Guess What I Want: Inferring the Semantics of Keyword Queries Using Evidence Theory

Jia-Jian Jiang, Zhi-Hong Deng*, Ning Gao, and Sheng-Long Lv

Key Laboratory of Machine Perception (Ministry of Education),
School of Electronic Engineering and Computer Science, Peking University
{jjj,ninggao}@pku.edu.cn, zhdeng@cis.pku.edu.cn
davidfracs@gmail.com

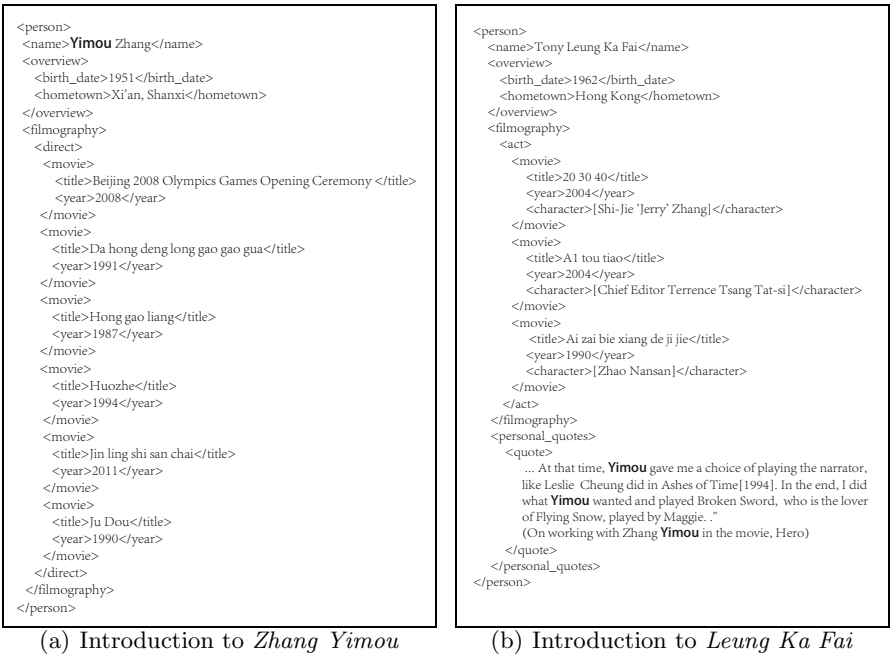
Abstract. The tagged and nested structure of an XML document provides quite detailed information about its structure and semantic, which is neglected by traditional keyword search model like TF-IDF and BM25 etc. Popular XML search models such as SLCA and XRANK tend to return the "deepest" node containing all given keywords, which usually leads to semantic loss. In this paper, we introduce the concept of *belief* in D-S evidential theory to evaluate primary search results, and propose a novel ranking model XSRET to rank them. In XSRET, We utilize XML's rich tag system to predict the semantics of keyword queries. For evaluating our SLCA-E model, we compare it with some state-of-the-art models, such as XSeek and XReal, and experimental result shows that XSRET outperforms these models. In addition, XSRET won the championship in the contest of data-centric track of INEX 2010.

1 Introduction

In traditional search model tf-idf [1][11], documents are defined to be search results, and ranking functions are based on relevance between documents and queries. Tf-idf model focuses on appearance frequency of a keyword while neglecting its semantics, even if it provides different information in different appearances. Therefore, if a keyword appears repeatedly in different documents or in different positions of the same document, their difference will be neglected, even if they actually have different meanings (e.g. *computer virus* and *Biological virus*). Tf-idf model uses only appearance frequency(term frequency and inverse document frequency) of a term to evaluate its weight, which is limited sometimes. For instance, fig.1 shows two result documents of query "Yimou", and document (a) is a segment of introduction to *Yimou Zhang* while document (b) is a segment of introduction to *Leung Ka Fai*. Judged by tf-idf model, document (b) will be a better answer than document (a), since "Yimou" appears three times in document (b) while only once in document (a). However, in fact, document (a) is a better result document to query "Yimou" than document (b). So why is that? Notice, all three appearances of "Yimou" in document (b) is in

* Corresponding author.

Q.Z. Sheng et al. (Eds.): APWeb 2012, LNCS 7235, pp. 388–398, 2012.



