A Schema-Driven Approach for Knowledge-Oriented Retrieval and Query Formulation

Hany Azzam, Sirvan Yahyaei, Marco Bonzanini, and Thomas Roelleke*
Queen Mary, University of London, UK. E1 4NS
{hany,sirvan,marcob,thor}@eecs.qmul.ac.uk

ABSTRACT

In order to search across factual knowledge and content explicated using different data formats this paper leverages a generic data model (schema) that transforms keyword-based retrieval models and queries to *knowledge-oriented* models and *semantically-expressive* queries. As each of the transformed retrieval models capitalises on a specific evidence space (term, classification, relationship and attribute), we demonstrate two possible combinations of these spaces, namely macro-based or micro-based. For bare keyword-based queries we demonstrate how the data model can be used to augment the queries with classifications, relationships, etc. that reflect the underlying constraints and objects found in the heterogeneous knowledge bases. Using the IMDb benchmark the results demonstrate the feasibility and effectiveness of the instantiated retrieval models and the query reformulation process.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Search and Retrieval**]: Retrieval Models

General Terms

Design, Theory, Algorithms

Keywords

Knowledge Representation, Semantic Retrieval, DB+IR

1. INTRODUCTION

Information extraction techniques such as those mentioned in [31] have been successfully applied to textual and semi-structured Web sources like Wikipedia to build large-scale "knowledge bases" such as YAGO [35]. These knowledge bases typically contain entities (e.g. people, locations, movies, etc.) and relationships (e.g. bornIn, actedIn, hasGenre, etc.).

*This research was conducted while the author was a Visiting Scientist at Yahoo Research Barcelona.

KEYS'12, May 20, 2012, Scottsdale, Arizona, USA.

These automatically generated knowledge bases can be integrated with content-based resources and used to develop a more *semantic-aware* search experience that affects query representation, retrieval models and result presentation [36]. For example, with the help of semantic annotations, more semantic-aware variants of the traditional text-only retrieval models can be developed to directly retrieve objects such as actors and locations instead of just documents or document elements [5].

The search process therefore becomes more diverse and complex as entities and relationships explicated by the knowledge bases are considered, in addition to text. However, the increasingly broad range of information extraction tools (and outputs) such as ASSERT [24], microformats such "geo" and "hAtom" and RDF/RDFa markup, poses several considerations to the search process [20, 8]. Firstly, when a new data format is introduced, it needs to be *quickly* integrated into a standard representation and exploited alongside the existing formats. Secondly, these tools and outputs alongside the data representation need to be exploited for retrieval. Specifically, if there is a representation that combines different types of data including content, how can it be searched *effectively*? Thirdly, a facility, preferably (semi-)automatic, is required so that the formulated keyword-based queries which search across the heterogeneous knowledge bases are *semantically-expressive*, i.e. the queries reflect the underlying objects, attributes, relationships, etc. [15, 2, 11].

This paper attempts to address these challenges. It utilises a *general purpose* data model (schema) to represent facts (e.g. entities and their relationships) and content knowledge (e.g. text in documents) in one congruent framework [3]. The data model guides the process of integrating different types of knowledge from different data sources. That is, once populated with data, the schema provides a facility to *quickly* create mashups by eschewing syntactical constraints and before further data cleansing takes place at application-specific layers.

To address the second challenge, the paper demonstrates a family of retrieval models that draw upon this data model and that can *effectively* exploit the consolidated data. The tight coupling between the data model and the retrieval models leads to models that support constraint-checking and ranking with respect to classifications, relationships and attributes, as well as terms (text). As such, the output of information extraction tools and different markups can be instantly incorporated into the retrieval process and reasoned upon.

With regard to query formulation, we extend the methods proposed by [11, 17] to the knowledge representation used herein. Instead of reformulating the query according to a particular data format (e.g. XML and RDF), we propose a more conceptual approach. Similar to how the retrieval models are instantiated, the query reformulation process is coupled with the data model. This enables

the query to be reformulated and enhanced with classifications, relationships and attributes *independently* from the data format.

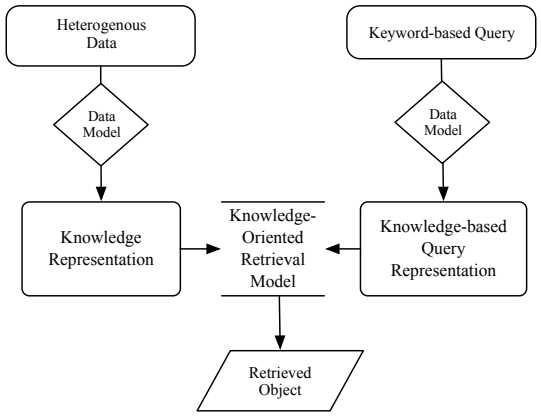


Figure 1: A Schema-Driven Approach for Knowledge-Oriented Retrieval and Query Formulation

Figure 1 demonstrates the schema-driven search process with the data model (schema) being the the core component. The schema relates "data" and "keyword-based query" to "knowledge" and "knowledge-based query" representation. On one side of the figure the explicated factual knowledge is mapped to the data model and coupled with the content resources. On the other side, the data model transforms the keyword-based query to a semantically expressive query. This can be performed either manually as the user expresses the query using the underlying semantic information or automatically through a mapping process which maps a query term to its corresponding class, relationship or attribute. The two representations (knowledge and query representation) are matched by the knowledge-oriented retrieval models that support constraint-checking and ranking with respect to the underlying data model and its components (classifications, relationships, attributes and terms).

