Combining Link and Content-based Information in a Bayesian Inference Model for Entity Search

Christos L. Koumenides
Electronics and Computer Science
University of Southampton
SO17 1BJ, Southampton, UK
clk1v07@ecs.soton.ac.uk

Nigel R. Shadbolt
Electronics and Computer Science
University of Southampton
SO17 1BJ, Southampton, UK
nrs@ecs.soton.ac.uk

ABSTRACT

An architectural model of a Bayesian inference network to support entity search in semantic knowledge bases is presented. The model supports the explicit combination of primitive data type and object-level semantics under a single computational framework. A flexible query model is supported capable to reason with the availability of simple semantics in queries.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models*

General Terms

Design, Theory

Keywords

Bayesian belief networks, entity search, information retrieval, semantic search

1. INTRODUCTION

Semantic search [1] is considered by many as the natural evolution of current search technology. While many conventional retrieval models have been proven to work effectively over coarse document collections, there are many inherent obstacles to overcome when focus starts to shift toward items of finer granularity. Product search is a typically cited example of this realm. Traditional approaches to Information Retrieval (IR) often treat documents as collections or bags of individual words, and their correspondence to a similar representation of user queries generally determines their level of similarity. This notion has often been coupled with features based on links, such as popularity and usage when search is conducted over Web-accessible documents. The idea of semantic search is to diverge from this coarse view and sometimes monotonic treatment of documents to a finer perspective, one that will be able to exploit and reason intelligently with granular data items, such as people, products, organisations or locations.

Entity search has been a key task in this context and an occasional direction in some of the recent research tracks in IR. The Semantic Web (SW) community has recently organised a Semantic Search Challenge[1] forum, aiming to prioritise and evaluate research into *ad-hoc object retrieval* [7]. Similarly to traditional document retrieval, the task focuses on keyword or free-form text search, except SW knowledge bases come to replace document collections. An entity on the SW can denote many things and is generally treated as anything that is addressable by a URI and can serve as the subject of a description (where a description is more formally depicted by a collection of triples, which may serve as the concise representation of a resource). Several promising developments have emerged, both throughout the two years that the Challenge has been active and other individual works that try to integrate the two disciplines (SW and IR).

In this paper, we present the ground architectural components of a new retrieval model for entity search. The model is based on the Bayesian Network (BN) approach and, as customary with similar approaches in IR, tries to generalise into a single computational framework the necessary constructs to reason with several sources of available knowledge. The model differs from similar deployments of BNs in IR in that it aims to represent, and make explicit in the inference process, the presence of multiple relations that potentially link semantic resources together or with primitive data values, as it is customary with SW data. Part of our goal in designing this model has been to enable reasoning with more complex or expressive information needs, with semantics specified explicitly by users or incorporated via more implicit bindings. The model is not necessarily restricted to SW data and may be applicable to any form of data that pertains to the triple-based representation of knowledge bases. The ground foundations of the model offer a rich setting to incorporate a variety of techniques for fusing probabilistic evidence, both new and familiar.

In what follows, we give a detailed description of our translation choices and topological properties of the model. As this is still work-in-progress, we leave out many aspects of the model, such as conditional probability assignments and inference progressions, as these would be more appropriate to a fuller publication along with results from our evaluation experiments.

[1] http://semsearch.yahoo.com

JIWES, SIGIR'12, Aug 16, 2012, Portland, Oregon, USA

2. BAYESIAN INFERENCE NETWORKS

Bayesian belief networks [11] are among the best understood stochastic methods for modelling joint probability distributions within a domain of interest. Formally, they are directed acyclic graphs (DAGs) in which nodes represent propositions, or random variables, and arcs portray dependence relations between propositions. Vertices are assigned to every variable in the domain and arrows are drawn toward each vertex X_i from the set of vertices Π_{X_i} perceived to have a direct influence (typically a *causal* influence) on X_i. The strength of these influences are expressed by conditional probabilities of the form $p(x_i|\Pi_{X_i})$ assigned to every variable in link matrix form, otherwise known as conditional probability tables (CPTs) in the case of discrete types of networks. These are judgemental estimates encoding our belief that a *child* proposition takes on a value $(X_i = x_i)$ given any value combination of its set of *parents* Π_{X_i}. In principle, the size of a complete matrix specification is exponential to the number of direct parents in the network. In practise, however, parent relationships are usually structured in prototypical clusters of variables requiring fewer quantifiable estimates, such as Noisy-OR gates [11]. The roots of a network are the nodes without parents and also require a CPT, except that it is degenerated into a single row of size n, representing the prior, or marginal probability of the node e.g. $p(x_i)$ for each of its n possible instantiation states.

