Noname manuscript No.
(will be inserted by the editor)

Information Retrieval as Semantic Inference

A Graph Inference Model applied to Medical Search

Bevan Koopman · Guido Zuccon · Peter Bruza · Laurianne Sitbon · Michael Lawley

Received: date / Accepted: date

Abstract This paper presents a Graph Inference retrieval model that integrates structured knowledge resources, statistical information retrieval methods and inference in a unified framework. Key components of the model are a graph-based representation of the corpus and retrieval driven by an *inference* mechanism achieved as a traversal over the graph.

The model is proposed to tackle the semantic gap problem — the mismatch between the raw data and the way a human being interprets it. We break down the semantic gap problem into five core issues, each requiring a specific type of inference in order to be overcome.

Our model and evaluation is applied to the medical domain because search within this domain is particularly challenging and, as we show, often requires inference. In addition, this domain features both structured knowledge resources as well as unstructured text.

Our evaluation shows that inference can be effective, retrieving many new relevant documents that are not retrieved by state-of-the-art information retrieval models. We show that many retrieved documents were not pooled by

B. Koopman
Australian e-Health Research Centre, CSIRO
E-mail: bevan.koopman@csiro.au

G. Zuccon
Queensland University of Technology
E-mail: g.zuccon@qut.edu.au

P. Bruza
Queensland University of Technology
E-mail: p.bruza@qut.edu.au

L. Sitbon
Queensland University of Technology
E-mail: laurianne.sitbon@qut.edu.au

M. Lawley
Australian e-Health Research Centre, CSIRO
E-mail: michael.lawley@csiro.au

keyword-based search methods, prompting us to perform additional relevance assessment on these new documents. A third of the newly retrieved documents judged were found to be relevant.

Our analysis provides a thorough understanding of when and how to apply inference for retrieval, including a categorisation of queries according to the effect of inference. The inference mechanism promoted recall by retrieving new relevant documents not found by previous keyword-based approaches. In addition, it promoted precision by an effective reranking of documents. When inference is used, performance gains can generally be expected on hard queries. However, inference should not be applied universally: for easy, unambiguous queries and queries with few relevant documents, inference did adversely affect effectiveness. These conclusions reflect the fact that for retrieval as inference to be effective, a careful balancing act is involved.

Finally, although the Graph Inference model is developed and applied to medical search, it is a general retrieval model applicable to other areas such as web search, where an emerging research trend is to utilise structured knowledge resources for more effective semantic search.

Keywords Semantic Inference · Medical Information Retrieval

1 Introduction

The challenge addressed by this paper is how to bridge the semantic gap: the mismatch between the raw data and the way a human being interprets it. Although the semantic gap problem is found in all domains, it is particularly prevalent in medical search. For example, when searching clinical records for patients suffering from *kidney disease*, a human being would readily infer that a relevant patient would be one undergoing *dialysis*. There exists valuable domain knowledge explicitly represented, yet trapped, in structured knowledge resources such as ontologies, which could potentially be leveraged to support such inferences. Although some state-of-the-art medical IR systems attempt to exploit these resources (Zhou et al, 2007; Koopman et al, 2012b; Limsopatham et al, 2013a,c), they lack the inference mechanisms that promote effective retrieval.

This article presents a Graph INference model (GIN), which we claim is a novel retrieval model integrating structured knowledge resources, statistical information retrieval methods and inference in a unified framework. The integration is provided by a graph-based representation of a corpus, with a structured knowledge resource providing the underlying skeleton. Information Units, be they terms, concepts or entities, are nodes in this graph. Edges represent relationships between these Information Units and these can be taken directly from the structured knowledge resource or derived from corpus statistics. Retrieval is modelled as an inference process and is realised as a traversal over the graph from nodes representing documents to those representing queries.

Some may view our retrieval as inference approach with surprise given the dearth of inference driven retrieval models. However, the strength of the retrieval as inference line of research, which started in the late nineteen eighties (Van Rijsbergen, 1986; Nie, 1989) and continued on into the nineties (Crestani and van Rijsbergen, 1995), was its ability to express different retrieval models within a *single* theoretical framework. This characteristic holds for the GIN, which can be more precisely viewed as not a single model, but a framework for expressing different inference based retrieval models. This article will investigate one such model. In addition, the promise of inference is the ability to infer relevant terms that are not usually captured by IR mechanisms such as pseudo-relevance feedback. In this article, we demonstrate how inference-based retrieval can return significant numbers of relevant documents which standard IR baseline models are blind to. The converse also holds, namely, inference has the potential to return larger numbers of irrelevant documents. This occurred when the inference mechanism utilised structured domain knowledge that was tenuous or not applicable to the specific context of the query. Finding the required balance between these two is, in our opinion, the most significant challenge for the retrieval as inference approach. Whilst we did not surmount this challenge, the analysis contributed in this article does provide a detailed understanding of when inference does promote effective retrieval and when it does not. It is our hope that this understanding will help bring a resolution of this challenge in the future.

In the next section, we categorise the different problems requiring inference. These are not specific to the medical domain. Therefore, inference is a general requirement for bridging the semantic gap. At the same time, structured knowledge resources akin to those used by the GIN are readily available outside the medical domain (for example, DBpedia[1] or Freebase[2]). Thus, the GIN provides a general framework to utilise structured knowledge resources for more effective semantic search and the lessons learned in the medical domain could apply more generally.

2 Inference Requirements for Information Retrieval

We break the semantic gap problem into five core issues. For each issue, we provide an example from the medical domain and then outline the type of inference required to address it.

2.1 Vocabulary Mismatch

Vocabulary mismatch occurs when particular concepts are expressed in a number of different ways, yet have a similar underlying meaning; for example,

[1] http://wiki.dbpedia.org/

[2] https://www.freebase.com/

Hypertension vs. *high blood pressure*. In addition, there are formal vs. colloquial variants for terms, regional differences and abbreviations and acronyms. These problems are present in all domains but due the complexity and nature of language in the medical domain there are often multiple variants for expressing the same concept, thus exacerbating the problem (Ely et al, 2000; Edinger et al, 2012; Koopman and Zuccon, 2014c). The effect in a retrieval scenario is that a query may have no overlapping terms with a document, yet the document could still be semantically highly relevant. A keyword-based IR system that returns only documents containing the query terms would not return these semantically relevant documents.

Two types of inference are required to overcome the vocabulary mismatch problem (Lancaster, 1986). First, statistical or associational inference can be employed to determine terms that are highly correlated in usage, such as synonyms. Standard IR approaches such as query expansion take advantage of terms with highly correlated usage; these approaches are an instantiation of associational inference. Second, and in contrast, deductive inference may be used in cases where linguistic resources (such as ontologies or thesauri) describe multiple alternative terms for a concept. The requirement for both association and deductive inference motivates research into a unified model that integrates structured ontologies and statistical, data-driven IR methods.

2.2 Granularity Mismatch

Queries are formulated using general terms/entities, whereas relevant documents contain specific instances of the general entities, or child concepts. For example, a query may contain *antipsychotic* while relevant documents would contain instances of antipsychotics, such as the drug *Diazepam* or the brand name *Valium*. Granularity mismatch is more prevalent in medical IR, particularly in searching electronic patient records which contain detailed descriptions and analyses of a patient's conditions, diagnoses and treatments, whereas queries express high-level information needs (Ely et al, 2000; Edinger et al, 2012; Koopman and Zuccon, 2014c). This mismatch between high-level query and low-level document renders an information retrieval system using keyword matches ineffective in searching medical data.

Overcoming granularity mismatch involves understanding when concepts are specialisations or generalisation of other concepts: a requirement that ontologies specifically model as parent-child or ISA relationships. However, ontologies typically do not provide a strength of association between parent and child (for example, *left kidney* is considered as similar to its parent *kidneys* as *kidney* is to its parent *organ*); thus it is not clear when it is appropriate to generalise or when to specialise.

The ability to infer more general or more specific concepts is essential for semantic search. The inference process is typically deductive in nature: determining when one concept is a parent or child of another. However, this inference mechanism needs to include a measure of uncertainty or similarity that is lacking in hierarchical ontologies. Inference with uncertainty is the foundation of probabilistic information retrieval models that estimate a probability of relevance. Thus this paper proposes a model that integrates explicit inheritance relationships from ontologies but also includes a necessary statistical estimation of uncertainty from IR models to address the issue of granularity mismatch.