The semantic information which is used in this paper to showcase the knowledge-oriented retrieval models and queries is primarily explicated using XML and a shallow parser (see Figure 2). However, since these models and queries are instantiated using a schema, they are independent of the underlying physical data representation. Thus, other data formats such as microformats and RDF can be incorporated into the aforementioned search process.

We hypothesise that this stepwise process, which consists of representing knowledge, constructing the retrieval models, reformulating the queries and obtaining the results, can improve keyword-based search over heterogeneous knowledge bases.

The remainder of the paper is organised as follows. Section 2 outlines the related work. Section 3 discusses the data model and demonstrates how the data is represented. Section 4 provides the definition of the keyword-based retrieval model and presents its semantic variants which exploit the underlying representation. Section 5 demonstrates how keyword-based queries are transformed to more semantically-expressive queries. Our hypothesis is supported by Section 6, which discusses the results of the experiments. A summary of the paper and contributions can be found in Section 7.

2. RELATED WORK

The proposed knowledge-oriented retrieval models are akin to retrieval models used in structured (XML) document retrieval in that they are based on combining different evidence spaces such as sections (section-frequency of a term) and titles, in addition to the usual spaces, namely documents (document-frequency). These different spaces can be combined in numerous ways. One can differentiate between two types of models: variants of the so-called probabilistic model (BM25), such as [27] and [18]; and variants of language modelling [23], such as [33] and [22]. Moreover, one can distinguish between models that aggregate evidence about terms and models that aggregate scores (e.g. passage retrieval [7]).

In semantic retrieval, structural elements can be utilised to estimate the relevance of an object. Unlike in structured document retrieval, structural elements bear a semantic meaning (e.g. actor and team in Figure 2). For example, [17] extend [22] to demonstrate how varying weights for different semantic elements across query terms can improve retrieval performance. Evidence associated with semantic annotations in the form of linguistic structures is also used in semantic retrieval, especially in question answering applications. [39] and [5] propose ranking answer-bearing sentences to questions by incorporating the semantic annotations in both the sentences and queries into the retrieval process. In both, the semantic annotations are comprised of only "semantic predicates".

Semantic annotations expressed in the form of Resource Description Framework (RDF) graphs, referred to as entity-relationship (ER) graphs, are also used as ranking criteria. These graphs are used as a source of evidence to construct graph-based ranking models and queries for retrieving semantic objects. For example, BANKS [4] covers important issues with respect to graph search and ranked retrieval on graphs. Nevertheless, the approach does not fully exploit the inherent semantics of entities and explicit relationships in the ER graphs as in [16] and [11]. These two approaches propose language modelling variants for ranking the results of keyword-augmented graph-pattern queries over ER graphs and make effective use of the explicit nature of the ER graphs. However, the proposed variants *only* utilise the annotated entities and relationships to answer the queries and do not combine other sources of evidence such as full-text and/or structural elements. In other words, one cannot incorporate context, such as term occurrences in documents, during ranking. Additionally, the schema-driven approach used in this paper provides the means to instantiate *any* probabilistic retrieval model and does not argue for a tailored model or query language. Using different retrieval models without having to change the external application or the query language maintains the desired flexibility when implementing knowledge-oriented retrieval systems. Another approach that utilises graph-based evidence for semantic ranking is [38]. This approach exploits evidence from "entity containment graphs" and a web search engine to compute the relevance of different entities. However, it is only applicable to entity ranking.

Other related approaches include [9] and [14]. Both propose using database-like technology to implement retrieval models. The former approach presents an array data model to abstract away from the physical details of retrieval. The latter approach integrates region algebra and language modelling for structured document retrieval into a framework that allows developers to define complex models for different applications.

More database-oriented approaches include custom-built relational systems for storing semantic data such as triple stores [13, 21], property table stores [37] and horizontal stores [1, 34]. However, independent benchmarking projects [6, 32] have shown that no system dominates and that there remains room for developing new techniques for storing and querying semantic data.

The concepts employed herein build upon some of the aforementioned research and theoretical foundations laid by [10] and [25]. The contribution of this paper is to leverage such concepts for knowledge-oriented retrieval and query formulation.

3. KNOWLEDGE REPRESENTATION

This section describes the data and the representation behind the knowledge-oriented retrieval models.

```
<movie id="329191" title="Gladiator" year="2000" genre="action">
    <actor> Russell Crowe </actor>
    <team>  Ridley Scott </team>
    <plot> When a [ARG1 general] is [TARGET betrayed by ] a
                [ARG0 prince] ... </plot>
</movie>
```

Figure 2: An Example of an IMDb movie and its semantic structures identified using XML and ASSERT

The XML-based representation and the semantic annotations generated using a shallow parser are illustrated in Figure 2. To represent the explicated factual knowledge and content, we utilise the relational implementation of the Probabilistic *Object-Relational* Content Model [3]. The relational schema, referred to as the Probabilistic *Object-Relational* Content Model, consists of the following relations: "term(Term,Context)", "term_doc(Term, Context)", "classification(ClassName, Object, Context)", "relationship(RelshipName, Subject, Object, Context)" and "attribute(AttrName, Object, Value, Context)". These relations are more generally referred to as *propositions* and Term, ClassName, RelshipName and AttrName are referred to as *predicates* (a specification originating from terminological logics [19]).