Conditional probability estimates are consistent if assessed by any set of functions $F_i(x_i, \Pi_{X_i})$ that satisfy

$$\begin{aligned} \sum_{x_i} F_i(x_i, \Pi_{X_i}) &= 1, \\ 0 \;\leq\; F_i(x_i, \Pi_{X_i}) \;&\leq\; 1 \end{aligned} \tag{1}$$

where the summation ranges over the states of X_i. The product form $\prod_i F_i(x_i, \Pi_{X_i})$ constitutes a joint probability distribution that supports the dependencies in the network.

Once factual knowledge about a domain has been compiled into a complete dependency graph, the resulting network becomes a computational architecture for reasoning about that knowledge. The links in the network are treated as message-passing facilities used to propel evidence about the instantiation of variables through the network, allowing to compute the probability or degree of belief associated with the remaining nodes. Belief propagation is viewed as a recursive interaction process between adjacent nodes, which works by looking up values stored in the CPTs of intermediate variables. Restrictions on the topology of a network can lead to different schemes for fusing and combining these probabilities. In general, there are two components that operate independently in a typical belief-updating process: a top-down form of inference in which parent nodes mediate *predictive* or *prior* support to their children, and bottom-up evidential reasoning in which children provide *diagnostic* support to their parents.

For singly connected networks, it is possible to devise *exact* propagation algorithms to infer the posteriors of all the nodes in a network (reach a state of equilibrium) in time proportional to the network's diameter [11]. The complexity of multiply connected networks (networks with cycles) is often treated with approximated or assumption-based reasoning, since propagation with exact algorithms will inevitably fall short (double counting of evidence, loopy propagation), a case generally considered to be NP-Hard [3].

2.1 Relevance to IR

Probabilistic models in IR have been integral for reasoning with uncertainty in a wide range of tasks. Some of the earliest and pioneering techniques in the field were designed around models that base their core assumptions on rudimental probabilistic and Bayesian principles, such as the binary independence and language modelling approaches [9]. BN representations emerged in the late 1980s as extensions of classical probabilistic models and since then have been applied in a variety of ways within the field, both in practical implementations and as conceptual frameworks.

Among the earliest introductions of the formalism to IR have been the works on the *Inference Network Model* [13] and *Belief Model* [12]. These were initially designed as retrieval frameworks aimed to generalise existing approaches (e.g. vectorial ranking) and integrate several sources of knowledge in a single framework (e.g. relevance feedback or multiple document and query representations). Later works extended the ideas to incorporate additional features into the ranking process, such as document structure [10, 4] and hypertext link analysis [5, 2]. Successful implementations are also found in document clustering and classification [6], conversational agents [8], and other related fields. Precise propagation and reasoning in Bayesian IR networks remained intractable tasks, and their design was largely focused on the interpretation of complex dependencies as canonical functions that are practical and easier to implement.