2.3 Conceptual Implication

Although a relevant document may contain no query terms, the document may contain signs or evidence that drives a conclusion of the query. Specifically, certain terms within the document may logically infer the query terms and, by extension, relevance of the document to the query. For example, consider the query *Kidney disease* and a document that contains the terms *Dialysis machine*. For this query, a person reading the document would deduce *Dialysis machine* → *Kidney disease*. Conceptual implication is different from vocabulary mismatch, where two concepts are expressed differently but have the same meaning and different from granularity mismatch, where one concept is general and the other is specialised. Instead, with conceptual implications the document contains evidence in the form of a concept that logically infers the conclusion of another concept.

Conceptual implication situations are particularly prevalent when deducing diseases (Ely et al, 2000; Edinger et al, 2012; Koopman and Zuccon, 2014c) where:

- *treatment* → *disease*: the presence of certain treatments implies that the person has a certain disease; for example certain types of chemotherapy drugs imply the presence of certain cancers.
- *organism* → *disease*: the presence of certain organisms in laboratory tests imply the disease; for example *Varicella zoster* virus → *Chicken pox*.

The required mechanism for conceptual implication is deductive inference and logical deduction is the cornerstone mechanism for reasoning in ontologies (Sowa et al, 2000).

2.4 Inferences of Similarity

While some concepts can be derived by conceptual implication, others are more associational in nature. In this case, the presence of a certain concept indicates high likelihood of another, or the two concepts are semantically similar in some way. Disease comorbidities are an example of this case; comorbidities are the presence of one disease or more in addition to a primary disease, or the effect of such additional diseases. For example, *anxiety* and *depression* are two commonly co-occurring disorders.

An IR system needs to account for the innate dependence between medical concepts to be effective. The form of inference required in this case is associational. The types of relationships and associations required are typically not modelled in ontologies designed for deductive reasoning. These relationships are more suitably derived by statistical inference mechanisms typical of data-driven IR models.

2.5 Context-specific Semantic Gap Issues

There are some additional more context-specific semantic gap issues that warrant consideration in the context of this study.

The first issue is the presence of negated language (e.g., *denies fever* or *no fracture*) and references to family history (e.g., *history of breast cancer in their family*). From an information retrieval perspective, negation may adversely affect search effectiveness (Koopman et al, 2010; Limsopatham et al, 2012). Negation is a well understood problem (Chapman et al, 2001) and there are specific IR methods that have proven effective in handling negation (Limsopatham et al, 2012; Koopman and Zuccon, 2014b). Negated content is detected by certain negation identifiers: terms such as *no*, *denies*, *without*, etc. If these negation identifiers are observed, then one can conclude that the concept following them is negated; therefore, the conclusion is derived deductively and deductive inference is the mechanism required to handle negation and family history.

Temporality is important issue in medical IR (Koopman and Zuccon, 2014c,a). In clinical patient records, there are often references to a patient's past medical history. While some of this content may be relevant to the patient's current condition (e.g., chronic conditions), others may no longer apply (e.g., acute conditions). An IR system may retrieve a patient record based on the terms found in the past medical history section, but the relevance of the record is dependent on whether the past conditions or treatments still apply to the patient or are dependent on the context of the query.

The age and gender of the patient can have an important bearing on relevance (Voorhees and Tong, 2011; Voorhees and Hersh, 2012). Some information needs require specific age and gender characteristics (e.g., *elderly woman*). In this case it would be important to understand that *elderly* implies age > 65 and *woman* implies *female*.

Finally, clinical records are often made up of different levels of evidence, often conveyed through different types of reports: history and examination reports for initial consultations, laboratory test results during the patient's treatment, and discharge summaries authored as a retrospective review of the patient's care. These different report types convey different information and therefore affect the way relevance is determined when query terms are found in each (Zhu and Carterette, 2012; Limsopatham et al, 2013b).

In this paper, our focus is on the first four, non context-specific, more general semantic gap issues — vocabulary mismatch, granularity mismatch, conceptual implication and inferences of similarity.

3 The Graph INference Model (GIN)

The GIN comprises two components: (1) a graph-based representation combining structured domain knowledge with corpus statistics; and (2) an inference mechanism that traverses the graph.

3.1 Graph-based Representation of a Corpus

The basis of the graph representation is an *Information Unit.*

Definition 1 Let $\mathbb{U}$ denote a non-empty set of Information Units.

An Information Unit $u \in \mathbb{U}$ is an abstract notion, e.g., an entity or concept defined in an ontology or controlled vocabulary. Alternatively, an Information Unit may be derived as a result of an information extraction process (e.g., a Person or a Place), or be an n-gram or term phrase (like those extracted by Bendersky and Croft (2008)). In its most basic form, an Information Unit could be a single term.

Information Units are related to each other in a many-to-many relationship:

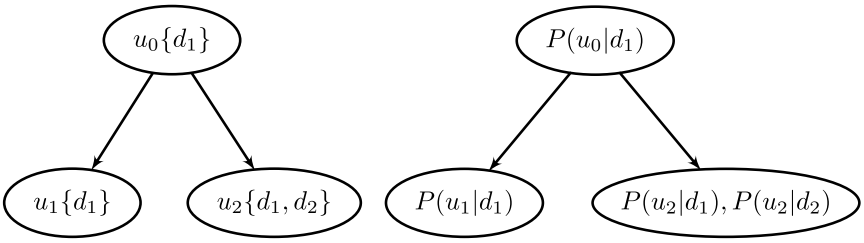
Definition 2 Let $\mathbb{R} \subseteq \mathbb{U} \times \mathbb{U}$ define a non-empty set of Information Relationships.

If the Information Units come from an ontology or thesaurus, the relationships may be explicitly pre-defined. This is the case for SNOMED CT[3], which includes explicit relationships between concepts. For some other types of Information Unit, such as terms or n-grams, Information Relationships may be determined by term co-occurrences. Other implementations may link Information Units that are semantically similar to each other. The particular implementation will most likely impose further restrictions on $\mathbb{R}$; for example, if the relationships are taken from SNOMED CT, which can be represented as a directed acyclic graph, then $\mathbb{R}$ would be irreflexive and antisymmetric. In the remainder of this paper we shall consider Information Relationships as directed. We adopt the notation uRu' to denote the existence of an Information Relationship between u and u'.

Information Relationships may belong to one or more Relationship Types.

Definition 3 Let $\mathbb{T}$ denote a set of Relationship Types.

[3] SNOMED CT is a widely adopted medical ontology; more details are provided in Section 4.1.



(a) Basic node-document representation.

(b) Representation with initial probabilities assigned to node.

Fig. 1 Example graph-based corpus representation.

A Relationship Type ($t \in \mathbb{T}$) could be taken from the relationship types found in medical ontologies such as SNOMED CT, (for example, *causative agent* or *active ingredient*). Each Information Relationship may belong to one or more Relationship Type according to a Type relationship.

Definition 4 Let T be a total function which maps Information Relationships to Relationship Types, $T : \mathbb{R} \rightarrow \mathbb{T}$.

Based on the above definitions, a graph can be defined where Information Units represent vertices or nodes[4] and Information Relationships represent the edges between Information Units. If Information Units and Information Relationships are, respectively, SNOMED CT concepts and relationships, then the resulting graph is simply the SNOMED CT ontology represented as a graph. This representation is employed in the implementation considered in this paper. An Information Graph is defined as follows:

Definition 5 Let $G = \langle \mathbb{U}, \mathbb{T}, T, \mathbb{R} \rangle$ denote an Information Graph.

The inclusion of queries and documents into this graph provides a representation that facilitates retrieval by inference.

Definition 6 A document d (query q) is a sequence of Information Units: $d = \langle u_0, ..., u_n \rangle$ ($q = \langle u_0, ..., u_m \rangle$).

An Information Graph can be used to model an entire corpus by first constructing a graph with Information Units as nodes and Information Relationships as edges and then attaching to each node the list of documents or the query in which that Information Unit appears. An example graph created using this approach is provided in Figure 1(a). In the remainder of this paper, "document nodes" and "query nodes" refers to Information Units contained in a document and query respectively.

Rather than just attaching documents and queries to a node, a weight or initial probability can be assigned. We call this an initial probability because it is assigned prior to retrieval and is independent of the query. Figure 1(b)

[4] In the following, Information Units and nodes will be used interchangeably.

shows how the graph is modified to store in a node the likelihood of the corresponding Information Unit within a document. Note that although the figure shows only the initial probability for the document attached to the node, the initial probability of each Information Unit in each document may be estimated for all documents in the collection if a smoothing process is used. How these probabilities are estimated is not constrained by the model and is an implementation-specific decision.

