A Probabilistic Retrieval Model for Semistructured Data

Jinyoung Kim, Xiaobing Xue and W. Bruce Croft

CIIR in Univ. of Mass. Amherst

ECIR'09

Outline

Semistructured Data

Characteristic:

- Data-centric
  - Each element describes a particular aspect of the item
- Noisy structure
  - Elements have full-text contents, with overlaps in language

Examples:

```
<movie>
 <title> French Kiss
 <year> 1995
 <language> English
 <genre> Comedy,Romance
 <cast>
    <actress> Ryan, Meg I
 </cast>
 <team>
    <director> Kasdan, Lawrence
 </team>
 <plot>
    An American woman Kate(Meg Ryan) ...
 </plot>
```

```
<resume>
 <title> Experienced programmer
 <description>
   I have 20 years ...
 </description>
 <educations>
   <school_record> Programming in Java
   ...
 </educations>
 <skills>
   <skill_record>
      Worked as Java  ...
   </skill_record>
   ...
```

Semistructured Data

Characteristic:

- Data-centric
  - Each element describes a particular aspect of the item
- Noisy structure
  - Elements have full-text contents, with overlaps in language

Examples:

```
<movie>
 <title> French Kiss
 <year> 1995
 <language> English
 <genre> Comedy,Romance
 <cast>
    <actress> Ryan, Meg I
 </cast>
 <team>
    <director> Kasdan, Lawrence
 </team>
 <plot>
    An American woman Kate(Meg Ryan) ...
 </plot>
```

```
<resume>
 <title> Experienced programmer
 <description>
   I have 20 years ...
 </description>
 <educations>
   <school_record> Programming in Java
   ...
 </educations>
 <skills>
   <skill_record>
      Worked as Java  ...
   </skill_record>
   ...
```

Retrieval of Semistructured Data

Existing solutions:

- Keyword query
  - 'meg ryan romance'
  - Easier yet ineffective
- Form-based search interface
  - actress : meg ryan / genre : romance
  - Effective yet more effort

Simplicity-effectiveness trade-off:

- How can we make the keyword query more effective?

Retrieval of Semistructured Data

Existing solutions:

- Keyword query
  - 'meg ryan romance'
  - Easier yet ineffective
- Form-based search interface
  - actress : meg ryan / genre : romance
  - Effective yet more effort

Simplicity-effectiveness trade-off:

- How can we make the keyword query more effective?

Outline

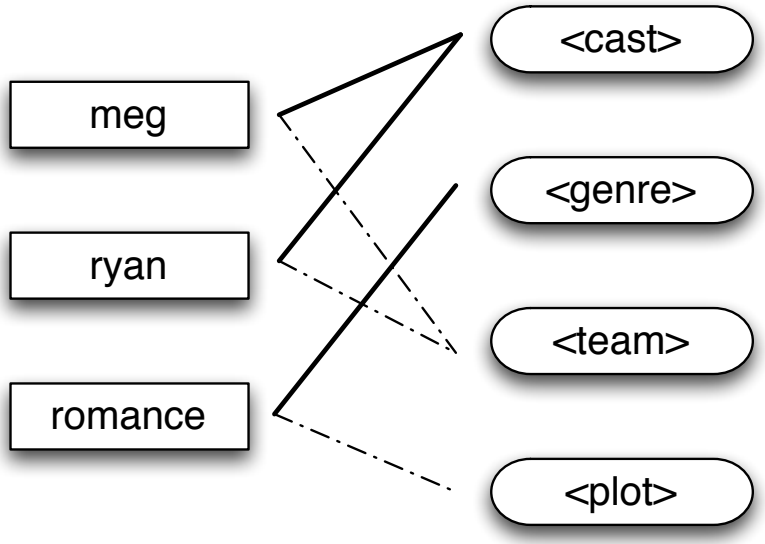
Mapping between Query-words and Elements

Key observation:

- Users implicitly map each query-word into XML element(s)

An example query :

- 'meg ryan romance'



Exploiting Mapping for Retrieval

1) Convert keyword query into a structured query [PCD09]
   - e.g. *//movie[cast='meg ryan' and genre='romance']*
   - Rank list is desirable in many cases
   - Inferred structure is ambigious
2) Incorporate structure into the probabilistic retrieval model
   - e.g. gibson : (0.4 *cast* / 0.3 *team* / ...)
   - Use this as element-level weights for each query-word

Exploiting Mapping for Retrieval

1) Convert keyword query into a structured query [PCD09]
   - e.g. *//movie[cast='meg ryan' and genre='romance']*
   - Rank list is desirable in many cases
   - Inferred structure is ambigious
2) Incorporate structure into the probabilistic retrieval model
   - e.g. gibson : (0.4 *cast* / 0.3 *team* / ...)
   - Use this as element-level weights for each query-word