(a) Introduction to *Zhang Yimou* (b) Introduction to *Leung Ka Fai*

Fig. 1. Two result documents when query="Yimou"

tag *quote*, that is, it is just mentioned by *Leung Ka Fai* for three times. Thus, tags in XML documents imply the semantic information of content under them, which should not be neglected.

Unstructured data such as HTML is a presentation language and hence cannot capture semantic information. XML data model addresses this limitation by allowing for extendable element tags, which can be arbitrarily nested to capture additional semantics. An XML document is organized by nested tags and the content between them, and tags imply semantics of the content under them. An XML document can be regarded as a tree, in which each node is a tag or content in the original XML document. In [2] [3] [4], nodes in an XML tree is classified into four types, they are entity node, connection node, attribute node and value node, and we continue to use it in this paper.

As discussed above, different appearances of a keyword in different documents or different positions of the same document have different semantics, while it is neglected by traditional research. In this paper, we focus on utilizing rich tags in XML datasets to infer semantics of keyword queries in XML search. Meanwhile, we exploit D-S evidential theory, which is a mathematical theory of evidence, to accomplish it. Our main contributions are summarized as follows:

1) This is the first work that utilizes evidence theory to XML keyword search, and we introduce the concept of *belief* to evaluate search results.

2) We utilize the tag system in XML datasets to infer the semantics of each keyword in given query, and explain it as the belief of given keyword to an attribute.

3) We promote a novel rank model XSRET to rank primary matches of given query, which is based on Dempster's rule of combination in D-S evidential theory.

The rest of the paper is organized as follows. We present preliminary on data model and D-S evidential theory in Section 2. Section 3 infers the semantics of given queries, and section 4 presents our rank model XSRET. Experimental evaluation is given in Section 5 and we conclude in Section 6.

2 Preliminaries

2.1 Data Structure

Xseek[1] classifies nodes in XML documents into four categories: entity, attribute, value, and connection. Among them, an entity can be regarded as a integrated unit that refers to something (a book, a person, a company, etc.), while attribute and value refer to the name and value of an attribute respectively , and connection refers to relationship between two entities (in XML documents, a connection node usually appears at the joint of some homonymous entities). For XML documents that obey a DTD profile, [1] makes the following inferences on node categories.

1) A node represents an entity if it corresponds to a *-node in the DTD.

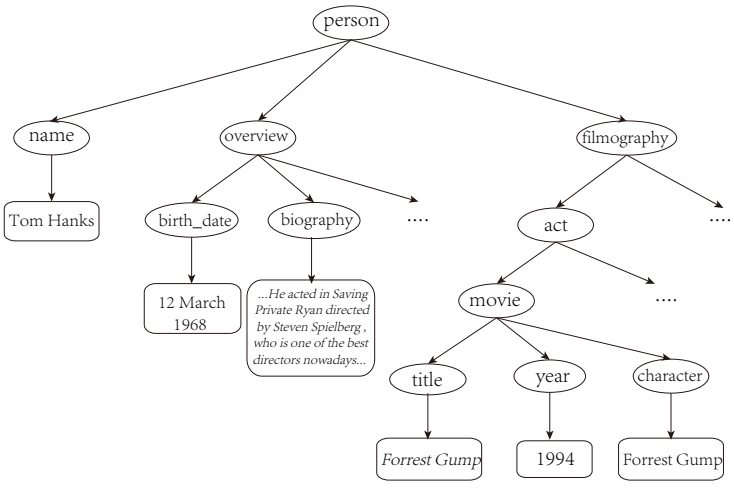
2) A node denotes an attribute if it does not correspond to a *-node, and only has one child, which is a value.

3) A node is a connection node if it represents neither an entity nor an attribute. A connection node can have a child that is an entity, an attribute, or another connection node.

For instance, fig.2 shows the tree-style structure of an XML document, together with a DTD schema it obeys. In fig.2, *person* and *movie* are entities, which have both attributes and entities as their child nodes. *Overview*, *filmography* and *act* are connections, which connect several entities. *name*, *birth_date*, *biography*, *title*, *year*, and *character* are attributes, which have only a value node as their child. All the remaining leaf nodes are values. In this paper, when a keyword k appears in an attribute node A(represented by its name) or its son node(value node), then we say keyword k is in attribute A.