3.2 Diffusion Factor

The *diffusion factor* models the strength of association between two Information Units.

Definition 7 Let δ be a *recursive* function $\delta : \mathbb{U}\times\mathbb{U} \to \mathbb{R}^+$ (the set of positive real numbers) that denotes the maximal diffusion between two Information Units, $u, u' \in \mathbb{U}$ such that:

$$\delta(u,u') = \begin{cases} 1, & \text{if } u = u' \\ \delta_0(u,u'), & \text{if } uRu' \\ \arg\max_{u_i \in \mathbb{U}:uRu_i} \delta(u,u_i) \otimes \delta(u_i,u'), & \text{otherwise.} \end{cases} \tag{1}$$

Line 1 represents the case of diffusion between a node and itself; line 2 represents the base case when there is a direct edge (uRu') between u and u'; line 3 represents the recursive case whereby diffusion is calculated for other nodes, u_i, connected to u.

The definition of $\bigotimes$ operator is implementation-dependent. However, if the diffusion factor is implemented using a probability, then the probabilities can be multiplied to combine diffusion factors:

$$\delta(u,u') = \begin{cases} 1, & \text{if } u = u' \\ \delta_0(u,u'), & \text{if } uRu' \\ \arg\max_{u_i \in \mathbb{U}:uRu_i} \delta(u,u_i) \cdot \delta(u_i,u'), & \text{otherwise} \end{cases} \tag{2}$$

Other alternative implementations for the $\bigotimes$ operator could take into account the actual number of transitions for estimating the diffusion or could implement the overall diffusion factor as the maximum or minimum value of the individual diffusion factors.

The arg max operator accounts for the case of multiple paths to transition between u and u'. In this case, the path with the greatest diffusion factor (least effort) is favoured.

Although not imposed by the general definition, the diffusion factor can be calculated in a number of different ways, both using corpus-based techniques and domain knowledge. For corpus-based techniques, a semantic similarity measure (e.g., Pointwise Mutual Information), would capture the strength of association between two connected Information Units, u and u_{i-1}; we denote this strength $\mathrm{sim}(u_{i-1}, u_i)$. For domain knowledge-based techniques, the Relationship Type would capture some measure of association; we denote this strength $\mathrm{rel}(u_{i-1}, u_i)$. The base case of the recursive diffusion factor (δ_0) between u and u' with uRu' can be estimated as a linear interpolation of the two functions:

$$\delta_0(u, u') = \alpha \ \mathrm{sim}(u, u') + (1 - \alpha) \ \mathrm{rel}(u, u') \quad 0 \leq \alpha \leq 1 \tag{3}$$

where the parameter α is the *diffusion mix* of the similarity and Relationship Type measure.

3.3 Retrieval Model

Given a query q, the GIN models retrieval as an inference process: the relevance of a document d is determined by the amount of evidence to support the implication $P(d \rightarrow q)$. This evidence is drawn from Information Units connected to the query nodes. Let $\mathbb{C} \subset \mathbb{U}$ be the set of Information Units connected to the query Information Units by means of one or more edges. Considering the simplest case of a document containing a single Information Unit u_d and a query containing a single u_q which is only connected to u_d (i.e., $\mathbb{C} = \{u_d\}$), then the relevance of d to q is given by

$$P(d \rightarrow q) = P(u_d \rightarrow u_q) \propto P(u_d|d) \, \delta(u_d, u_q).$$

where $P(u_d|d)$ is the initial probability (strength of the Information Unit u_d in the document); while $\delta(u_d, u_q)$ is the diffusion factor (how strongly u_d and u_q are associated).

Having provided a means of evaluating $P(u_d \rightarrow u_q)$, we can now consider the more general problem of inferring the query from the document, i.e., $P(d \rightarrow q)$. The single Information Unit inference definition can be extended to that of query and document by evaluating each combination of query Information Unit $u_q \in q$ and document Information Unit $u_d \in d$:

$$\begin{aligned} P(d \rightarrow q) &= \bigodot_{u_q \in q} \ \bigsqcup_{u_d \in d} P(u_d \rightarrow u_q) \\ &\propto \bigodot_{u_q \in q} \ \bigsqcup_{u_d \in d} P(u_d|d) \, \delta(u_d, u_q). \end{aligned} \tag{4}$$

This is the general retrieval function of the Graph Inference model. It has two placeholders for operators: $\bigodot$, for Information Units in the query and $\bigsqcup$, for Information Units in the document. Their definitions are left to the specific implementation but we consider two possible alternatives here. First, if the query Information Units are assumed independent (as is the case for many retrieval models) and the document Information Units are also considered independent, then the probabilities are multiplied; therefore $\bigodot = \prod$ and $\Box = \prod$ to derive the retrieval status value function:

$$\mathrm{RSV}(d, q) = \prod_{u_q \in q} \prod_{u_d \in d} P(u_d|d)\, \delta(u_d, u_q). \tag{5}$$

In this implementation, the Information Units u_i, related to u_q, are considered as additional information regarding the query, with the diffusion factor controlling the strength of association between the two. This is akin to the query expansion process where additional query terms are derived. The implementation shown above in Equation 5 is similar to the approach used in probabilistic language modelling.

An alternative implementation is still to consider query Information Unit as independent but to consider the document Information Units as dependent. In this case, the query placeholder $\bigodot$ is a product ($\bigodot = \prod$), thus multiplying the independent query Information Units, but the related Information Units in the document are summed ($\Box = \sum$). This gives the retrieval status value function:

$$\mathrm{RSV}(d, q) = \prod_{u_q \in q} \sum_{u_d \in d} P(u_d|d)\, \delta(u_d, u_q). \tag{6}$$

In this case, the Information Units related to u_q via the graph represent an alternative representation of the query Information Unit u_q and provide an additional source of supporting evidence (albeit a weaker source according to the discounting applied by the diffusion factor).

The general retrieval function from Equation 4 can be applied in a number of different ways; two are presented above but others are possible. Figure 2 shows a number of different possible implementations. The Graph Inference model intentionally generalises these operators so a particular implementation is not imposed by the model. This means that the model can be applied to a number of different scenarios, making it a general model from which particular inference-based retrieval models can be instantiated.

3.4 Worked Retrieval Example

Consider a query $q = \langle u_q \rangle$ and three documents $d_1 = \langle u_1, u_2, u_q \rangle$, $d_2 = \langle u_3, u_q \rangle$, $d_3 = \langle u_4 \rangle$. Figure 3 shows the retrieval process and also illustrates the graph representation of this corpus: u_q represents the query and is indicated as a square node; other Information Units found in documents are elliptical. Documents are attached to the nodes they encompass, along with an initial probability $P(u_i|d_j)$. The edges between nodes are based on some source of domain knowledge resource (e.g., ontology relationships). Edges are labelled with the score that the source node contributes to that document. Each sub-figure represents the scoring process for the three documents. Grey nodes indicate Information Units not present in the document and that thus contribute only the background smoothing probability. Black nodes represent

$$P(d \rightarrow q) \propto \overset{\textstyle \prod \quad \sum \quad \cdots}{\overset{\searrow \; \downarrow \; \swarrow}{\underset{u_q \in q}{\bigodot}}} \; \underset{\substack{u_d \in d \\ \nearrow \; \uparrow \; \nwarrow \\ \textstyle \prod \quad \sum \quad \cdots}}{\square} \; P(u_d|d)\, \delta(u_d, u_q).$$

(a) Retrieval Function

$$\delta(u, u') = \begin{cases} 1, & \text{if } u = u' \\ \delta_0(u, u'), & \text{if } uRu' \\ \arg\max_{u_i \in \mathbb{U}: uRu_i} \delta(u, u_i) \overset{\textstyle \prod \; \cdots}{\overset{\searrow \swarrow}{\otimes}} \delta(u_i, u'), & \text{otherwise} \end{cases}$$

(b) Diffusion Factor

Fig. 2 Possible implementation options for the Graph Inference model retrieval function and diffusion factor.