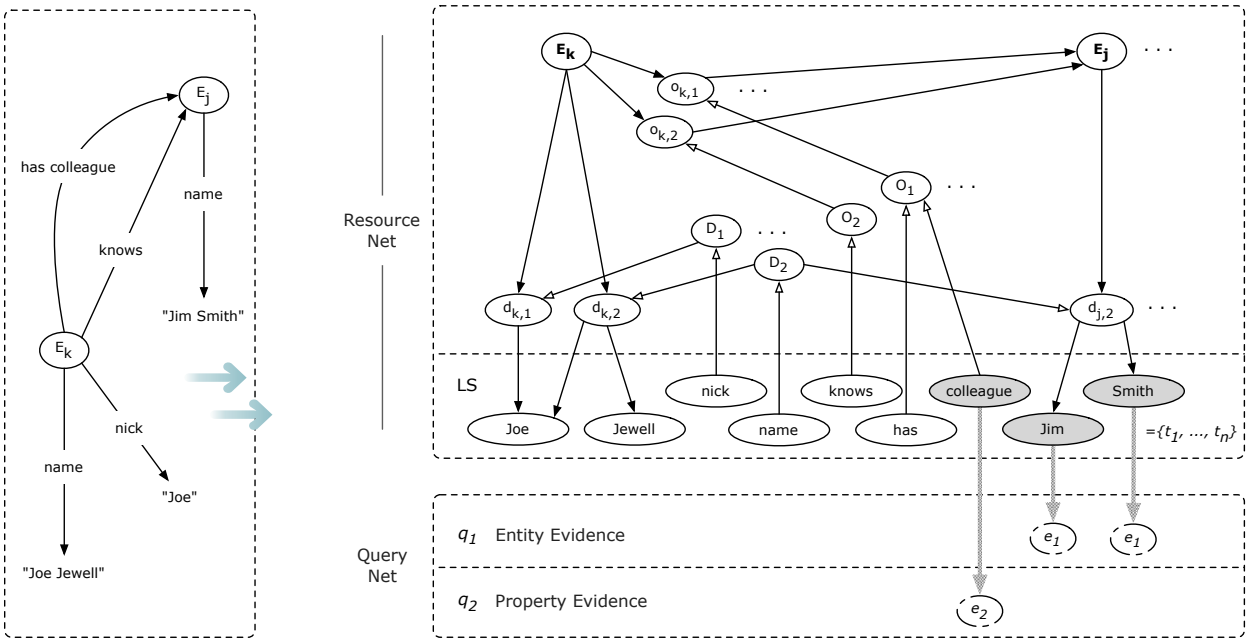
3. A BN MODEL FOR ENTITY SEARCH

The underlying building block of semantic knowledge bases is a subject-predicate-object triple, whereby subjects and objects are allowed to be interchanged. A knowledge base may be thought of otherwise as a loosely coupled directed labeled graph (DLG), where subjects and objects are treated as nodes and predicates as labeled edges (relations) between them. DLGs are a common and generic model to describe possibly any type of semantic network or association graph. On the SW, relationships are first-class URI resources and can be defined locally or reused from existing vocabularies. The goal of our translation is to devise a generative model for projecting the DLG manifestation of knowledge bases to a form of DAG, on which we can delicate retrieval of resources to an evidential reasoning process. The outcome is not initially acyclic, cycles exist in the model, but this is a common scenario with BNs and will demand special treatment and reasoning during the inference process.

The resulting model is not necessarily restricted to SW data, since a translation from a DLG model can have a broader perspective. Dependence implications from SW assertional and terminological constructs will be treated by the same general-purpose statistical schemes. As it is customary with BNs in IR, we treat the model as an expressive architectural framework on which we can approximate reasoning using various generic functions of standard IR schemata (e.g. functions to estimate term frequency, field weighting, and link proximity).

3.1 Overview

A perspective view of the model is presented in Figure 1. From the outset, the model consists of two component networks: a static *resource network* containing information about data resources and their semantic interrelations, and



(a) DLG **(b)** Translated BN

Figure 1: Perspective view of the model, considering a query for "*colleagues of Jim Smith*".

a dynamic *query network* containing a (tacit) specification of the user's information need.

The *resource network* is a dense network intended to capture and quantify semantics as probabilistic dependencies among binary random variables. The network is built once for a given collection and remains unchanged during query processing. Nodes in the resource network are binary-valued propositions and will take on values from the set {*true, false*}. Our focus is on the evaluation of entities, portrayed as a series of $E_{\_}$ variables in the model. All of our assumptions will be defined accordingly to reason with entities.

The *query network* is a dynamic component represented by two virtual layers (q_1 and q_2). Query layers enclose the initial evidence to be factored into the resource network. These are used to fix the instantiation of resources and render the flow of propagation. We explore two types of evidence in our experiments: the presence of relations in the query and the presence of literals associated with entities. The latter will be the initial evidence to engage in propagation and the former will be used to affect the impulse of evidence through the associated dependency links. The query network is a temporal network created whenever a user queries the collection and only exists during query processing. Once a result is obtained, the query will be discarded, unless further processing or expansion is expected. Query nodes are always assigned the value of *true*, indicating that an information need has been observed and the corresponding query formulated.