2.2 Search Result Definition

[12][13][14]firstly exploit the statistics of underlying XML database to address search intention identification, and [13] proposes object-level matching semantics called Interested Single Object (ISO) and Interested Related Object (IRO) to capture single object and multiple objects as user's search targets respectively. In this paper, we use the SLCA-based semantic model SLCA-E to define the result.




```
<!ELEMENT person (name, overview?, filmography?)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT overview (birth_date?, biography?)>
<!ELEMENT birth_date (#PCDATA)>
<!ELEMENT biography (#PCDATA)>
<!ELEMENT filmography (act?, direct?, write?)>
<!ELEMENT act (movie+)>
<!ELEMENT direct (movie+)>
<!ELEMENT write (movie+)>
<!ELEMENT movie (title, year, character?)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT character (#PCDATA)>
```

Fig. 2. A segment of tree-style XML document and its DTD Schema

VLCA models tend to return the "deepest" node containing the keywords, which usually leads to semantic loss. For instance, in fig.2, given a query "Tom Hanks", SLCA model returns the leaf node that contains "Tom Hanks" as result, which is confusing. In order to return a meaningful element, SLCA-E restricts the search result to be an entity, which is a fully-semantic unit of real world objects. An entity e is a tree or subtree rooted in an entity node, which is usually the root of a document tree.

Definition 1. *given a list of keywords* $k_1, k_2, , , k_n$ *and an XML tree T, an answer to keywords* $k_1, k_2, , , k_n$ *is an entity e that contains at least one instance of each keyword, meanwhile no other entity below it is an answer entity(of keywords* $k_1, k_2, , , k_n$*).*

[4] Proposed two efficient algorithms to compute the SLCAs of a keyword query. When we compute SLCA-Es of a keyword query we just have to compute its SLCAs first, and then check each element in the set unless it is an entity. If any

element in SLCAs is not an entity node, we will visit its parent and replace the prior element with its parent if its parent is an entity. Otherwise we will visit the parent of its parent repeatedly until we find an entity node.

2.3 D-S Evidential Theory

The Dempster - Shafer evidential theory [10] is a mathematical theory of evidence. It allows one to combine evidence from different sources and arrive at a degree of belief (represented by a belief function) that takes into account the available evidences.

In D-S evidential theory, frame of discernment $\Theta = \{\theta_1, \theta_2, \cdots, \theta_n\}$ is the set representing all possible states of a system under consideration, and $A \in 2^{\Theta}$ is a possible solutions. Evidence theory assigns a belief mass $m : 2^{\Theta} \to [0, 1]$ to each solution, which is called *Basic Probability Assignment(BPA)*, and the belief mass satisfies $m(\emptyset) = 0$ and $\sum_{A \subseteq \Theta} m(A) = 1$.

Furthermore, D-S evidential theory defines belief and plausibility of solution A as follow.

$$Bel(A) = \sum_{B \subseteq A} m(B) \tag{1}$$

$$Pl\,(A) = 1 - Bel\left(\bar{A}\right) = \sum_{B \subseteq \Theta} m(B) - \sum_{B \subseteq \bar{A}} m(B) = \sum_{B \cap A \neq \emptyset} m(B) \tag{2}$$

In D-S evidential theory, $[Bel(A), Pl(A)]$ forms the probability interval of belief on solution A, and particularly, when $\forall A, B \subseteq \Theta, A \bigcap B = \emptyset$, then $Bel(A) = Pl(A)$.

Dempster's Rule of Combination settled the problem how to combine evidence from difference sources. Suppose there exist n evidences and their corresponding mass functions, then their combination is calculated as follow.

$$\begin{cases} m_{\oplus}\,(A) = m_1 \oplus \cdots \oplus m_{\mathrm{n}}\,(A) = K^{-1} \sum\limits_{A_1 \cap \cdots \cap A_n = A} m_1(A_1) \cdots m_n(A_n) \\ K = \sum\limits_{A_1 \cap A_2 \cap \cdots \cap A_n \neq \emptyset} m_1(A_1) m_2(A_2) \cdots m_n(A_n) \end{cases} \tag{3}$$