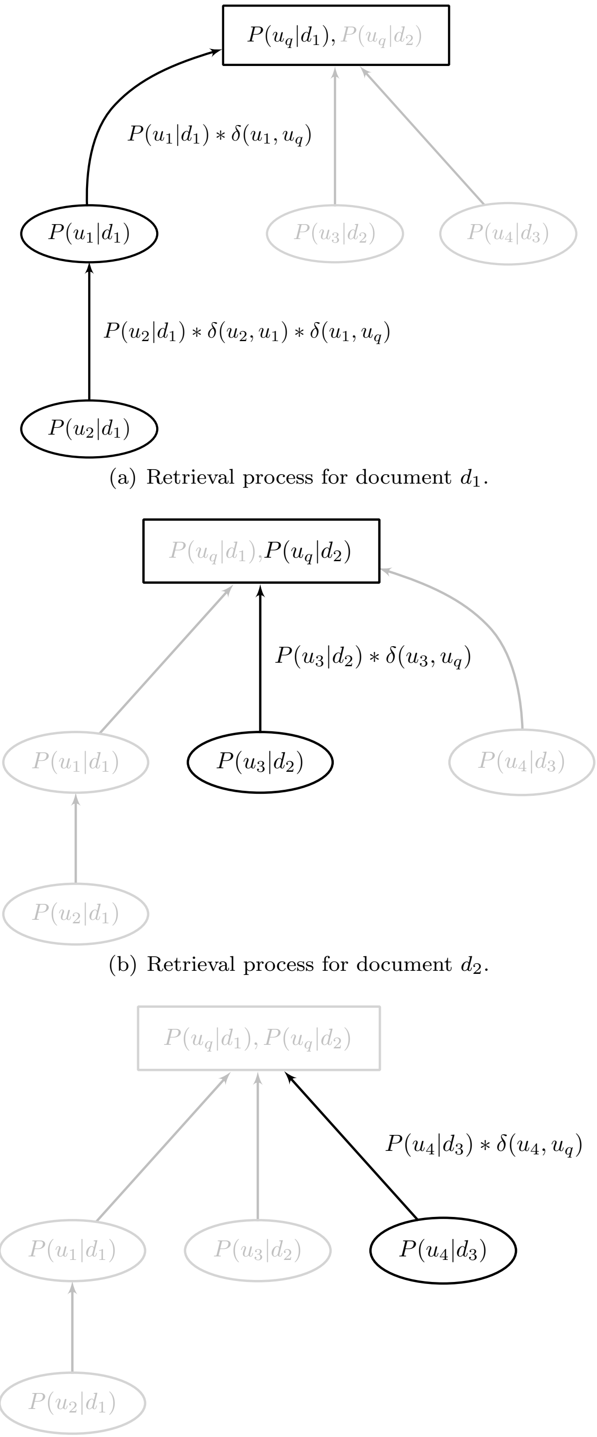
Information Units in the document. For sake of simplicity, we focus on only those Information Units present in that particular document and their contribution to the retrieval score.

Figure 3(a) shows the graph traversal used to score d_1. The score for d_1 is the result of three sources of evidence (excluding the background smoothing contribution). Firstly, d_1 contains the query Information Unit u_q, thus receiving the contribution $P(u_q|d_1)$. Secondly, d_1 also contains u_1, which is related to the query u_q: d_1 thus receives $P(u_1|d_1)$ but discounted by the diffusion factor $\delta(u_1, u_q)$. Finally, d_1 also contains u_2, related to u_q via u_1; this evidence contributes $P(u_2|d_1) * \delta(u_2, u_1) * \delta(u_1, u_q)$ to the score of d_1. It is the combination (by multiplication) of these three sources of evidence that determines the score of d_1 under the GIN. Most IR models would consider only the first estimate, $P(u_q|d_1)$.

Figure 3(b) illustrates the process for d_2. Only two sources contribute to the score of d_2: $P(u_q|d_2)$, because the document contains the query; and $P(u_3|d_2) * \delta(u_3, u_q)$, because d_2 contains one other Information Unit related to the query through the edge u_3–u_q. Both documents d_1 and d_2 contain the query and Information Units related to the query. However, d_1 contains additional evidence in the form of u_2.

Figure 3(c) illustrates the process for d_3. This document does not contain any *query* Information Units, but it does contain u_4, which is related to the query. Most IR models would ignore this document.[5] However, d_3 is retrieved by the GIN, which assigns to d_3 the score $P(u_4|d_3)$ discounted by the association between u_4 and u_q (i.e., $\delta(u_4, u_q)$).

[5] Theoretically, most IR models do not impose the restriction that only documents containing a query term should be returned. In practice, however, they typically score only documents that contain at least one query term.



(a) Retrieval process for document d_1.

(b) Retrieval process for document d_2.

(c) Retrieval process for document d_3.

Fig. 3 Retrieval process for three example documents using Graph Inference model.

4 Model Implementation

4.1 Domain Knowledge Resource: SNOMED CT

For our implementation, the definitions of Information Units and Relationships are taken from the SNOMED CT ontology and this is used as the underlying structure to generate the graph representation of the corpus. SNOMED CT encodes a wide variety of medical knowledge within a concept inheritance hierarchy, with relationships (Information Relationships in our model) connecting concepts (Information Units). While other resources could have been used, such as the UMLS (another large medical domain knowledge resource), SNOMED CT was chosen because it contains a wide range of medical knowledge in a single, self contained resource, whereas UMLS is in fact a conglomeration of different resources (meta-ontology), each with varying coverage. SNOMED CT also has a rigorous quality control process.

4.2 Mapping Terms to Concepts

A method is required to transform the free-text content of documents and queries into the Information Units (SNOMED CT concepts) of our graph representation. This is achieved using MetaMap (Aronson and Lang, 2010), a medical information extraction system. MetaMap is widely adopted in medical NLP (Aronson and Lang, 2010; Nadkarni et al, 2011) and has proven effective for medical concept identification (Pratt and Yetisgen-Yildiz, 2003). A number of concept-based IR methods have been developed using MetaMap; some of these have been shown to outperform pure term-based systems (Koopman et al, 2012b; Limsopatham et al, 2013a,c). We follow the approach detailed by Koopman et al (2012b) for mapping free-text into SNOMED CT concepts using MetaMap.

4.3 Indexing

After mapping the terms to SNOMED CT concepts, the documents are processed using a standard IR indexer to create an inverted file index. This index forms the input, together with the chosen structured domain knowledge resource (SNOMED CT), of the GIN's indexing process detailed in Algorithm 1. Using this method, each concept in the index becomes a node in the graph. The graph also contains many additional nodes representing concepts not in the corpus but related (via the ontology) to concepts that are in the corpus. These can provide additional domain knowledge at retrieval time and could link two concepts that appear in the corpus but have no direct edge between them.

Algorithm 1 Pseudo code for efficient GIN indexing.

```
Input: Idx, Ont                                          ▷ Index, Ontology
Output: $G = \langle V, E \rangle$                         ▷ Graph (vertices and edges)
for $u_i \in$ Idx do
    $v_i$ = CREATE_VERTEX($u_i$)
    for $u' \in$ related_concepts(Ont, $u_i$) do
        $v'$ = CREATE_VERTEX($u'$)
        diffusion = $\delta(u_i, u', \alpha)$            ▷ Calculate diffusion factor
        $e_i$ = CREATE_EDGE($v_i$, $v'$, diffusion)
serialize_graph(path(Idx), $G$)
function CREATE_VERTEX($u$)
    $v$ = vertex($u$)
    if $v \notin V$ then
        $V = V + v$                                    ▷ Add node to graph
        return $v$
function CREATE_EDGE($v_1$, $v_2$, diffusion)
    if ($v_1$, $v_2$, diffusion) $\notin E$ then
        $e$ = edge($v_1$, $v_2$, diffusion)
        $E = E + e$                                    ▷ Add edge to graph
        return $e$
```

Diffusion Factor

The diffusion factor between two concepts is a linear interpolation of two measures: semantic similarity and Relationship Type, as shown in Equation 3. In our implementation, similarity was estimated as the cosine angle between two concept document vectors, as this proved to be the most robust and effective method in a study of corpus-based measures for medical concept similarity (Koopman et al, 2012a).

The second component of the diffusion factor is the Relationship Type weighting. Relationship Types are taken directly from SNOMED CT, which has explicit relationships between concepts; for example, *ISA*, *causative agent* or *finding site*. These different Relationship Types can indicate a strength of association: an *ISA* relationship might indicate a strong relationship between two concepts, whereas relationships such as *severity* indicate a much weaker association. On initial examination, however, we found that SNOMED CT contained mostly *ISA* relationships for the collection used in our experiments (Figure 4). Thus, Relationship Types were not a discriminating enough feature for inclusion in the diffusion factor, motivating us to ignore the Relationship Type component (by setting $\alpha = 1$ in Equation 3).

4.4 Retrieval

The GIN's retrieval process traverses the graph created when indexing the corpus. In our implementation, we use a standard Dirichlet-smoothed language model to estimate the initial probabilities $P(u_d|d)$ at retrieval time, although alternative weighting measures could have been used (e.g., BM25, Divergence from Randomness, TF-IDF, etc.).