The term-based relations ("term" and "term_doc"), depicted in Figure 3, model term-oriented representations, which are commonly used in traditional IR application. The "term" relation stores the parsed text and the context within which the text was found. The context represents the location information, which in this case is stored as a path, expressed in XPath[1]. Figure 3 also shows that the context can be expressed using Uri's such as "russell_crowe" and "prince_241" (for presentation purposes we only show the URN of the uniform resource identifier). However, since the example and the evaluation benchmark are based on XML, we primarily express the context using XPath. The relation "term_doc" is derived from the "term" relation. This relation maintains only the root context of each term-element pair, which helps to propagate the content knowledge found in the children contexts to the parent.

The classification, relationship and attribute relations represent object-class, subject-object and object-value associations. These relations are necessary to model the object-oriented concepts and factual knowledge in general. Figure 4 shows the transition from the standard object-relational model (ORM) towards the object-relational *content* model (ORCM). To indicate the wider applicability of the schema-driven approach, the schema design step includes special relations to model aggregation ("part_of") and inheritance ("is_a"). Further discussion of these relations is beyond the scope of this paper. In similar approaches, such as terminological logic [19], there are only concepts (classes) and roles (relationships), and these are monadic and dyadic predicates, respectively. The ORCM schema, however, adds "terms", and the attribute "context". As such, content is viewed as a separate concept from the concepts of traditional object-oriented modelling which helps to describe it in a more formal and knowledge-oriented way.

On the information need side, the data model can enrich query representation. This facilitates the expression of more complex and expressive representations of information needs. As such, the matching process is transformed from a one-step matching process into a multistep matching process that joins knowledge representations with the enriched query representations. This results in a more powerful and complex matching process that truly exploits different types of evidence to achieve more effective search. The method to automatically transfer a keyword-based query to a semantically-expressive one is discussed in Section 5.

[1]http://www.w3.org/TR/xpath

term	
Term	Context
gladiator	329191/title[1]
2000	329191/year[1]
russell	329191/actor[1]
roman	329191/plot[1]
...	...

(a) Term propositions in element contexts

term_doc	
Term	Context
gladiator	329191
2000	329191
russell	329191
roman	329191
...	...

(b) Term propositions in root contexts

classification		
ClassName	Object	Context
actor	russell_crowe	329191
prince	prince_241	329191
...	...	...

(c) Classification propositions in root contexts

relationship			
RelshipName	Subject	Object	Context
betrayedBy	general_13	prince_241	329191/plot[1]
...	...	...	...

(d) Relationship propositions in element contexts

attribute			
AttrName	Object	Value	Context
title	329191/title[1]	"Gladiator"	329191
year	329191/year[1]	"2000"	329191
...	...	...	...

(e) Attribute propositions in root contexts

Figure 3: The Probabilistic Object-Relational Content Model Representing a Movie: Ellipses indicate that some data have been omitted. The location where different elements occur are stored as paths, expressed in XPath. For readability we use a simplified syntax, e.g., "329191/title[1]" points to the attribute describing a movie's title.

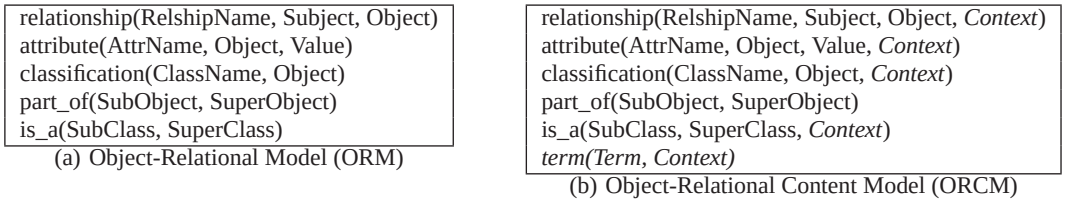
4. KNOWLEDGE-ORIENTED RETRIEVAL MODELS

This section introduces the instantiated retrieval models based on the schema. Each instantiation is based on one of the schema's pillars (term, classification, relationships and attributes). We first demonstrate the traditional *term-oriented* TF-IDF retrieval model and then discuss how this keyword-based model is transformed using the schema to a semantically-aware retrieval model.