Probabilistic Retrieval Model for Semistructured Data

Notation:

- Query $Q = (q_1, q_2, ..., q_m)$
- Collection schema $C = (E_1, E_2, ..., E_n)$
- XML documents $d = (e_1, e_2, ..., e_n)$

Posterior probability that q_i be mapped into E_j:

$$P_M(E_j|q_i) = \frac{P_M(q_i|E_j)P_M(E_j)}{P(q_i)} \quad (1)$$

- $P_M(q_i|E_j)$: $tf(q_i, E_j)/|E_j|$
- $P_M(E_j)$: prior belief on element E_j

Probabilistic Retrieval Model for Semistructured Data (PRMS)

- $P_{QL}(q_i|e_j)$: element-level evidence

$$P(Q|d) = \prod_{i=1}^{m} \sum_{j=1}^{n} P_M(E_j|q_i) P_{QL}(q_i|e_j) \quad (2)$$

Probabilistic Retrieval Model for Semistructured Data

Notation:

- Query $Q = (q_1, q_2, ..., q_m)$
- Collection schema $C = (E_1, E_2, ..., E_n)$
- XML documents $d = (e_1, e_2, ..., e_n)$

Posterior probability that q_i be mapped into E_j:

$$P_M(E_j|q_i) = \frac{P_M(q_i|E_j)P_M(E_j)}{P(q_i)} \tag{1}$$

- $P_M(q_i|E_j)$: $tf(q_i, E_j)/|E_j|$
- $P_M(E_j)$: prior belief on element E_j

Probabilistic Retrieval Model for Semistructured Data (PRMS)

- $P_{QL}(q_i|e_j)$: element-level evidence

$$P(Q|d) = \prod_{i=1}^{m} \sum_{j=1}^{n} P_M(E_j|q_i) P_{QL}(q_i|e_j) \tag{2}$$

Probabilistic Retrieval Model for Semistructured Data

Notation:

- Query $Q = (q_1, q_2, ..., q_m)$
- Collection schema $C = (E_1, E_2, ..., E_n)$
- XML documents $d = (e_1, e_2, ..., e_n)$

Posterior probability that q_i be mapped into E_j:

$$P_M(E_j|q_i) = \frac{P_M(q_i|E_j)P_M(E_j)}{P(q_i)} \tag{1}$$

- $P_M(q_i|E_j)$: $tf(q_i, E_j)/|E_j|$
- $P_M(E_j)$: prior belief on element E_j

Probabilistic Retrieval Model for Semistructured Data (PRMS)

- $P_{QL}(q_i|e_j)$: element-level evidence

$$P(Q|d) = \prod_{i=1}^{m} \sum_{j=1}^{n} P_M(E_j|q_i) P_{QL}(q_i|e_j) \tag{2}$$

Comparison with Existing Approaches

Element-level weight $(\mu_1, ..., \mu_n)$

- fixed regardless of query-word (c.f. $P_M(E_j|q_i)$)

Hierarchical Language Model: [OC04]

$$P(Q|d) = \prod_{i=1}^{m} \sum_{j=1}^{n} \mu_j P_{QL}(q_i|e_j) \tag{3}$$

BM25F: [RZT04]

$$S(q,d) = \sum_{q_i \in Q} idf(q_i) \frac{weight(q_i,d)}{k_1 + weight(q_i,d)} \tag{4}$$

$$weight(q_i,d) = \sum_{e_j \in d} \frac{\mu_j tf(q_i,e_j)}{(1-b_j) + b_j \times \frac{|e_j|}{|E_j|}} \tag{5}$$

Comparison with Existing Approaches

Element-level weight $(\mu_1, ..., \mu_n)$

- fixed regardless of query-word (c.f. $P_M(E_j|q_i)$)

Hierarchical Language Model: [OC04]

$$P(Q|d) = \prod_{i=1}^{m} \sum_{j=1}^{n} \mu_j P_{QL}(q_i|e_j) \tag{3}$$

BM25F: [RZT04]

$$S(q,d) = \sum_{q_i \in Q} idf(q_i) \frac{weight(q_i,d)}{k_1 + weight(q_i,d)} \tag{4}$$

$$weight(q_i,d) = \sum_{e_j \in d} \frac{\mu_j tf(q_i,e_j)}{(1-b_j) + b_j \times \frac{|e_j|}{|E_j|}} \tag{5}$$

Comparison with Existing Approaches

Element-level weight $(\mu_1, ..., \mu_n)$

- fixed regardless of query-word (c.f. $P_M(E_j|q_i)$)