3 Inferring the Semantics of Keywords in a Given Query

Given a query containing n keywords as follow.

$$Q = \{k_1, k_2, \cdots, k_n\}$$

As each keyword can appear in different attributes, what we focus on is computing belief of all these attributes to that keyword. Suppose that Θ_{k_i} is the frame of discernment of keyword k_i, which contains all attributes in which keyword k_i appears.

$$\forall k_i, \exists \Theta_{k_i} = \{A_{i1}, A_{i2}, \cdots, A_{im}\}$$

here we define every solution to be a set of a single attribute, i.e. $\{A_{ij}\}$, and we use A_{ij} to briefly represent a solution. Thus Θ_{k_i} is also the set of all solutions of keyword k_i. As all solutions of keyword k_i is given, what we have to do next is to compute the belief of each solution(attribute) A_{ij} to keyword k_i. Notice, given any two different attributes A_{ip} and A_{iq} which satisfy $A_{ip} \bigcap A_{iq} = \emptyset$, we have following conclusion.

$$\mathrm{Bel}\left(\mathrm{A_{ij}}\right) = \mathrm{Pl}\left(\mathrm{A_{ij}}\right) \tag{4}$$

Thus, the probability interval of belief on A_{ij} is strict to a point, and we can use $Bel(A_{ij})$ to represent the belief on attribute on attribute A_{ij}.

3.1 Mass Function Based on Frequency and Length

To compute the belief of an attribute to given keyword, we should firstly determine some evidences to evaluate it, together with mass functions based on these evidences. Evidences can be various properties, and in this paper, we choose frequency of attribute containing given keyword and length of attributes to be two evidences.

As in tf-idf model, term frequency gives a measure of the importance of a term t within a particular document d. Similarly, the more keyword k appears in attribute A, the more likely that keyword k is used to describe attribute A. For instance, keywords like *Zhang Yimou* and *Leung Ka Fai* have a rather high distribution in attribute *name* of entity *person*, while *documentary* and *animation* mostly appear in attribute *genre* of entity *movie*. Therefore, we choose the frequency distribution of a keyword in its corresponding attributes to be one evidence, that is the frequency of attribute A containing given keyword. Since mass function m should satisfy $\sum_{A \subseteq \Theta} m(A) = 1$, thus, we define the mass function based on distribution as follow.

$$\mathrm{m_{freq}}\left(\mathrm{A_{ij}}|\mathrm{k_i}\right) = \frac{\mathrm{freq}\left(\mathrm{A_{ij}}, \mathrm{k_i}\right)}{\sum_j \mathrm{freq}\left(\mathrm{A_{ij}}, \mathrm{k_i}\right)} = \frac{\mathrm{freq}\left(\mathrm{A_{ij}}, \mathrm{k_i}\right)}{\mathrm{freq}\left(\mathrm{k_i}\right)} \tag{5}$$

where $\mathrm{freq}\left(\mathrm{A_{ij}}, \mathrm{k_i}\right)$ refers to the frequency of appearance of keyword $\mathrm{k_i}$ in attribute $\mathrm{A_{ij}}$.

In addition to frequency distribution, we choose length of attributes to be another evidence, since frequency distribution sometimes is not enough for determination. As the instance shown in figure 1, keyword "Yimou" appears three times in document (b) while only once in document (a), however, document (a) is a better result than document (b). Although "Yimou" appears three times in document (b), yet it appears in the quote of *Leung Ka Fai* instead of describing the property of him. Therefore, we can conclude that although a document may contain the given keyword, yet the keyword may possibly not used to refer its property.

It is difficult to judge whether a keyword refers to the property of the document containing it, but we can use the evidence of length to infer the probability. As shown in figure 1 and figure 2, there are many attributes in an entity, and some

of them is long text, while some others only contain few keywords. It is obviously that the longer an attribute is, the more redundant information it has, or in other words, the shorter an attribute is, the more explicit its keyword is. We define the mass function based on length of attributes as follow.

$$m_{len}\left(A_{ij}|k_i\right) = \frac{\left(len\left(A_{ij}\right)\right)^{-1}}{\sum_j \left(len\left(A_{ij}\right)\right)^{-1}} \tag{6}$$

where $len\left(\mathrm{A_{ij}}\right)$ is average length of all attribute nodes $\mathrm{A_{ij}}$ that contains keyword k_i.