4.1 Term-based Retrieval

DEFINITION 1. *TF-IDF* RSV*:*
The TF-IDF term weight is a combination of TF and IDF values.

$$w_{\text{TF-IDF}}(t,d,q) := \text{TF}(t,d) \cdot \text{TF}(t,q) \cdot \text{IDF}(t) \qquad (1)$$



(a) Object-Relational Model (ORM)

(b) Object-Relational Content Model (ORCM)

Figure 4: Schema Design Step: From the Object-Relational Model to the Object-Relational Content Model

The retrieval status value (RSV) is the sum over the TF-IDF weights.

$$\mathrm{RSV}_{\textit{TF-IDF}}(d,q) := \sum_{t \in d \cap q} w_{\textit{TF-IDF}}(t,d,q) \quad (2)$$

$\mathrm{TF}(t,d)$ is the within-document term frequency component, and it can be set to one of the following: the total frequency $\mathrm{tf}_d := n_L(t,d)$ where $n_L(t,d)$ is the number of *locations* (hence subscript L) in a document d at which term t occurs; the BM25-motivated quantification $\mathrm{tf}_d/(\mathrm{tf}_d + K_d)$ where K_d is a normalisation factor reflecting the document length and is usually proportional to the pivoted document length ($\mathrm{pivdl} := \mathrm{dl} \,/\, \mathrm{avgdl}$), which is higher for long documents than for shorter ones.

$\mathrm{IDF}(t)$ is the inverse document frequency component. This can be the negative logarithm of the term probability $P_D(t|c) := n_D(t,c)/N_D(c)$. Here, $n_D(t,c)$ is the number of *Documents* in which term t occurs in collection c, usually denoted as document frequency $\mathrm{df}(t,c) := n_D(t,c)$. $N_D(c)$ is the total number of *Documents*. $\mathrm{IDF}(t)$ can also be normalised, for example, $\mathrm{idf}(t)/\mathrm{maxidf}$. This corresponds to the logarithm to base $N_D(c)$, since usually the maximum idf value is $\mathrm{maxidf} := -\log 1/N_D(c)$. The normalised IDF is also referred to as the probability of "being informative", where the probability that a term will occur is equal to the probability that it will not be informative in maxidf trials [28].

The definition is based on the one given in [30] and some of its probabilistic and information-theoretic interpretations proposed in [28, 26]. The setting of the $\mathrm{TF}(t,d)$ to the BM25-motivated quantification and the probabilistic interpretation of the IDF-based estimates are the settings used for the experiments in Section 6.

4.2 Basic Semantic Retrieval Models

The following definition captures the general form of the knowledge-oriented retrieval models.

DEFINITION 2. *General Form of the Retrieval Models: Let X be a predicate type, i.e. $X := T$ for terms, $X := C$ for class names, $X := R$ for relationship names and $X := A$ for attribute names. Let d be a document*[2] *and q be a query and both contain terms, class names, relationship names, etc.*

$$\mathrm{RSV}_{\textit{X-Model-pred}}(d,q) := \sum_{x \in X(d \cap q)} w_{\textit{X-Model}}(x,d,q)$$

The definition builds on $w_{\text{TF-IDF}}$ (Equation 2). The following instantiations illustrate how the general form is specialised with respect to the predicate type, which results in term, class, relationship and attribute-based models: [TCRA]-IDF. For a readable formulation, we allow type-aware functions, e.g. $\mathrm{IDF}(t)$ is IDF over Terms, $\mathrm{IDF}(a)$ is IDF over Attributes, etc., and similarly $\mathrm{AF}(a,d)$ corresponds to attribute frequency, $\mathrm{CF}(c,d)$ is class frequency, etc.

DEFINITION 3. *[TCRA]F-IDF Retrieval Models:*

$$\mathrm{RSV}_{\textit{TF-IDF}}(d,q) := \sum_{t \in T(d \cap q)} \mathrm{TF}(t,d) \cdot \mathrm{TF}(t,q) \cdot \mathrm{IDF}(t) \quad (3)$$

$$\mathrm{RSV}_{\textit{CF-IDF}}(d,q) := \sum_{c \in C(d \cap q)} \mathrm{CF}(c,d) \cdot \mathrm{CF}(c,q) \cdot \mathrm{IDF}(c) \quad (4)$$

$$\mathrm{RSV}_{\textit{RF-IDF}}(d,q) := \sum_{r \in R(d \cap q)} \mathrm{RF}(r,d) \cdot \mathrm{RF}(r,q) \cdot \mathrm{IDF}(r) \quad (5)$$

$$\mathrm{RSV}_{\textit{AF-IDF}}(d,q) := \sum_{a \in A(d \cap q)} \mathrm{AF}(a,d) \cdot \mathrm{AF}(a,q) \cdot \mathrm{IDF}(a) \quad (6)$$

The term-, classification-, relationship- and attribute-based models are referred to as basic models because they are based on one of the predicate types. Models that combine the RSV's of models are referred to as macro models, and models that combine parameters on the level of predicates are referred to as micro models (both are discussed in the following sections).

Other instantiations based on the general form that are specialised with respect to *propositions* as opposed to predicate types are possible. The proposition-based retrieval models are identical in form to the predicate-based ones. However, the frequency counts of "full propositions" is the statistical evidence used rather than the frequency counts of the predicate names. For example, in *predicate-based* classification retrieval we count the number of times an object X is classified as an "actor"; however, in *proposition-based* classification retrieval the number of times the object "russell_crowe" is classified as an "actor" is counted. In this paper, we only demonstrate the predicate-based retrieval models.

