

Helsinki's EXTIRP@INEX

Antoine Doucet, Lili Aunimo,
Miro Lehtonen, Renaud Petit

University of Helsinki
Department of Computer Science

2nd INEX Workshop - Schloss Dagstuhl, 17.12.2003

EXTIRP's introduction

EXacT coverage IR based on static Passage clusters

- Exact Coverage IR: CO topics only
 - We only used the <title> and <description> elements (TD)
- Documents' logical structure is only used as a means to model **coherent** minimal fragments

EXTIRP's outline

EXacT coverage IR based on static Passage clusters

- Defining Minimal Retrieval Units (MRU)
- Modeling those units
- Computing an RSV for each minimal unit, by measuring its similarity w.r.t. a given query
- Expanding MRU RSVs to their ancestors

EXTIRP's extras

EXacT coverage IR based on static Passage clusters

- A new type of phrases in IR
- Blind Relevance Feedback

Minimal Retrieval Units

Defining Minimal Retrieval Units

- Refined XML fragments
- Several Granularities
 - "meta"-paragraphs: (p,p1,p2,ip1,ip2,ip3,bq)
 - "meta"-subsections: (sec,fm,bm,dialog,vt)
- True inline elements can be duplicated (b,i,tt,..)
 - assumption: they are more important
- Minimal Size Threshold

Defining Minimal Retrieval Units

- Nearest Level Titles included: (st,apt,atl,..)
- Skipped elements:
(ref,fig,tbl,footnote,sub,pdt,pp,art,tf,math,
tmath,volno,issno,no,colspec,spanspec,..)

Modeling Minimal Retrieval Units

Modeling Minimal Retrieval Units

- Two models
 - Maximal Frequent Sequences
 - Baseterm Vector Space Model

Maximal Frequent Sequences

- A sequence that occurs frequently enough
- Words do not need to be contiguous (a gap is allowed)
- A Very compact representation
 - as opposed to statistical phrases

Maximal Frequent Sequences

Preprocessing

...President of the United States Bush...

...President George W. Bush...

Maximal Frequent Sequences

Preprocessing: a gap is allowed

...President of the United States Bush...

- President United
- President States
- **President Bush**
- United States
- United Bush
- States Bush

Maximal Frequent Sequences

Recall the other text fragment:

...President George W. Bush...

- President George
- **President Bush**
- George Bush

Maximal Frequent Sequences

In both of these text fragments:

`...President of the United States Bush...`

`...President George W. Bush...`

A phrase "President Bush" is found.

Maximal Frequent Sequences

- Offline, each MRU is bound to a (possibly empty) set of Maximal Frequent Sequences
- We decompose each MFS into a set of word pairs and associated weights
- The base weight is the idf of the pair, modified by the way the pair was formed
 - e.g., if a pair of words was formed from non-adjacent word occurrences, its weight is lowered

Maximal Frequent Sequences

- We define this weight to be the "quantity of relevance" brought by the corresponding pair, would that pair be found in a query.
- Given a query, each MRU receives the sum of all quantities of relevance brought by the matching pairs it was attached.

Maximal Frequent Sequences

- Knowledge Discovery Technique
 - Complexity Problems...
 - We could extract MFS through a trick:
 - ❖ Cluster the set of MRUs into disjoint subsets using the k-means clustering algorithm (for it's partitional and of linear complexity)
 - ❖ Compute the set of MFS for each subset
 - ❖ Join all MFS sets

Vector Space Model

- With MFS, many unfrequent words are not found. Frequent words not frequently co-occurring together with other frequent words are also missed.
- This motivates the need for a complementary baseterm representation.
- For this, we used a standard tfidf representation of Salton et al. and calculated RSVs using the cosine function.

Evaluating Minimal Retrieval Units

- Two representation => Two RSVs
- We need to aggregate those 2 RSVs
- We did this using linear interpolation factors:
$$RSV = \alpha * RSV_{\text{baseterm}} + \beta * RSV_{\text{MFS}}$$
- Where α is the number of distinct words from the query and β is the number of distinct words found in keyphrases from the query.

RSV Upward Propagation

RSV Upward Propagation

- For each MRU m , each ancestor a of m gets the following score:

$$score(a) + = \frac{score(m)}{size^p(a)}$$

- where p is known as the *RSV upward propagation factor*
- The higher p , the more the big elements are favored

RSV Upward Propagation

- Post-Processing:
 - Do not show to the user what she may have already seen
 - For each XML document:
 - ❖ rank all elements by decreasing score
 - ❖ prune all fragments with a lower score than at least one of their ancestors

Blind Relevance Feedback

- Top ranked document elements are considered
- Selected terms are added to the query with half the weight of the original terms
- Added 0 to 10 terms to each query

Our Runs

Runs	MRU granularity	Upward Propagation Factor	QE	Strict inex_eval	Generalized inex_eval
Run1	Paragraph	2	No	.0061	.0105
Run2	Subsection	2	Yes	.0323	.0222
Run3	Subsection	5	Yes	.0449	.0235

Results

- `inex_eval`, better scores are obtained when retrieving more large elements
- better results with `inex_eval_ng`.

Conclusion – Discussion

Conclusion

- No redundancy in the index
 - No token is indexed twice
 - The index is smaller than the collection
- Still, any “senseful” element can be retrieved
- No redundancy in the results
 - Bad results with `inex_eval`
 - ❖ Still, `run3` does better than restricting retrieval to full documents
 - Better ones with `inex_eval_ng`

Conclusion

- Most of this is brand new..
 - Refining XML fragments
 - Use of MFS in IR
 - Upward propagation
- The propagation formula is exponential making it tough to adapt to a new collection