The retrieval function evaluates the relevance of a particular document d to a query q, but it does not consider which documents are chosen for scor-

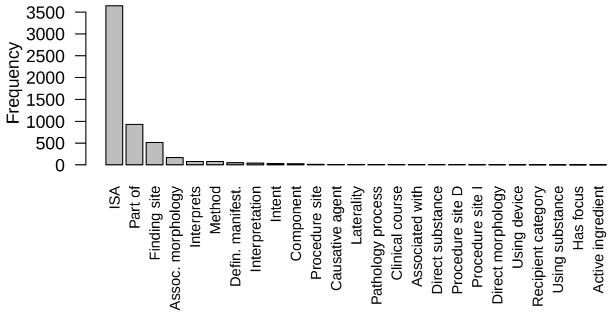


Fig. 4 Frequency of relationships types that connect TREC MedTrack query concepts to other concepts in SNOMED CT. ISA relationships are by far the most common relationship type.

ing. Evaluating all documents in the collection against a query is infeasible, so a subset of possibly relevant documents is required for evaluation. In other retrieval models, this is often simply determined by those documents that contain at least one query term. However, the GIN has the ability to score potentially relevant documents that do not contain the query but may contain information related to the query (see document d_3 in the example of Section 3.4). For feasibility reasons, an alternative method is therefore required to limit which documents should be scored using the GIN. The observation can be made that according to Equation 2 the diffusion factor decreases rapidly when the node is further from the query. Beyond a certain point, the diffusion factor is so small that a document is not worth considering because its probability is insignificant once weighted by the diffusion factor. As a result, we need to consider only those documents attached to Information Unit nodes k edges away from the query node. Retrieval can therefore be modelled as a depth-first-search (DFS), originating from the query node, visiting only nodes k edges away. This process is detailed in Algorithm 2.

Algorithm 2 Pseudo code for depth-first-search retrieval.

```
Input: Idx, $Q, G, k$                                  ▷ Index, Query, Graph, Max depth
Output: $scores \leftarrow \{d_0, \ldots, d_n\}$        ▷ Document scores
for $u_q \in Q$ do
    DFS($u_q$, 0)                                       ▷ Start traverse from query node, depth 0
function DFS($u$, $depth$)
    if $depth \leq k$ then
        for $d_i \in$ Idx.docs($u$) do                  ▷ Docs. containing $u$
            $scores[d_i] = scores[d_i] + P(u|d_i) * \delta(u, u_q)$
        for $u' \in$ children($u$) do
            DFS($u'$, $depth + 1$)                      ▷ Recursively traverse children
```

When the maximum depth parameter k is set to 0, then the algorithm processes only query nodes and does not traverse any edges. In this case, if the initial probabilities are Dirichlet smoothed estimates, then $k = 0$ represents a standard probabilistic language model with Dirichlet smoothing, constituting a benchmark for comparison.

The GIN was implemented in C++ with the indexing and retrieval components implemented using the Indri library[6] and constructing graphs using the LEMON library.[7] The graph was serialised using LEMON and stored inside the Lemur index directory.

5 Empirical Evaluation

We start by describing our experimental setup using the TREC Medical Records Track (MedTrack).

Then Experiments 1 describes the results from the standard TREC Medtrack setup. This experiment also reveals that the GIN returned many documents never judged by TREC relevance assessments, which may have significantly affected the evaluation measures. As a result we describe Experiments 2 — additional relevance assessments from medical professionals to understand to what extent the GIN was retrieving new relevant documents.

For all experimental results, we consider easy and hard queries separately in order to understand the effect of inference on each.

5.1 Experimental Setup

The test collection used in our experiments was the TREC 2011 & 2012 Medical Records Track (Voorhees and Hersh, 2012; Voorhees and Tong, 2011). TREC MedTrack contained 100,866 clinical patient records of various types (pathology, radiology, discharge summaries, etc.) from U.S. hospitals. The task description for TREC MedTrack was to identify cohorts of patients matching a specific query criteria for inclusion in a clinical trial. TREC MedTrack models the clinical task of cohort identification as an adhoc retrieval task. In this task, queries are clinical trial inclusion criteria; documents are records representing a particular patient. Further details about the specific task and data for TREC MedTrack are provided by Voorhees and Tong (2011) and Voorhees and Hersh (2012). The track guidelines stipulated that the unit of retrieval was a patient record rather than an individual report; reports belonging to a single patient's record were treated as sub-documents and concatenated into a single document called a patient *visit* document.[8] The resulting corpus contained 17,198 patient visit documents. Full details of the corpus statistics, after indexing with the GIN, are provided in Table 1.

The evaluation measures used in MedTrack 2011 were bpref and precision @ 10 (P@10). However, in MedTrack 2012 inferred measures and P@10 were used.

[6] http://lemurproject.org.

[7] http://lemon.cs.elte.hu.

[8] Collapsing reports to patient visits was a common practise among many MedTrack participants (Voorhees and Hersh, 2012; Voorhees and Tong, 2011).

Table 1 Corpus statistics after indexing the concept-based TREC Medtrack collection using the GIN.

Number of documents	17,198
Vocabulary size	36,467
Average document length (tokens)	3,906
GIN graph:	
Number of nodes	49,153
Number of edges	99,161
Average degree (edges per node)	2.02
Serialised graph size	4.4MB

Inferred measures required specific relevance assessments (prels) not available for 2011, but bpref and P@10 could be used for 2012 as qrels were available. While it is possible to separate the evaluation into two parts (34 queries for 2011 and 47 for 2012), it is more desirable to have a single, larger query set for more powerful statistical analysis. Therefore, we combine the query sets and use bpref and P@10.

The depth parameter k controls how many edges are traversed from the query node. It is a key parameter underpinning the retrieval process — the higher the k the more the inference as k represents the length of the path traversed by the GIN. For this reason it is a focal point in the evaluation. Consequently, k was manipulated at $k = 0$ (lvl0), $k = 1$ (lvl1) and $k = 2$ (lvl2), reflecting deepening levels of inference from the query node.[9] To further understand how the traversal depth affects retrieval effectiveness, we varied $k = [1, .., 10]$ on a per-query basis.

Lvl0 reflects the situation when only the query nodes are processed, which equates to a concept-based baseline. For additional comparison, we also include a standard term baseline — also using a Dirichlet-smoothed language model. All these models — lvl0 baseline, GIN lvl1 and lvl2, and terms — contain the Dirichlet smoothing parameter μ. This parameter was tuned with respect to bpref for the two baselines (lvl0 and terms), performing a linear search of the parameter space $[0, 30000]$ (with increments of 1,000) over the whole query set[10]. The GIN at lvl1 and lvl2 shared the same setting of μ as lvl0. In this way, lvl0 represents a strong, tuned baseline, whereas the GIN is not tuned to avoid overfitting.

5.2 Experiment 1 - MedTrack

The goal of this experiment was to demonstrate the effect that different levels of inference within the GIN had on retrieval effectiveness. In addition, we investigated whether inference was more effective for “hard” queries: those queries exhibiting poor performance across systems participating in MedTrack.

[9] Retrieval effectiveness degraded on *average* for $k > 2$.

[10] Settings of μ were: lvl0, 22,000; terms, 13,000.

Table 2 GIN retrieval results using MedTrack. †=paired t-test against lvl0, $p < 0.05$. Hard queries were defined as half the query set with the lowest median bpref across all teams participating in TREC Medtrack.