TF-IDF is used in this paper because the results suggest that in the case of the IMDb collection the retrieval performance of TF-IDF with the special setting of $\mathrm{TF}(t,d)$ to the BM25-motivated quantification is quite similar to the performance of the BM25 retrieval model. This does not exclude the fact that an attribute-, class-, relationship-based BM25 and language modelling (LM) can be instantiated from the schema [3]. However, in the case of BM25, for example, we will need to tune the b and $k1$ parameters for each predicate type and then re-tune for each possible combination of the predicates which results in a very large parameter space. The TF-IDF, on the other hand, does not require any parameter tuning which makes it a more appealing candidate to showcase the feasibility of our approach.

4.3 Combined Semantic Retrieval Models

This section introduces two possible approaches to combining the basic semantic retrieval models.

4.3.1 Macro Models

[2]Here d represents the conventional notion of the context as a document. However, the context can be a local passage, a movie, a database tuple, etc. See Figure 3 for examples of different contexts.

Macro models are additive models, where the ranking is modelled by estimating the score of each basic predicate-based model independently and then combining the scores in a suitable way. The combination is performed using a weighted linear addition.

The following set of definitions illustrates the canonical form of the macro models. X is a set of predicates (term, class name, relationship name and attribute name), and w_X denotes the weighting parameter for each predicate.

DEFINITION 4. *XF-IDF Macro Model:*

$$\mathrm{RSV}_{\textit{XF-IDF-macro}}(d,q) := \sum_{X \in \{T,C,R,A\}} w_X \cdot \mathrm{RSV}_{\textit{XF-IDF}}(d,q) \quad (7)$$

The retrieval process consists of the following steps:

1. For each query term a ranked list of corresponding mappings (semantic predicates such as classes, attributes and relationships) is produced, e.g. query term "brad" is mapped to "actor" (see Section 5 for further details).
2. The document space is determined by selecting all the documents that contain at least one query term.
3. For all $X \in \{T, C, R, A\}$ the score of each document is calculated and the total score is computed using Definition 4. The weights of the mappings are used as the query weights in Equation 4 ($\mathrm{CF}(c,q)$), Equation 5 ($\mathrm{RF}(r,q)$) and Equation 6 ($\mathrm{AF}(a,q)$).

The following provides an example of how macro models are leveraged. A query such as "*action movie about a general who is betrayed by a prince*" can be logically formulated in terms of the used knowledge representation as follows:

```
# action general prince betray
?− movie(M) & M.genre("action") &
  M[ general(X) & prince(Y) & X.betrayedBy(Y) ];
```

This formulation is based on the Probabilistic Object-Oriented Logic (POOL) proposed in [29, 12]. To process the query in a macro-based way, for each query term (e.g. action) a term-based retrieval model is used to estimate the initial probabilities. Next, the corresponding predicate (e.g. M.genre("action")) re-ranks the initial set of results using the appropriate basic semantic retrieval model (e.g. attribute-based). For example, for query terms, "general prince betray", and their corresponding predicates, M[general(X)], M[prince(Y)] and M[X.betrayedBy(Y)], the class-based and relationship-based models are used to perform the re-ranking.

4.3.2 Micro Models

There are certainly several possible ways to combine predicate spaces in a micro way. A simple way to construct the joined space is to unite all the predicates (attribute names, relationship names, class names and terms) into one single non-normalised relation. Afterwards, query to document matching can take place and probabilities and frequencies can be estimated and aggregated. Another approach is for the micro models to first estimate the probabilities for *each* query term and its corresponding predicate and *then* to combine the weights. The combination of the scores is similar to the macro model in Definition 4. However, the probability estimation in Equations 4, 5 and 6 is constrained by the result of the mapping process.

Similar to the retrieval process for the macro models, each query term is mapped to classes, attributes and relationships. For example, a weighted list of classes is produced for each query term that indicates the probability of the query term belonging to a particular classification. The document space is determined by selecting all the documents that contain at least one of the query terms. To calculate the total score for each document, the score for each predicate type is estimated. Unlike the macro model, in micro models the document space for each predicate type is constrained by the results of the mapping process. For example, where a particular term is mapped to a particular classification, only documents that contain this classification are considered and for the other documents the weight of the term is zero. For a term that is mapped to a classification, an attribute or a relationship, the documents that contain that relation are *boosted* in proportion to the mapping weight and predicate score of the term in those documents.

5. QUERY FORMULATION

This section describes how the mappings between the query terms and the predicates are deduced. In particular, we demonstrate how keyword-based queries are mapped using the schema to more semantically-expressive queries.

To exploit the proposed retrieval models, while ensuring that users can express their information needs using keyword-based queries, we use the schema to develop a query reformulation process that automatically enriches queries with semantic constraints. This process, which is inspired by [17], generates semantically-expressive queries without the need for manual query formulation.