3.2 Computing the Belief of Attributes

Given mass functions based on frequency and length, we can combine them to compute the belief of attributes based on Dempster's rule of combination as follow.

$$\begin{cases} \mathrm{Bel}\left(\mathrm{A_{ij}}|\mathrm{k_i}\right) = \mathrm{m_{freq}} \oplus \mathrm{m_{len}}\left(\mathrm{A_{ij}}|\mathrm{k_i}\right) = \frac{1}{\Delta_\mathrm{i}} \mathrm{m_{freq}}\left(\mathrm{A_{ij}}|\mathrm{k_i}\right) \times \mathrm{m_{len}}\left(\mathrm{A_{ij}}|\mathrm{k_i}\right) \\ \Delta_\mathrm{i} = \sum_{\mathrm{j}=1}^{\mathrm{m}} \mathrm{m_{freq}}\left(\mathrm{A_{ij}}|\mathrm{k_i}\right) \times \mathrm{m_{len}}\left(\mathrm{A_{ij}}|\mathrm{k_i}\right) \end{cases} \tag{7}$$

For instance, given a keyword "French" and four attributes which contain it. In pre-processing, we can obtain the information about total appearance frequency of "French" as well as attributes containing "French" and their average length. We can judge the belief of the four attributes to "French" as follow.

Table 1. BPA of attributes when keyword="French"

Attributes(A)	$m_{dist}(A|\text{“}French\text{”})$	$m_{len}(A|\text{“}French\text{”})$	$\mathrm{Bel}(A|\text{“}French\text{”})$
language	38986/90196	6/7.8482	0.8809
name	5029/90196	6/13.8515	0.0645
character	12090/90196	6/36.1828	0.0544
biography	2071/90196	6/2300.91	0.0002

4 Computing the Belief of Entities

In our rank model XSRET(XML Search Ranking based on Evidence Theory), we computes the belief of each entity to the given query, and rank the entities based on their beliefs. We define Θ_Q as the frame of discernment of query Q, in which each entity e_i contains all keywords in query Q.

$$\forall Q, \exists \Theta_Q = E = \{e_1, e_2, \cdots, e_z\}$$

As queries are made up of several keywords, we compute the belief of each entity based on the belief of appearances of keywords in it. in this paper, we treat each keyword as an evidence, and define mass functions based on appearances of keywords in the entity. We define the belief of entity e_t to query Q as follow.

$$\begin{cases} \mathrm{Bel}\left(\mathrm{e_t}|\mathrm{Q}\right) = \mathrm{m_{k_1}} \oplus \mathrm{m_{k_2}} \oplus \cdots \oplus \mathrm{m_{k_n}}\left(\mathrm{e_t}|\mathrm{Q}\right) = \frac{1}{\Lambda_\mathrm{t}} \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{m_{k_i}}\left(\mathrm{e_t}|\mathrm{Q}\right) \\ \Lambda_\mathrm{t} = \sum_{\mathrm{t}=1}^{\mathrm{z}} \left(\prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{m_{k_i}}\left(\mathrm{e_t}|\mathrm{Q}\right) \right) \end{cases} \tag{8}$$

where $m_{k_i}(e_t|Q)$ is the belief of entity e_t to query Q when considering keyword k_i, and we define it as follow.

$$m_{k_i}(e_t|Q) = \sum_{A_{ij} \in e_t \wedge k_i \in A_{ij}} Bel(A_{ij}|k_i) \quad (9)$$

For instance, when query Q={USA, president, documentary}, to compute the belief of all entities that contain all three keywords, we firstly compute the sum of belief of all attributes to each keyword, and then add them together.