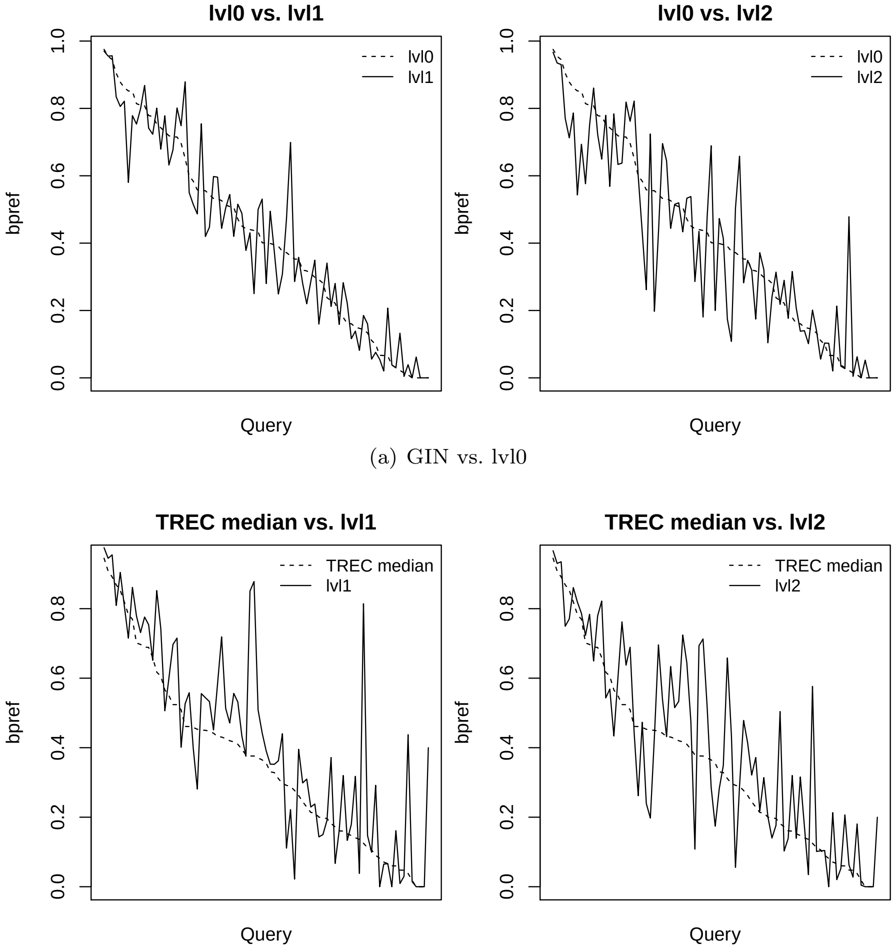
Depth (k)	**All Queries**		**Hard (TREC Median)**	
	Bpref	P@10	Bpref	P@10
terms	0.3917	0.4975	0.1866	0.2650
lvl0	0.4290	0.5123	0.1985	0.2800
lvl1	0.4229	0.4481†	0.2024	0.2425
lvl2	0.4138	0.4259†	0.2072	0.2275

To this end, we computed the median bpref of all submissions in MedTrack (2011 and 2012 combined); hard queries were defined as half the query set (40 out of 81) with the lowest bpref value.

Table 2 shows the retrieval results for each of the three depth settings and for the term baseline. Both bpref and P@10 were lower for the GIN (lvl1 and lvl2) compared against the concept baseline (lvl0). To further understand the differences between the three levels, the retrieval effectiveness of individual queries is shown in Figure 5(a). Queries are ordered by decreasing bpref of the lvl0 baseline. The plots show that both lvl1 and lvl2 made gains on some queries and losses on others. The gains and losses tended to be greater for lvl2 than for lvl1. Figure 5(b) shows how the GIN compared with the TREC median performance. More gains were observed for hard queries. To further quantify this, we considered the performance of only hard queries shown in Table 2. Even though Table 2 shows marginal increases in bpref, the query-by-query results of Figures 5(a) & 5(b) show improvements for the majority of hard queries, some of which exhibit considerable increases. The table confirms that the GIN made greater improvements on hard queries and that these improvements were greater when more of the inference mechanism is applied (i.e., for the GIN at lvl2).

Bias in evaluation

Empirically, the GIN did not demonstrate statistically significant improvements over the concept baseline (lvl0), but this does not constitute the whole story. A large number of unjudged documents — those never assessed by TREC judges — were retrieved by the GIN. Considering the top 20 documents returned for a query, the number of unjudged documents for lvl1 and lvl2 was respectively 2.3 and 3 times greater than that of the term baseline (see Table 3). Such a large number of unjudged documents can significantly affect the evaluation measures and underestimate the GIN's performance. For precision, an unjudged document is considered not relevant; thus greater numbers of unjudged documents will lower precision. Our results showed that P@10 was significantly lower for the GIN than the concept baseline. In contrast, the bpref measure ignores unjudged documents; this was reflected in our results where bpref differed only slightly between models.



(a) GIN vs. lvl0

(b) GIN vs. TREC median

Fig. 5 Bpref performance (y-axis) of each individual query (x-axis). Queries are ordered by decreasing bpref of the baseline (lvl0 for (a) and TREC Median for (b)); therefore, easy queries are on the left of the x-axis and hard queries are on the right.

The motivation for using the GIN's inference mechanism is that it may retrieve additional relevant documents that are not retrieved by keyword-based approaches. We conjecture that part of the unjudged documents retrieved using the GIN were in fact relevant but were never included in the pool — a pool constructed from largely keyword-based systems (Voorhees and Tong, 2011; Voorhees and Hersh, 2012). Therefore, we obtained additional relevance assessments from medical professionals to understand to what extent the GIN was retrieving new relevant documents.

Table 3 Number of unjudged documents in top 20 positions and P@20 for different retrieval models.

Model	**Unjudged documents in top 20 results**	**P@20**
Terms	210 (2.5 docs / query)	0.4244
Concept baseline (lvl0)	257 (3.0 docs / query)	0.4389
GIN lvl1	468 (5.5 docs / query)	0.4086
GIN lvl2	616 (7.2 docs / query)	0.3630

5.3 Experiment 2: Additional Qrels

5.3.1 User Experimental Design

We recruited four 4th-year medical students from the University of Queensland. As part of their training, they had completed rotations in a number of different medical specialities and, as such, their expertise was equivalent to medical graduates recruited as assessors in MedTrack (Voorhees and Tong, 2011; Voorhees and Hersh, 2012).

For each query we proposed to judge a selection of documents that had not previously been judged in MedTrack. These documents were selected by pooling the unjudged documents from the top 20 results of three retrieval runs: (1) the concept baseline model (lvl0) (2) the GIN lvl1; (3) the GIN lvl2. Using this method, complete judgements were obtained for the top 20 documents returned for each query by each of the three systems listed above.

The task description given to assessors was the same as that of the original MedTrack task. To familiarise the assessors with the judging task, they were first given documents from two control queries. The control queries contained a selection of both unjudged documents and those already judged by TREC assessors. In this way, they could be used to determine inter-coder agreement — both amongst our assessors and against the original TREC assessors.

A total of 1030 documents were judged. Inter-coder agreement between the four assessors (based on the two control queries) was 0.85. This is in line with an inter-coder agreement of 0.8 found by the MedTrack organisers.[11] Agreement between the four assessors and the TREC assessors was 0.80. These new qrels are made available at `https://github.com/ielab/MedIR2014-RelanceAssessment`; additional analysis of both the new qrels and the actual assessment task are detailed in Koopman and Zuccon (2014c).

Of the 1030 documents judged, 29% were found to be relevant. In comparison, the original relevance assessments provided by TREC contained only 18% relevant documents. Therefore, the pool of documents from our systems (lvl0, lv1 and lv2) contained more relevant documents than the pool of documents provided by systems participating in TREC.

[11] Based on personal communication with Bill Hersh, MedTrack organiser, 29 May 2013.

Table 4 GIN retrieval results using old (TREC) and combined (TREC++) qrels. The percentages and † indicate how the measure changed using the two different qrels (†=paired t-test $p < 0.05$).

Qrel set	Sys	All Queries		Hard Queries	
		P@10	P@20	P@10	P@20
TREC	lvl0	0.5123	0.4389	0.2800	0.2150
	lvl1	0.4481	0.4086	0.2425	0.2025
	lvl2	0.4259	0.3630	0.2275	0.1988
TREC++	lvl0	0.5415† (+6%)	0.4732† (+8%)	0.3025† (+6%)	0.2387† (+11%)
	lvl1	0.5037† (+12%)	0.4604† (+12%)	0.2850† (+18%)	0.2475† (+22%)
	lvl2	0.4878† (+15%)	0.4220† (+16%)	0.2775† (+22%)	0.2438† (+23%)

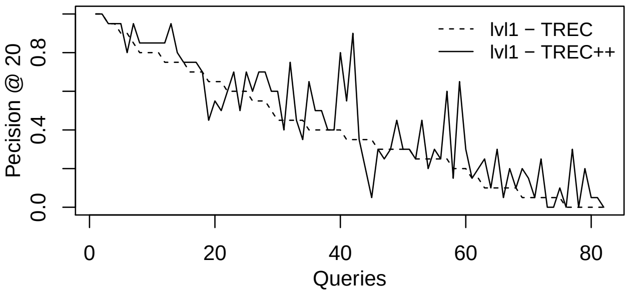


Fig. 6 Per-query performance of GIN lvl1 using old (TREC) and new qrels (TREC++).