5.1 Class and Attribute Names

In the case of class- and attribute-based retrieval models, we map each query term to the top-k corresponding class or attribute names (element types). For example, for a query such as "fight brad pitt" and assuming a sample data formatted as

```
<movie><title>Fight ... </ title ><actor>Brad Pitt </ actor></movie>
```

the inferred top-1 attribute/class name would be "title" for query term "fight" and "actor" for query terms "brad" and "pitt". These class and attribute names are instantly pulled out of the index. The probability of the mapping between a query term and an class/attribute name is estimated using the number of mappings between a term and a class/attribute name divided by the total number of mappings in the index. The intuition behind the mapping is that if a term occurs frequently within a certain element type then the term is more likely to be "characterised" by that particular type [17].

We manually classified all the terms of the 40 queries used in the experiments according to the available classes and attributes in the collection and evaluated the mapping process for these queries. In the class mapping, top-1, top-2 and top-3 mappings achieved 72%, 90% and 100% accuracy, respectively. In the attribute mapping, 90% and 100% accuracy was achieved by selecting top-1 and top-2 mappings.

5.2 Relationship Names

A mapping method is also used for the relationship-based retrieval model. Given a query term, the mapping process infers whether a term is a predicate ("RelshipName") or a subject/object of a particular predicate. If the term is mapped to a predicate, then that predicate constitutes one of the mappings. However, if the term is mapped to a subject/object then we determine the corresponding predicate for that particular subject/object.

In greater detail, the probability (weight) of a term being mapped to predicate/subject/object is based on the frequency of each term within a particular context (e.g. the "plot" field). This probability is calculated using the "relationship" relation (Figure 3). For example, the term "betrayed by" occurs frequently in the context plot

					MAP	Diff %
TF-IDF Baseline (Section 4.1)					46.88	-

	w_{Term}	$w_{\text{ClassName}}$	$w_{\text{RelshipName}}$	w_{AttrName}		
XF-IDF Macro Model (Section 4.3.1)	0.4	0.1	0.1	0.4	47.36	+01.02%
	0.5	0.5	0	0	38.13	-18.66%
	0.5	0	0	0.5	**57.98**†	+23.67%
	0.5	0	0.5	0	46.81	-0.001%

	w_{Term}	$w_{\text{ClassName}}$	$w_{\text{RelshipName}}$	w_{AttrName}		
XF-IDF Micro Model (Section 4.3.2)	0.5	0.2	0	0.3	53.74	+14.63%
	0.5	0.5	0	0	43.98	-06.18%
	0.5	*0*	*0*	*0.5*	*53.88*†	*+14.93*%
	0.5	0	0.5	0	46.88	±0%

Table 1: The best model in each combination is *italicised*; best overall model is in bold; results that are statistically significant above the baseline (p < 0.05) are denoted by †, as determined by a signed t-test.

as a predicate, i.e. the relationship name ("RelshipName"). Consequently, the term "betrayed by" is most likely a relationship name as opposed to being a subject or an object. On the other hand, if the probability of a term being a relationship name is *lower* than it being a subject or an object, then the query term is associated with the most frequent predicate(s) that occurs with this subject or object. For example, in the plot in Figure 2 the query term "general_13" is a subject, and the most frequent predicate associated with it is "betrayed by". Overall, the result of the relationship names mapping is a list of weighted predicates which are associated with a subject or an object. The query term and its top-k mappings are used for relationship-based retrieval.

6. EXPERIMENTAL RESULTS

The purpose of the evaluation is to investigate how well the instantiated knowledge-oriented retrieval models perform and to assess the feasibility of the query reformulation method. The results show the effectiveness of the instantiated models compared to keyword-only retrieval models.

6.1 Setup

The IMDb dataset was constructed from text data (http://www.imdb.com/interfaces#plain) and is formatted in XML. Each document corresponds to a movie. The element types are "title", "year", "releasedate", "language", "genre", "country", "location", "colorinfo", "actor", "team" and "plot".

There are several processing steps that present the data in the form that is best suited to achieve effective retrieval. For example, we have opted to propagate the keywords that occur within elements such as "actor" and "team" upwards to their corresponding part. Having a coarser schema helps to improve the accuracy of the derived mappings for each query term (see Section 5). Another reason for a coarser schema is that the terms that occur in a specific context have also been propagated upwards to the root element. This propagation helps to model document-based retrieval as opposed to element-based one.

To generate the relationships we parsed the plot elements using ASSERT version 0.14b. The parser identifies verb predicate-argument structures and labels the arguments with semantic roles (see Figure 2). To represent the verb predicate-argument structures in the schema, the verb, labelled *target* is represented as the "RelshipName" (see Figure 3).

[17] provided the test-bed which included 50 queries (40 queries for testing and 10 for parameter tuning). These queries were created assuming a situation in which a user wants to find a movie using partial information spanning over many elements. Relevant documents were found manually. The dataset was not stemmed except for the predicates generated by ASSERT. This was done to improve recall. Stopwords were not removed.

The models required some parameters to be tuned in advance. We determined the weighting parameter for the combined models. We set aside 10 training queries to find the best-performing parameters and used these parameters for the test queries. To estimate the weighting parameters we performed an iterative search with a step size of 0.1 for the weighting parameter, resulting in 11 possible values. Moreover, to ensure that this was a valid probability distribution, we placed a constraint that the weights add up to one. Naturally, the aforementioned parameters are dependent on the characteristics of the underlying dataset. Therefore, the indicated values of w_X parameters vary, and, hence the mentioned values provide only a guide to how they were set.

