: Macro-average Pr@10 change with the number of topics on 20NG, Wiki, and SWiki. TS means that only topic mixtures are used to compute document similarity, DS means that only DS words are used to perform retrieval, and ODP means that topics are got by counting ODP pages.

Figure 5: Macro-average Pr@10 change with the number of document specific words on 20NG, Wiki, and SWiki. LSI and LDA mean that topics are obtained by LSI decomposition (3000 documents were used to perform LSI decomposition) and LDA model, respectively.
RESULTS – INDEXING SCHEME

Figure 6: Macro-average Pr@10 of index scheme change with the number of topics on 20NG, Wiki, and SWiki

Figure 7: Macro-average Pr@10 of index scheme change with the number of document specific words on 20NG, Wiki, and SWiki
RESULTS - SCALABILITY

Building cost

<table>
<thead>
<tr>
<th></th>
<th>Time cost</th>
<th>Memory cost (5M corpus)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>50 hours</td>
<td>2.1GB</td>
</tr>
<tr>
<td>Loaded</td>
<td>70 minutes</td>
<td>800MB</td>
</tr>
</tbody>
</table>

Time to answer a query

Illustration is taken from the paper
CONCLUSION

**DDM**
- Long query document

**Index & ranking**
- Computation cost
- Memory cost
- Accurate search results

**Scalability**
- Expand to large scale corpus
REFERENCES

Presentation is based on:

Thank you for listening!