5.3.2 Results

Table 4 presents the retrieval results of the GIN (lv1, lv2) and the concept baseline (lvl0) using the old qrels (TREC) and the new qrels (TREC++). The percentages indicate how the measure has changed between the old and new qrels. Considering bpref, there was little change in overall effectiveness using the new qrels. This is not surprising as bpref considers only judged documents so the large number of unjudged documents in the TREC qrels did not significantly affect this evaluation measure. However, for P@10 and P@20, all three systems were found more effective when evaluated with the new qrels. The effectiveness was underestimated for all three systems (lvl0, lvl1 and lvl2) but was significantly more so with the GIN. Furthermore, lvl2, which leverages more of the GIN inference mechanism, was underestimated more than lvl1. This means that lvl2 was returning a larger number of unjudged but relevant documents.

Considering only P@20, Figure 6 shows how the performance of individual queries changed between the old and new qrels. A significant number of queries had improved performance using the new qrels, with only a handful showing degradation. Additionally, a greater number of improvements was observed in hard queries (righthand side of the plot). This highlights that hard queries were the ones where performance was most underestimated.

Overall, when considering P@20, the GIN at lvl1 outperformed the lvl0 baseline for hard queries; although this results was not found to be statistically significant and improvements were only observed for hard queries not all queries (from Table 4). To understand why this is the case, and to reveal deeper insights into when inference was working or not, we provide a detailed query-by-query analysis in the section that follows.

6 Analysis

To understand the effect of the depth parameter, retrieval effectiveness using different settings of $k = [1, .., 10]$ was examined on a per-query basis. The heatmap in Figure 7 shows the change in bpref compared to the lvl0 baseline for different settings of k. Dark areas indicate that effectiveness improved for that setting of k when compared to lvl0 ($k = 0$), while light areas indicate that effectiveness degraded when compared to lvl0. There is considerable variation between queries. Some queries had a constant improvement over lvl0 for different depth settings, for example queries 108, 140 and 171. Other queries degraded as the depth increased, for example 104, 109 and 161. Some queries improved over lvl0 in the first few levels but then degraded at greater levels, for example 113, 119 and 135. Generally, the best improvements were observed for k = 1–3. Finally, the optimal value of k varied considerably based on the query.

The heatmap was used to group queries according to the performance results that they exhibit at different depth settings. Next, we analyse such groupings to understand how inference in the GIN works and under which conditions.

6.1 Consistent Improvement

A number of queries exhibited a consistent improvement over the baseline for different depth settings (see Figure 8 for the performance of two example queries). These types of queries tended to have relevant related concepts traversed by the GIN at levels greater than 0. For example, Figure 9 shows the partial traversal graph (with associated annotations for explanation) for query 171, which seeks patients with a specific disease (Thyrotoxicosis). The GIN was able to infer other relevant documents that contained the cause of Thyrotoxicosis (Hyperthyroidism) and the part of the body affected (Thyroid structure). Including these relevant related concepts always improved performance over the lvl0 baseline. In addition, the diffusion factors were effective at limiting the introduction of noise for greater levels and as a result no degradation was seen for levels up to 10.

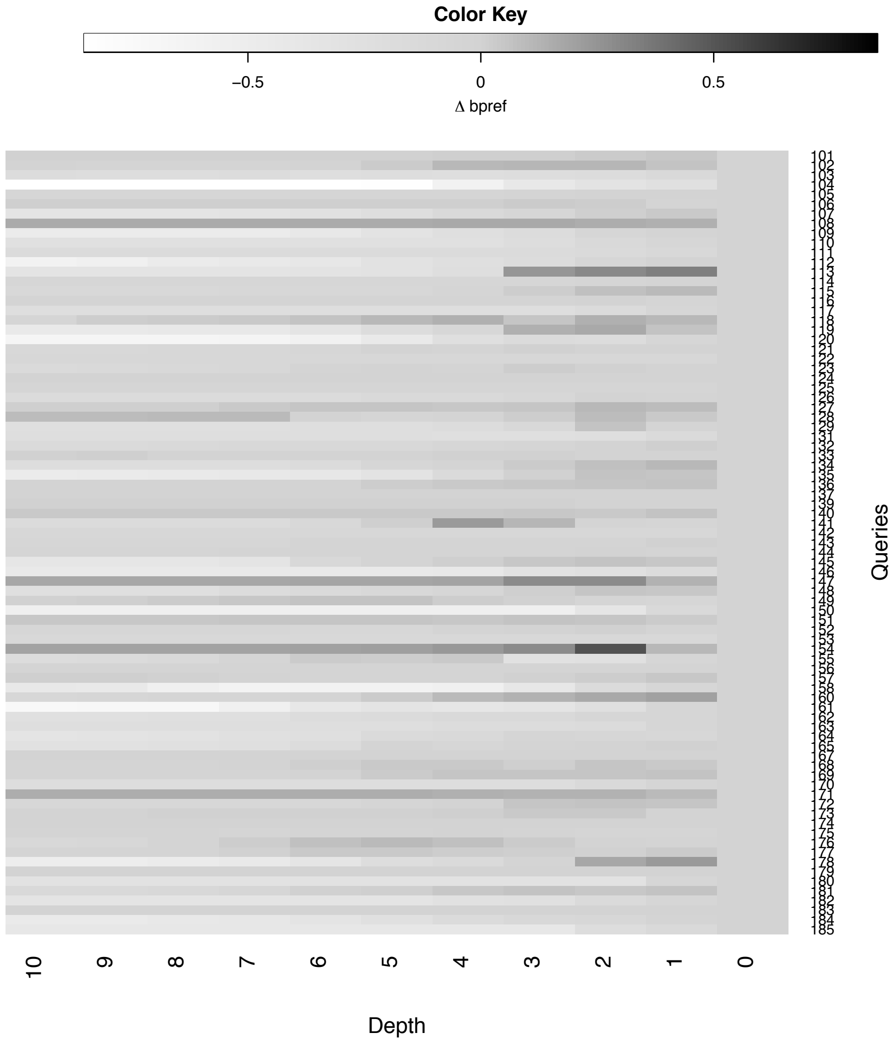
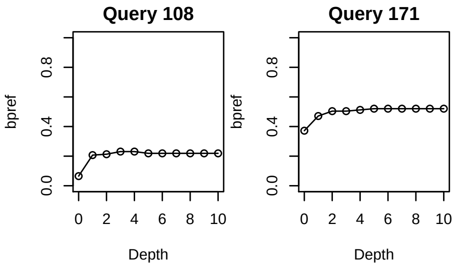


Fig. 7 Change in bpref compared to lvl0 for different settings of k. For each k, dark cells represent gains over lvl0, light cells represent losses.

Queries like 171 tended to suffer from the Conceptual Implication problem (Section 2). It was the deductive inference mechanism of the GIN that addressed the Conceptual Implication problem by traversing valuable SNOMED CT relationships; thus, the GIN was able to infer concepts that implied the query concepts and as a consequence promoting documents that contained these implied concepts.



108 Patients treated for vascular claudication surgically
171 Patients with thyrotoxicosis treated with beta blockers

Fig. 8 Queries with consistent improvements. Note that depth=0 denotes the performance of the concept baseline.

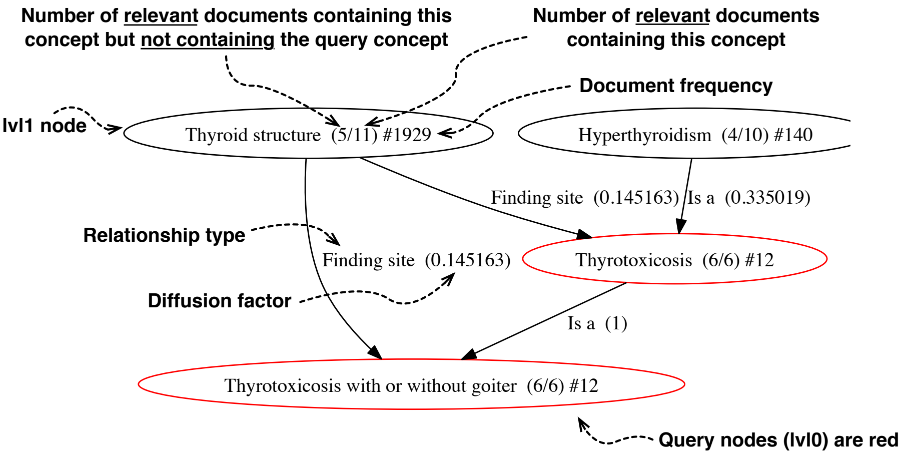
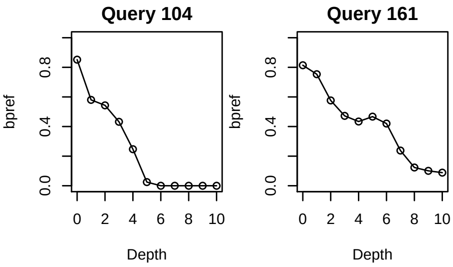

Number of relevant documents containing this concept but not containing the query concept
Number of relevant documents containing this concept
Document frequency
lvl1 node
Thyroid structure (5/11) #1929
Hyperthyroidism (4/10) #140
Finding site (0.145163)
Is a (0.335019)
Relationship type
Finding site (0.145163)
Thyrotoxicosis (6/6) #12
Diffusion factor
Is a (1)
Thyrotoxicosis with or without goiter (6/6) #12
Query nodes (lvl0) are red

Fig. 9 Partial traversal graph for query 171.