The *baseline* is based on the document-oriented TF-IDF. In this model the structure of the data is not taken into consideration and a bag-of-words representation is utilised to estimate the scores.

6.2 Results & Discussion

Table 1 gives a numerical representation and compares the retrieval quality of the knowledge-oriented retrieval models with the term-only retrieval model (TF-IDF). The table also compares the macro- and micro-based models. The values in parenthesis represent the relative (percentage) difference in performance from the baseline results. We use MAP (Mean Average Precision) as the performance metric. The w_X parameters, which must add up to one are used to control the contribution of each RSV to the overall ranking. To run the experiments all of the mappings were considered. The tuning processed showed that the best performing parameters for the macro-based models are $w_T = 0.4, w_C = 0.1, w_A = 0.4, w_R = 0.1$, while for the micro-based models the best performing parameters are $w_T = 0.5, w_C = 0.2, w_A = 0.3, w_R = 0$. Table 1 also reports the extreme combinations, i.e. w_T and one of $w_{\{C,A,R\}}$ are set to 0.5, while the other two parameters are set to 0.

We see from Table 1 that, in most cases, the knowledge-oriented retrieval models have much higher precision than the term-only (bag-of-words) approach. Using the best performing parameters from the tuning, the baseline is outperformed by both the macro- and micro-based models. The difference in performances

is more evident in the micro-based combination, with a 14.63% improvement over the TF-IDF retrieval model. The macro-based TF-IDF+AF-IDF retrieval model (i.e. $w_T = 0.5$ and $w_A = 0.5$) provides a 23.67% improvement over the bag-of-words approach, showing the best overall performance. Similarly, the micro-based TF-IDF+AF-IDF retrieval model outperforms the baseline with a 14.93% improvement.

The TF-IDF+CF-IDF retrieval model ($w_T = 0.5$ and $w_C = 0.5$) does not outperform the baseline in the case of macro-based combination nor micro-based combination. The difference is particularly stressed in the macro-based combination, where the TF-IDF+CF-IDF retrieval model provides a -18.66% difference on the baseline.

The relationship-based retrieval model ($w_T = 0.5$ and $w_R = 0.5$) has little impact on the overall RSV. This is because there are very few documents with relationships in the dataset (from 430,000 documents there are only 68,000). Many of the documents do not contain the plot element or the plot is too short for the parser to generate meaningful relationships. These two factors degrade the impact of the model on the overall RSV. With a larger dataset, we may see the benefit of the relationship-based retrieval model.

Overall, the evaluation demonstrated the effectiveness of the proposed framework to instantiate knowledge-oriented retrieval models and semantically expressive queries. Future work will incorporate other baselines that already consider the underlying structure and semantics in the data. Additionally, future work will show how other data formats and sources of knowledge can be incorporated in the retrieval process, especially sources of knowledge that are rich with relationships.

7. CONCLUSION

This paper contributed an approach through which retrieval models and queries traditionally designed for keyword-based retrieval become instead based on terms, classifications, relationships and attributes. The transformation is a direct result of leveraging a schema that represents content and factual knowledge in one congruent framework.

The schema-driven approach helps to quickly incorporate different data formats into a standard representation and instantiate retrieval models that support constraint-checking and ranking with respect to these data formats. Furthermore, keyword-based queries are automatically mapped to their corresponding semantic predicates which reflect the underlying schema (e.g. the keyword "brad" is mapped to the class "actor"). Together, the knowledge-oriented retrieval models and the semantically-expressive queries are shown to be an effective approach for semantic-aware retrieval over heterogeneous data.

In conclusion, having a schema is a good first step because adopting a common model and base schema can make it possible for search services to utilise additional data formats that are appropriate for them. It is crucial that the discussion evolves from details of encoding/notation to vocabulary design and schemas for effective, semantically-aware keyword-based search.

Future work will investigate how other keyword-based retrieval models can be instantiated from the schema. Furthermore, we will investigate the extent to which the performance of the models depends on the quality of the generated query mappings and the used data formats. The examples and experiments presented in this paper primarily focused on semantic data explicated using XML and predicate-argument annotations. Future work will show how other data formats and sources of knowledge can be incorporated in the retrieval process.

8. ACKNOWLEDGMENTS

We are grateful for the support of Yahoo! Labs Barcelona. We would also like the reviewers for their excellent suggestions.

9. REFERENCES

[1] D. J. Abadi, A. Marcus, S. R. Madden, and K. Hollenbach. Scalable semantic web data management using vertical partitioning. In *VLDB*, pages 411–422, 2007.

[2] S. Agrawal, S. Chaudhuri, and G. Das. DBXplorer: A system for keyword-based search over relational databases. In *ICDE*, pages 5–16, 2002.

[3] H. Azzam and T. Roelleke. A generic data model for schema-driven design in information retrieval applications. In *ICTIR*, 2011.

[4] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using banks. In *ICDE*, pages 431–440, 2002.