Table 2. BPA of documents when query={USA, president, documentary}

Entities(e_i)	$m_{\text{“}USA\text{”}}(e_i\|Q)$	$m_{\text{“}president\text{”}}(e_i\|Q)$	$m_{\text{“}documentry\text{”}}(e_i\|Q)$	$Bel(e_i\|Q)$
e_1	$Bel(country\|\text{“}USA\text{”})$ =0.4853	$2Bel(title\|\text{“}president\text{”})+$ $Bel(keyword\|\text{”}president\text{”})$ =1.6948	$Bel(genre\|\text{“}documentary\text{”})$ =0.6915	0.99635
e_2	$Bel(country\|\text{“}USA\text{”})$ =0.4853	$Bel(trivia\|\text{“}president\text{”})$ =0.0062	$Bel(genre\|\text{“}documentary\text{”})$ =0.6915	0.00364
e_3	$Bel(plot\|\text{“}USA\text{”})$ =0.0004	$Bel(plot\|\text{“}president\text{”})$ =0.0147	$Bel(genre\|\text{“}documentary\text{”})$ =0.6915	0.00001

In table 2, e_1 is a movie named *Portraits of Presidents: Presidents of a World Power*, which is a documentary movie mentioned about tens of American presidents, while e_2 is a movie named *Surviving the Eruption at Mt. Pinatubo* and e_3 is a movie named *Transpersonal Conversations: Frances Vaughan, Ph.D* , both of which are obviously irrelevant. The BPA of these three entities evaluate them as expected. The overall processing of our model XSRET is shown in fig.3.

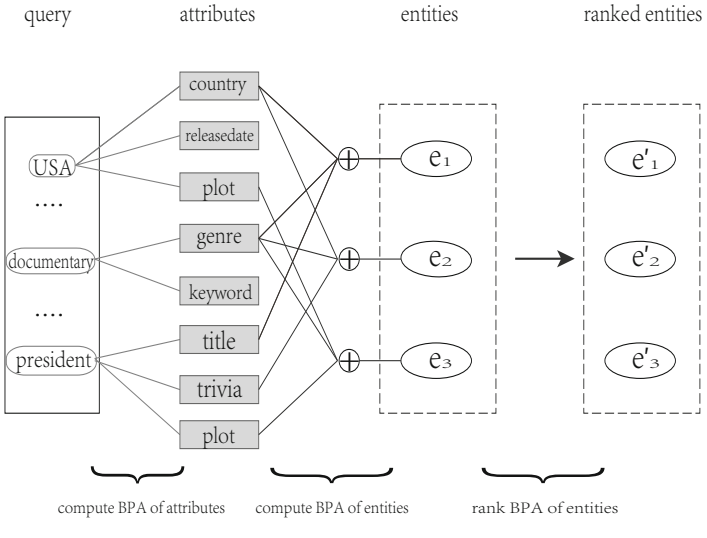


Fig. 3. Procedure example of XSRET model

5 Experiment

5.1 Datasets

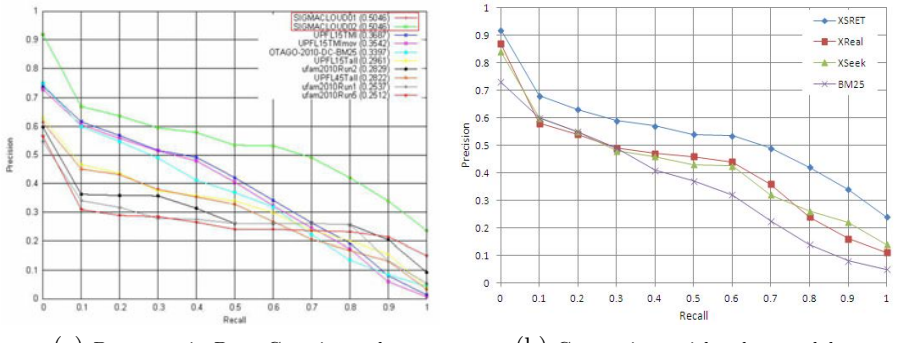
In this paper, we use the IMDB data collection from INEX 2011 Data-Centric track, which is newly built from www.imdb.com. It consists of information about more than 1,590,000 movies and people involved in movies, e.g. actors/actresses, directors, producers and so on. Each object is richly structured. The total size of these files is 21.9GB, and figure 2 shows a segment of these files.