Mappings between the two networks determine the inference paths to be traversed during evaluation of entities. The Literal Space (LS in Figure 1) acts as the main mapping facility between the query and the rest of the network. The LS contains an assortment of text representation nodes extracted from the primitive data-type values in the knowledge base. Mappings are dynamic and can entail topological restrictions on the inference and instantiation entailments of resource variables. On the whole, retrieval will be geared in terms of the concurrence of two estimates: **entity-diagnosis** and **entity-prediction**. How these are extracted and coordinated will be determined by the instantiation conditions of the variables and the criteria in our ranking strategy.

3.2 The Resource Network

There are three types of nodes in the resource network: nodes depicting candidate entities for retrieval (we will refer to them as *entity members* or *member variables*), nodes depicting relations between entities and with primitive datatype values (otherwise, object and datatype property nodes), and text representation nodes that depict the actual datatype values in a knowledge base. Property nodes are demarcated between local and global, as will be explained shortly. A local context for each entity is defined in the model, reflecting the local use of semantics in the model (datatype and object relations). The following terminology will remain fixed, although with arbitrary content:

- $\mathcal{U}$ is the set of all resources in a knowledge base that participate in a subject-predicate-object triple
- $\mathcal{S} \subseteq \mathcal{U}$ is the set of all subjects
- $\mathcal{O} \subseteq \mathcal{U}$ is the set of all objects
- $\mathcal{L} \subseteq \mathcal{O}$ is the set of all primitive data type objects
- $\mathcal{R} \subseteq \mathcal{U}$ is the set of all properties/relations

Subjects and objects are allowed to be interchanged, hence the condition $\mathcal{S} \cap \mathcal{O} \neq \emptyset$ can hold, given the completeness of the working set. Relations are partitioned into object properties $R_o \subseteq \mathcal{R}$ (linking resources together) and datatype properties $R_d \subseteq \mathcal{R}$ (linking resources in $\mathcal{S}$ to literals in $\mathcal{L}$). The subsumption $\mathcal{R} \subseteq \mathcal{S} \cup \mathcal{O}$ is also true, since a property can itself be the subject or object of a different relation.

3.2.1 Entity members

A subset $E \in \mathcal{S}$ from the knowledge base is selected as candidate for retrieval and translated to n binary random variables, $\{E_i, ..., E_n\}$ in the Belief Net. We keep the definition of E arbitrary for now and include any one or more first-class resources that participate in a triple (according to our earlier definition, this may include either relations and/or subjects). A member variable set to true ($E_i = true$) is said to be activated by the query for evaluation. Activation of member variables will be subject to whether a diagnostic path is open between the member variable and evidence in the query. A path is initiated via a mapping to the LS through which diagnosis can reach the member via any number of *datatype* properties (covered next). Figure 2 shows two paths through which diagnosis can reach member variables.

Entity members are evaluated in isolation, so each will consume a separate propagation process. A member variable will be instantiated to a truth state ($E_i = true$) when any of its diagnostic paths contains a binding to query evidence. A binding to query evidence will mostly involve instantiation of nodes in the LS, although other restrictions are also applicable. Consequently, retrieval considers entity members that have been activated as *true* and will dedicate a separate trial for each. Active members will either be treated for evaluation or used to support the evaluation of others.

3.2.2 Property nodes

The set of properties $\mathcal{R}$ in a knowledge base is composed of two different sets, $\mathcal{R} = R_o \cup R_d$: The set $R_o = \{O_i, ..., O_n\}$, containing binary random variables representing the n translated object properties, and the set $R_d = \{D_i, ..., D_n\}$, representing the n translated datatype properties [2]. Property nodes in the BN are separated between *local property nodes* (local to each entity member) and *global property nodes* (global across the entire knowledge base). The aforementioned definitions correspond to global property variables. The reason for defining two types of properties is pragmatic and will be explained shortly.

Global Property Nodes. Global property nodes are modelled as conditionally dependent on term nodes in the LS representing the actual labels associated with properties in the knowledge base. Property labels are used to establish mappings with the query network, allowing property nodes to be predicted as potential query elements. In our current implementation, labels are extracted from `rdfs:label` relations associated with property definitions (w.r.t. SW data) or deduced from the property URIs via simple heuristics (e.g. where `http://xmlns.com/foaf/0.1/name` resolves to *name*) when no such label exists. Other label forms may be preferred over `rdfs:label` during the translation phase. Details of implementation are irrelevant at this point and will be the subject of a separate report.