Hierarchical Language Model: [OC04]

$$P(Q|d) = \prod_{i=1}^{m} \sum_{j=1}^{n} \mu_j P_{QL}(q_i|e_j) \tag{3}$$

BM25F: [RZT04]

$$S(q,d) = \sum_{q_i \in Q} idf(q_i) \frac{weight(q_i,d)}{k_1 + weight(q_i,d)} \tag{4}$$

$$weight(q_i,d) = \sum_{e_j \in d} \frac{\mu_j tf(q_i,e_j)}{(1-b_j) + b_j \times \frac{|e_j|}{|E_j|}} \tag{5}$$

Outline

Experimental Setup

IMDB XML Collection:

- 430,000 movie descriptions (avg. length : 96 words)
- 50 queries (10 training / 40 test)
- 1 - 3 relevant documents per query (known-item)

Monster XML Collection:

- 1,034,795 resumes (avg. length : 154 words)
- 50 actual queries by real users (10 training / 40 test)
- 10 - 20 relevant documents per query found by pooling (ad-hoc)

Tested Runs:

- DQL : document query-likelihood [PC98]
- HLM : hierarchical language model [OC04]
- BM25F : BM25 using raw frequency combination [RZT04]
- PRMS : suggested model

Experimental Setup

IMDB XML Collection:

- 430,000 movie descriptions (avg. length : 96 words)
- 50 queries (10 training / 40 test)
- 1 - 3 relevant documents per query (known-item)

Monster XML Collection:

- 1,034,795 resumes (avg. length : 154 words)
- 50 actual queries by real users (10 training / 40 test)
- 10 - 20 relevant documents per query found by pooling (ad-hoc)

Tested Runs:

- DQL : document query-likelihood [PC98]
- HLM : hierarchical language model [OC04]
- BM25F : BM25 using raw frequency combination [RZT04]
- PRMS : suggested model

Experimental Setup

IMDB XML Collection:

- 430,000 movie descriptions (avg. length : 96 words)
- 50 queries (10 training / 40 test)
- 1 - 3 relevant documents per query (known-item)

Monster XML Collection:

- 1,034,795 resumes (avg. length : 154 words)
- 50 actual queries by real users (10 training / 40 test)
- 10 - 20 relevant documents per query found by pooling (ad-hoc)

Tested Runs:

- DQL : document query-likelihood [PC98]
- HLM : hierarchical language model [OC04]
- BM25F : BM25 using raw frequency combination [RZT04]
- PRMS : suggested model

Query & Mapping Probability Examples – IMDB

Table: Example queries

#	Query	Elements
1	meg ryan war	cast, genre
2	redemption crime	title, genre
3	love letter iwai	title, team
4	ziyi zhang hidden tiger	team, title
5	brokeback ang lee	title, team

Table: Estimated mapping probabilities

Query 1			
meg	cast:0.407	team:0.382	title:0.187
ryan	cast:0.601	team:0.381	title:0.017
war	genre:0.927	title:0.070	location:0.002
Query 2			
redemption	title:0.983	location:0.017	year:0.000
crime	genre:0.990	title:0.010	location:0.000

Query & Mapping Probability Examples – IMDB

Table: Example queries

#	Query	Elements
1	meg ryan war	cast, genre
2	redemption crime	title, genre
3	love letter iwai	title, team
4	ziyi zhang hidden tiger	team, title
5	brokeback ang lee	title, team

Table: Estimated mapping probabilities

Query 1			
meg	cast:0.407	team:0.382	title:0.187
ryan	cast:0.601	team:0.381	title:0.017
war	genre:0.927	title:0.070	location:0.002
Query 2			
redemption	title:0.983	location:0.017	year:0.000
crime	genre:0.990	title:0.010	location:0.000

Results – IMDB

Table: Retrieval performance

Measure	DQL	HLM	BM25F	**PRMS**
MAP	0.374	0.344(-8.8%)	0.574(53.4%)	**0.661(76,7%)**
RecipRank	0.405	0.350(-15.7%)	0.576(42.2%)	**0.664(64.0%)**

Table: Element weights and the retrieval performance for an example query

PRMS - RecipRank : 1.0				
gladiator	title:0.634	genre:0.000	plot:0.25	team:0.000
action	title:0.005	genre:0.971	plot:0.01	team:0.000
Maximus	title:0.627	genre:0.000	plot:0.181	team:0.000
Scott	title:0.011	genre:0.000	plot:0.020	team:0.440
HLM - RecipRank : 0.01				
all terms	title:0.229	genre:0.108	plot:0.024	team:0.048