[5] M. W. Bilotti, P. Ogilvie, J. Callan, and E. Nyberg. Structured retrieval for question answering. In *SIGIR*, pages 351–358, 2007.

[6] C. Bizer and A. Schultz. Benchmarking the performance of storage systems that expose SPARQL endpoints. In *ISWC*, 2008.

[7] J. Callan. Passage-level evidence in document retrieval. In *SIGIR*, pages 302–310, 1994.

[8] J. Callan. Search engine support for software applications. In *CIKM*, pages 1–2, 2010.

[9] R. Cornacchia and A. P. de Vries. A parameterised search system. In *ECIR*, 2007.

[10] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, pages 864–875, 2004.

[11] S. Elbassuoni and R. Blanco. Keyword search over RDF graphs. In *CIKM*, pages 237–242. ACM, 2011.

[12] N. Fuhr, N. Goevert, and T. Roelleke. Dolores: A system for logic-based retrieval of multimedia objects. In *SIGIR*, pages 257–265, 1998.

[13] S. Harris and N. Gibbins. 3store: Efficient bulk RDF storage. In *PSSS*, volume 89, 2003.

[14] D. Hiemstra and V. Mihajlovic. A database approach to information retrieval: The remarkable relationship between language models and region models. Technical Report arXiv:1005.4752, May 2010. Comments: Published as CTIT Technical Report 05-35.

[15] V. Hristidis and Y. Papakonstantinou. DISCOVER: Keyword search in relational databases. In *VLDB*, pages 670–681, 2002.

[16] G. Kasneci, F. M. Suchanek, G. Ifrim, M. Ramanath, and G. Weikum. Naga: Searching and ranking knowledge. In *ICDE*, pages 953–962, 2008.

[17] J. Kim, X. Xue, and W. B. Croft. A probabilistic retrieval model for semistructured data. In *ECIR*, pages 228–239, 2009.

[18] W. Lu, S. E. Robertson, and A. MacFarlane. Field-weighted XML retrieval based on BM25. In *INEX*, pages 161–171, 2005.

[19] C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In *SIGIR*, pages 298–308, 1993.

[20] P. Mika, E. Meij, and H. Zaragoza. Investigating the semantic gap through query log analysis. In *The Semantic Web - ISWC 2009*, volume 5823 of *Lecture Notes in Computer Science*, pages 441–455. 2009.

[21] T. Neumann and G. Weikum. RDF-3X: A RISC-style engine for RDF. *PVLDB*, 1(1):647–659, 2008.

[22] P. Ogilvie and J. Callan. Language models and structured document retrieval. In *INEX*, pages 33–40, 2002.

[23] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281, 1998.

[24] S. Pradhan, W. Ward, K. Hacioglu, J. H. Martin, and D. Jurafsky. Shallow semantic parsing using support vector machines. In *HLT-NAACL*, pages 233–240, 2004.

[25] C. Ré and D. Suciu. Materialized views in probabilistic databases: for information exchange and query optimization. In *VLDB*, pages 51–62, 2007.

[26] S. Robertson. Understanding inverse document frequency: on theoretical arguments. *Journal of Documentation*, 60(5):503–520, 2004.

[27] S. Robertson, H. Zaragoza, and M. J. Taylor. Simple BM25 extension to multiple weighted fields. In *CIKM*, pages 42–49, 2004.

[28] T. Roelleke. A frequency-based and a Poisson-based probability of being informative. In *SIGIR*, pages 227–234, 2003.

[29] T. Roelleke and N. Fuhr. Retrieval of complex objects using a four-valued logic. In *SIGIR*, pages 206–214, 1996.

[30] G. Salton, A. Wong, and C. Yu. Automatic indexing using term discrimination and term precision. *Information Processing and Management*, 12:43–56, 1976.

[31] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.

[32] M. Schmidt, T. Hornung, G. Lausen, and C. Pinkel. SP2Bench: A SPARQL performance benchmark. *CoRR*, 2008.

[33] B. Sigurbjornsson, J. Kamps, and M. de Rijke. An element-based approach to XML retrieval. In *INEX*, 2003.

[34] M. Stonebraker, D. J. Abadi, A. Batkin, X. Chen, M. Cherniack, M. Ferreira, E. Lau, A. Lin, S. Madden, E. J. O'Neil, P. E. O'Neil, A. Rasin, N. Tran, and S. B. Zdonik. C-Store: A column-oriented DBMS. In *VLDB*, pages 553–564, 2005.

[35] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *WWW*, New York, NY, USA, 2007.

[36] R. van Zwol and T. van Loosbroek. Effective use of semantic structure in XML retrieval. In *ECIR*, pages 621–628, 2007.

[37] K. Wilkinson, C. Sayers, H. A. Kuno, and D. Reynolds. Efficient RDF storage and retrieval in jena2. In *SWDB*, pages 131–150, 2003.

[38] H. Zaragoza, H. Rode, P. Mika, J. Atserias, M. Ciaramita, and G. Attardi. Ranking very many typed entities on wikipedia. In *CIKM*, pages 1015–1018, 2007.

[39] L. Zhao and J. Callan. A generative retrieval model for structured documents. In *CIKM*, pages 1163–1172, 2008.