A global property node set to *true* ($\{D_i, O_i\} = true$) is said to be instantiated by evidence in the query. This type of evidence is predictive, thus the prediction of properties entails their instantiation state. Observed properties will be used as **logical conditions** to delimit the instantiation of local properties as a result of the mapping to the query.

[2]It should be clear whether O_i (D_j respectively) refer to the actual properties or the translated nodes in the BN

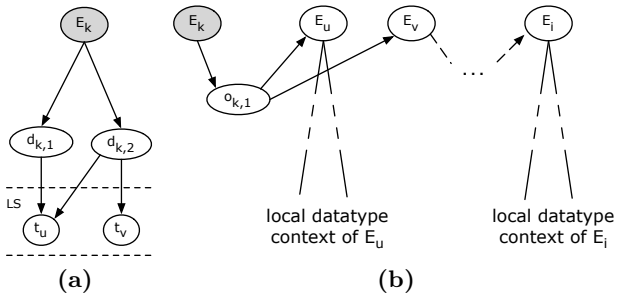


Figure 2: Diagnosis reaching a member variable via (a) the member's local datatype context, and (b) the local datatype contexts of other entities.

Instantiation of global properties will not be fused in the inference process but will be used to render/influence the conditional dependencies associated with local property nodes and the candidate entities. This opens the possibility of enabling query semantics to influence the diagnosis arriving at entities from the LS.

Local Property Nodes. Local property nodes ($d_{i,j}$ or $o_{i,j}$) are defined to associate higher order properties (global properties) to a local context defined for each individual entity member. Local properties always descend from a single global property node and a single entity member, which act as the parents of the respective node. The naming convention used to distinguish properties in the diagram is adopted to reflect its parents e.g. a node with the set of parents E_i and D_j is named $d_{i,j}$ accordingly. Local property nodes are conditionally independent with each other, given their set of parents and children in the LS. There are no direct connections between them and can have multiple descendants in the network e.g. a node can link to several entity members in the network (case of object relations) and to several term nodes in the LS (case of datatype relations).

A binary value ($true/false$) associated with a local property will reflect the instantiation of the corresponding global property node i.e. a variable is set to *true* exactly when its parent property is *true* ($o_{i,j} = true : O_j = true$). Consequently, properties will be marked as either *true* or $false$ exclusively in each query evaluation. The conditional dependencies of local properties (e.g. $p(d_{i,j}|E_i)$ or $p(o_{i,j}|E_i)$) based on their states are the main methods for external parameterisation of the model, allowing to interpret the evidence of relations in the query. This is a desirable property for resolving more complex queries, and quantities will vary according to the type of query formalism explored (whether evidence should be treated as more explicit or implicit provisions).

Local property nodes have a significant role in the network. First, they facilitate the translation of the bidirectional use of properties on the SW, something not possible with a global form of representation alone. Recall that subjects and objects in a triple-based knowledge base are interchangeable, meaning that the same relation can be used to link to and from the same entity. In BNs, a node can only exist on one side of the relation i.e. nodes cannot be both the cause and effect in a given relation. Second, local properties delineate a clearly defined sample space on which paths from the LS

can be quantified individually for each entity. For example, $p(t_k|d_{i,j})$ allows quantifying the relation of term node t_k to a specific datatype property in the context of E_i. This brings together a nice formalism for traditional weighted-field retrieval, essentially treating an entity member and its set of local properties as a structured document, but with the added expressivity due to the different instantiation states of properties.

3.2.3 The Literal Space

The set $U \subseteq \mathcal{L} : U = \{t_1, ..., t_n\}$ represents the set of all index terms extracted from a knowledge base (including property labels), modelled as n random variables. Every node in the LS corresponds to an index term extracted via some form of term extraction technique. For example, if the string "semantic search" has been extracted into the distinct terms "semantic" and "search", then two representation nodes are created. Term nodes are considered conditionally independent with each other given their set of parents (local datatype properties) and children (global property nodes). There can be several paths between the nodes in the LS and the local contexts of entities (e.g. a literal associated with `foaf:name`, `dc:title`, and `rdfs:label` bounded to the same entity), and terms can be shared across member contexts. The dependency of global properties on term nodes asserts that prediction of properties will be initiated from inside the LS, although the connection will be treated like a decision link, since predictive evidence will not be fused further in the network.