Results – IMDB

Table: Retrieval performance

Measure	DQL	HLM	BM25F	**PRMS**
MAP	0.374	0.344(-8.8%)	0.574(53.4%)	**0.661(76,7%)**
RecipRank	0.405	0.350(-15.7%)	0.576(42.2%)	**0.664(64.0%)**

Table: Element weights and the retrieval performance for an example query

PRMS - RecipRank : 1.0				
gladiator	title:0.634	genre:0.000	plot:0.25	team:0.000
action	title:0.005	genre:0.971	plot:0.01	team:0.000
Maximus	title:0.627	genre:0.000	plot:0.181	team:0.000
Scott	title:0.011	genre:0.000	plot:0.020	team:0.440
HLM - RecipRank : 0.01				
all terms	title:0.229	genre:0.108	plot:0.024	team:0.048

Results – Monster

Table: Example queries

#	Query
1	executive assistant power point excel type schedule
2	microbiologist hopkinsville veterinary
3	technical writer scottsdale arizona java coldfusion
4	healthcare financial analyst minneapolis bachelors
5	developer asp sql vb unix Cincinnati

Table: Retrieval performance

Measure	DQL	HLM	BM25F	**PRMS**
MAP	0.432	0.439(1.6%)	0.33(-23.7%)	**0.530(22.7%)**
Prec@5	0.51	0.555(8.8%)	0.46(-9.8%)	**0.61(19.6%)**
Prec@10	0.502	0.487(-3%)	0.415(-17.4%)	**0.577(14.9%)**
Prec@20	0.483	0.394(-18.4%)	0.311(-35.7%)	**0.545(12.8%)**

Results – Monster

Table: Example queries

#	Query
1	executive assistant power point excel type schedule
2	microbiologist hopkinsville veterinary
3	technical writer scottsdale arizona java coldfusion
4	healthcare financial analyst minneapolis bachelors
5	developer asp sql vb unix Cincinnati

Table: Retrieval performance

Measure	DQL	HLM	BM25F	**PRMS**
MAP	0.432	0.439(1.6%)	0.33(-23.7%)	**0.530(22.7%)**
Prec@5	0.51	0.555(8.8%)	0.46(-9.8%)	**0.61(19.6%)**
Prec@10	0.502	0.487(-3%)	0.415(-17.4%)	**0.577(14.9%)**
Prec@20	0.483	0.394(-18.4%)	0.311(-35.7%)	**0.545(12.8%)**

Outline

Conclusion & Future Work

Our work showed that:

- Query-element mapping can be estimated by collection statistics
- Incorporating this into retrieval model improves performance

PRMS was also effective in:

- IMDB collection using 1,200 real user queries
- W3C e-mail collection using 125 TREC known-item queries
  - PRMS (MRR:0.617) closely matches best TREC run (MRR:0.621)

Future Work

- Better estimation of mapping probabilities & element-level scores
- Application to heterogeneous XML collection retrieval

Conclusion & Future Work

Our work showed that:

- Query-element mapping can be estimated by collection statistics
- Incorporating this into retrieval model improves performance

PRMS was also effective in:

- IMDB collection using 1,200 real user queries
- W3C e-mail collection using 125 TREC known-item queries
  - PRMS (MRR:0.617) closely matches best TREC run (MRR:0.621)

Future Work

- Better estimation of mapping probabilities & element-level scores
- Application to heterogeneous XML collection retrieval

Conclusion & Future Work

Our work showed that:

- Query-element mapping can be estimated by collection statistics
- Incorporating this into retrieval model improves performance

PRMS was also effective in:

- IMDB collection using 1,200 real user queries
- W3C e-mail collection using 125 TREC known-item queries
  - PRMS (MRR:0.617) closely matches best TREC run (MRR:0.621)

Future Work

- Better estimation of mapping probabilities & element-level scores
- Application to heterogeneous XML collection retrieval

References

[OC04] Paul Ogilvie and Jamie Callan.
Hierarchical language models for xml component retrieval.
In *INEX*, pages 224–237, 2004.

[PC98] Jay Ponte and W. Bruce Croft.
A language modeling approach to information retrieval.
In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development Information Retrieval*, pages 275–281. ACM, ACM, 1998.

[PCD09] Desislava Petkova, W. Bruce Croft, and Yanlei Diao.
Refining keyword queries for XML retrieval by combining content and structure.
In Proceedings of ECIR '09. Springer, 2009.

[RZT04] Stephen Robertson, Hugo Zaragoza, and Michael Taylor.
Simple bm25 extension to multiple weighted fields.
In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49, New York, NY, USA, 2004. ACM.