5.2 Results

We tested on all of the 28 queries from Data-Centric track topics presented in table 3(we neglected symbols like "", + and - here). Limited by space, we

Table 3. 28 topics in Data-Centric track of INEX 2010

Q_1	Yimou Zhang 2010 2009	Q_{15}	may the force be with you
Q_2	Dogme movies	Q_{16}	best director award Steven Spielberg
Q_3	stan lee actor	Q_{17}	Avatar James Francis Cameron
Q_4	brad pitt producer	Q_{18}	Stanley Kubrick movies director
Q_5	Shirley Temple	Q_{19}	Heath Ledger actor movies list
Q_6	Tom Hanks Ryan	Q_{20}	Movies directed Jean Pierre Jeunet Marc Caro
Q_7	Ingmar Bergman biography	Q_{21}	Movies Klaus Kinski actor movies good rating
Q_8	titanic jack rose	Q_{22}	Scarlett Johansson John Slattery
Q_9	hua mulan animation	Q_{23}	Comedy Woody Allen Scarlett Johansson
Q_{10}	director fearless jet li	Q_{24}	around the world in eighty days
Q_{11}	ancient Rome era	Q_{25}	tom hanks steven spielberg
Q_{12}	quentin tarantino thriller	Q_{26}	true story drugs addiction
Q_{13}	tom cruise movies	Q_{27}	making of The Lord of the Rings
Q_{14}	Clint Mansell composer	Q_{28}	romance movies by Richard Gere or George Clooney



(a) Best runs in Data-Centric track (b) Comparison with other models

Fig. 4. Experiment results measured by MAP

briefly present the experimental results, and fig.4 shows the comparison result of our mode XSRET with other methods. In fig.4, (a) presents the performance of the whole document based MAP metric, and XSRET (SIGMACLOUD01 and SIGMACLOUD02) performs substantially better than the other runs at all recall points. In addition, we compare XSRET with latest models XReal and XSeek and present the result in fig.4(b), and XSRET gains a better result.

6 Conclusions

In this paper, we have a primary discussion about evaluating search results in XML keyword search based on D-S evidential theory. We utilize XML's rich tag system and structure information to infer the semantics of keyword queries, and we focus on analysis of each appearance of keyword in results. Experiments show that our rank model XSRET is more effective than the existing approaches.

Acknowledgement. This work is partially supported by Project 61170091 supported by National Natural Science Foundation of China and Project 2009AA01Z136 supported by the National High Technology Research and Development Program of China (863 Program).

References

1. Carmel, D., Maarek, Y.S., Mandelbrod, M., et al.: Searching XML documents via XML fragments. In: SIGIR 2003, pp. 151–158 (2003)
2. Liu, Z., Chen, Y.: Identifying meaningful return information for xml keyword search. In: SIGMOD Conference (2007)
3. Liu, Z., Walker, J., Chen, Y.: XSeek: A Semantic XML Search Engine Using Keywords. In: VLDB 2007, pp. 1330–1333 (2007)
4. Huang, Y., Liu, Z., Chen, Y.: eXtract: A Snippet Generation System for XML Search. In: VLDB 2008, pp. 1392–1395 (2008)
5. Xu, Y., Papakonstantinou, Y.: Efficient keyword search for smallest LCAs in XML databases. In: SIGMOD, pp. 537–538 (2005)
6. Guo, L., Shao, F., Botev, C., Shanmugasundaram, J.: XRANK: ranked keyword search over XML documents. In: SIGMOD (2003)
7. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a Core of Semantic Knowledge. In: WWW (2007)
8. Suchanek, F., Kasneci, G., Weikum, G.: YAGO: A Large Ontology from Wikipedia and WordNet. Journal of Web Semantics (2008)
9. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: A Nucleus for a Web of Open Data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
10. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976) ISBN 0-608-02508-9

11. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York (1986)
12. Bao, Z., Lu, J., Ling, T.W.: XReal: An Interactive XML Keyword Searching. Demo paper in CIKM (2010)
13. Bao, Z., Lu, J., Ling, T.W., Xu, L., Wu, H.: An Effective Object-Level XML Keyword Search. In: Kitagawa, H., Ishikawa, Y., Li, Q., Watanabe, C. (eds.) DASFAA 2010. LNCS, vol. 5981, pp. 93–109. Springer, Heidelberg (2010)
14. Bao, Z., Ling, T.W., Chen, B., Lu, J.: Effective XML Keyword Search with Relevance Oriented Ranking. Full paper in ICDE (2009)