The LS exposes a natural interface between the query and the rest of the network. Query evidence need only attach to the LS, while different propagation signals using different combinations of query nodes can result in a variety of expressive query formalisms. Evidence will initially flow from the query to the LS and propagate through the rest of the network by unfolding the space covered by term nodes, for every entity being evaluated. A term node instantiated to true ($t_i = true$) is said to be observed by evidence via a direct mapping to the query.

3.3 The Query Network

The query network reflects the overall strategy for meeting a user's information need. In general, we treat information requests as tacit specifications of a data resource, provided as either a combination of keywords or a form of semi-structured natural language description, which remain mostly ambiguous and internal to the requestor. A ranking strategy is intended to transform these implicit specifications into an execution plan for evaluating and retrieving instances from a knowledge base.

Query evidence is enclosed within two distinct layers: Entity Evidence and Property Evidence. Query layers depict different aspects of a request, such as the presence of a literal or a property definition, and may be evaluated in various combinations for potentially more optimal results. We expect that queries of the form "*friends of Jim Smith*" or "*drama movies directed by Jim Smith*" will be treated with special emphasis on their semantics. It will be possible to evaluate several such patterns in a single query e.g. "***friends of** Jim Smith who **live in** California*". The semantics are implicit and should not block any other paths in the model. Ideally, we would want to maximise precision in a given context without affecting recall in the final results.

Query layers attach to the LS by a set of links whose only purpose is to instantiate term nodes to some initial state. Hence information flows one way only - from the query layers to the variables affected by the observations. The query, in effect, instantiates a part of the network composed of the nodes and links participating in the computation. Our focus is on query layers that are induced via fully automatic means. These will remain ambiguous specifications of the aspects they intend to cover and their impact will be implicit on the inference, just enough to intensify the probability of observing the corresponding resources. Manual query construction can aid to transform evidence into more explicit provisions for the inference, thus facilitate better understanding of the user's intent. The contents of each layer are explained next.

3.3.1 Entity Evidence

The first layer, q_1, encloses a set of independent dummy variables representing the (processed) terms in the user's query that match to indexes descending from local datatype properties. This excludes terms associated with global property nodes. Every node in this layer is considered a disparate frame of knowledge that will be used to propagate diagnosis to the *local datatype contexts* of entity members. Nodes that do not match to index terms will contain no mapping to the LS.

3.3.2 Property Evidence

The second layer, q_2, encloses a set of potential property definitions present in the query. Nodes in this layer attach to terms in the LS linked to global property nodes. The idea is that a strong evidence in the query may instantiate a global property node to $true$. Since global properties influence directly the instantiation of local property nodes, this will intensify or weaken the evidence that flows through the local context of entities (initiated from q_1) via the respective local property node e.g. via $p(o_{i,j}|E_i)$. This, in turn, solidifies in the inference the presence of a relation in the query.

In the case that global properties are associated with several terms in the LS, then we must decide whether there is substantial evidence in the query to affect their instantiation. This will need to be captured by heuristics that can approximate the degree of coverage of the property definition (associated indexes in the LS) by the respective query layer (q_2). Details of the current implementation will be presented in a forthcoming publication.

4. COMPLETING THE MODEL

4.1 Estimating conditional probabilities

In order to complete the translation and firm up the model for inference, the remaining issue is to quantify the conditional and marginal probabilities for all the nodes in the network. The resulting distributions will be unified and organised into inference progressions that will form our ranking strategy. Conditional probabilities are the mechanisms by which we reason in the model, in essence giving us a quantitative perspective over the dependencies. Estimates are required for five different node types: term and query nodes, local and global property nodes, and entity members.

The arbitrary complexity and size of the model suggest that we must seek alternative strategies, beyond exact heuristics depended on precise CPT specifications, if we are to achieve

computationally tractable inference in the network. Associating with every variable a CPT that enlists probability estimates for all possible value combinations of its parents is rather impractical, if at all feasible, since the construction of exact CPTs requires prior knowledge of the type and number of parents being conditioned. Many of our probability assignment choices are also tightly coupled with the assumptions in the ranking strategy. For example, many complex interactions/dependencies are enclosed within prototypical functions that resemble traditional scalar and other functions used in IR (hence can affect how the estimates are computed).

Considering term nodes for an example, our strategy has been to decompose the specification $p(t_v|d_{i,k}, ..., d_{j,n})$ into a series of prototypical functions $f : (t_v, d_{i,j}) \rightarrow W_{d,t}$ over all property nodes ascending from t_v. This is justified by a set of independence assumptions between terms and properties that we are willing to accept. In effect, if we consider Πt_v to be the parents of term t_v the weighting function can provide a value $p(t_v|d)$ for every $d \in \Pi t_v$. $W_{d,t}$ can feature the effect of an indexing weight, such as the relative frequency of a term inside the value associated with the respective property. Depending on how we combine and normalise evidence from multiple terms prior to reaching the entity variables, term weights need not be probabilistic estimates either, and can feature any variation of a TF-IDF weighting scheme. We will detail probability assignments and inference progressions in a forthcoming publication.

4.2 Ranking strategy

Our intuition for ranking is that every entity member is treated separately for evaluation. The knowledge base, therefore, gets partitioned between two, possibly uneven, disjoint parts in every evaluation: events that relate to the given entity and events that do not. When a query is issued to the system, it is treated as an observable event that is intersected with the partitions of the universe. What we are interested to measure is the degree of coverage by the query of the space covered by a given entity. Considering an entity E_i and a query specification Q our goal is to estimate $p(E_i|Q)$. Using Bayes rule, the expression $p(E_i|Q) \propto p(Q|E_i)p(E_i)$ forms the basis of the network shown in Figure 1. The process proceeds by unfolding the equation and inferring its parts via probabilistic inference: bottom-up belief propagation for the likelihood $p(Q|E_i)$ and top-down propagation for the priors $p(E_i)$.

5. CONCLUSIONS AND FUTURE WORK

This paper seeks to contribute to a better understanding of the use of Bayesian inference networks to support entity search in semantic knowledge bases. We have presented the architectural components of an expressive retrieval model capable to exploit and reason with semantics in queries and data resources. The ground foundations of the model offer a rich setting to satisfy an interesting set of queries. Our main line of future research involves evaluating the model to determine the proper contexts for potential deployments and use.

6. ACKNOWLEDGMENTS

This research has been supported by the EnAKTing project, funded by the Engineering and Physical Sciences Research Council under the contract EP/G008493/1. Special thanks to our colleagues Manuel Salvadores and Tope Omitola for their valuable comments.

7. REFERENCES

[1] R. Baeza-Yates, M. Ciaramita, P. Mika, and H. Zaragoza. Towards Semantic Search. In *Natural Language and Information Systems*, pages 4–11. Springer Berlin / Heidelberg, 2008.

[2] P. Calado, B. Ribeiro-Neto, N. Ziviani, E. Moura, and I. Silva. Local versus global link information in the web. *ACM Trans. Inf. Syst.*, 21(1):42–63, Jan. 2003.

[3] G. F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2-3):393 – 405, 1990.

[4] F. Crestani, L. M. de Campos, J. M. Fernández-Luna, and J. F. Huete. A multi-layered bayesian network model for structured document retrieval. In *ECSQARU'03*, pages 74–86, 2003.

[5] W. B. Croft and H. Turtle. A retrieval model incorporating hypertext links. In *Proceedings of the second annual ACM conference on Hypertext*, HYPERTEXT '89, pages 213–224, New York, NY, USA, 1989. ACM.

[6] L. Denoyer and P. Gallinari. Bayesian network model for semi-structured document classification. *Inf. Process. Manage.*, 40(5):807–827, Sept. 2004.

[7] H. Halpin, D. M. Herzig, P. Mika, R. Blanco, J. Pound, H. S. Thompson, and T. T. Duc. Evaluating ad-hoc object retrieval. In *Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010)*. 9th International Semantic Web Conference (ISWC2010), 2010.

[8] K.-M. Kim, J.-H. Hong, and S.-B. Cho. A semantic bayesian network approach to retrieving information with intelligent conversational agents. *Inf. Process. Manage.*, 43(1):225–236, Jan. 2007.

[9] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July 2008.

[10] S. H. Myaeng, D.-H. Jang, M.-S. Kim, and Z.-C. Zhoo. A flexible model for retrieval of sgml documents. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 138–145, New York, NY, USA, 1998. ACM.

[11] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[12] B. A. N. Ribeiro and R. Muntz. A belief network model for ir. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, pages 253–260, New York, NY, USA, 1996. ACM.

[13] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst.*, 9:187–222, July 1991.