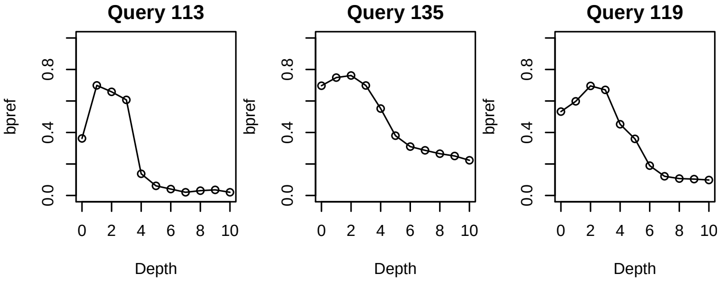
6.2 Consistent Degradation

A number of queries exhibited decreasing performance at greater depth levels. These were queries that did not require inference and tended to have a small number of relevant documents and an unambiguous query definition (Figure 10). For example, the "Robot" concept (query 104) and the "Adult respiratory distress syndrome" concept (query 161) provided all that was required to retrieve and rank relevant documents. Using the lvl0 query concepts, most relevant documents were ranked effectively. At greater levels, there were a large number of very general concepts that tended to degrade retrieval performance.



104: `Patients diagnosed with localized prostate cancer and treated with robotic surgery`
161: `Patients with adult respiratory distress syndrome`

Fig. 10 Queries that exhibited decreasing performance at greater depth levels.



113: `Adult patients who received colonoscopies during admission which revealed adenocarcinoma`
119: `Adult patients who presented to the emergency room with with anion gap acidosis secondary to insulin dependent diabetes`
135: `Cancer patients with liver metastasis treated in the hospital who underwent a procedure`

Fig. 11 Queries with effective reranking using the GIN.

6.3 Improving Precision due to Reranking

Queries that benefitted from reranking tended to suffer from the granularity mismatch problem (three example queries are provided in Figure 11). Granularity mismatch was addressed by the GIN's deductive inference mechanism, realised as traversals over *ISA* relationships. For example, query 135 (Figure 12) contained a very specific query concept (shown in red). A number of documents contained this specific query concept, however, these documents also contained a number of more general concepts related to the query concept via a ISA relationship. By traversing the ISA relationships to these more general concepts the attached documents were scored again for these related concepts, thus increasing their relevance score and effectively reranking them.

In addition, queries that benefit from reranking also tended to have two dependent aspects to the query; e.g., query 113 had a procedure ("colonoscopy") and diagnosis ("Adenocarcinoma") and query 119 had a symptom ("anion gap acidosis") and a disease ("insulin dependent diabetes").

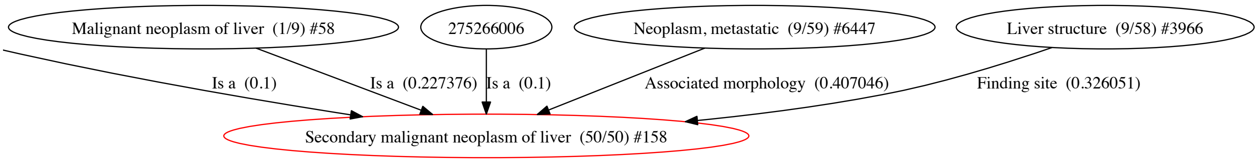
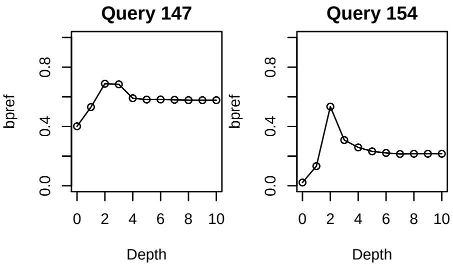

Malignant neoplasm of liver (1/9) #58
275266006
Neoplasm, metastatic (9/59) #6447
Liver structure (9/58) #3966
Is a (0.1)
Is a (0.227376)
Is a (0.1)
Associated morphology (0.407046)
Finding site (0.326051)
Secondary malignant neoplasm of liver (50/50) #158

Fig. 12 Partial traversal graph for query 135.



147: `Patients with left lower quadrant abdominal pain`

154: `Patients with Primary Open Angle Glaucoma`

Fig. 13 Queries where the GIN retrieved new relevant documents.

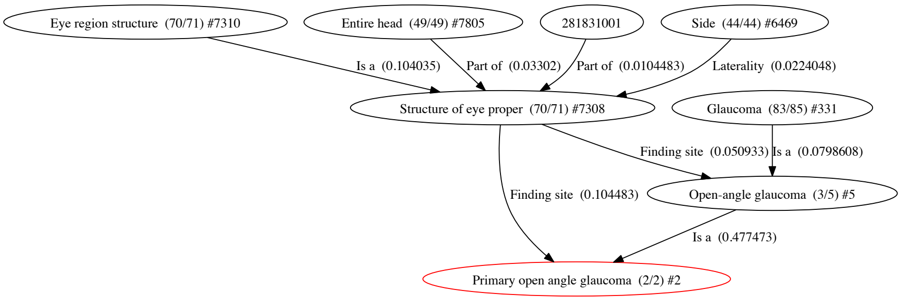
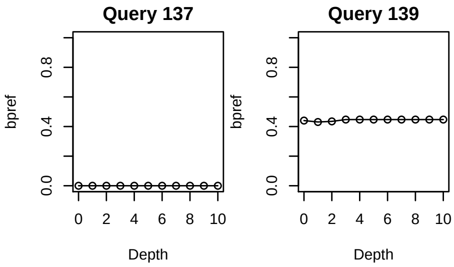


Fig. 14 Partial traversal graph for query 154.

6.4 Improving Recall due to Inference of New Relevant Documents

In contrast to improved precision due to reranking, the effectiveness of some queries improved by retrieving relevant documents *not* retrieved by the lvl0 baseline but provided by the inference mechanism (Figure 13). For example, for query 154 (Figure 14) only 2 relevant documents were found at lvl0 because the "Primary open angle glaucoma" query concept is too specific. At lvl1, the more general concept "Open angle glaucoma" is traversed, resulting in 3 additional relevant documents being retrieved. At lvl2, the "Glaucoma" concept is traversed, resulting in 83 additional relevant documents.

These queries exhibited both granularity and vocabulary mismatch. In this case, the GIN traversed concepts related to the query, identifying the valuable



```
137:  Patients with inflammatory disorders receiving TNF inhibitor treatments
139:  Patients who presented to the emergency room with an actual or suspected
      miscarriage
```

Fig. 15 Queries that exhibited constant performance for different depth settings.

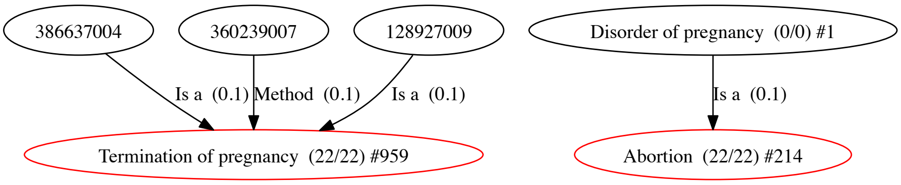


Fig. 16 Partial traversal graph for query 139.

information in SNOMED CT required to retrieve additional relevant documents not found using just the query concepts. The GIN was always more effective than the concept baseline, no matter the depth setting (although the best performance was found for depth settings 1–3). Granularity mismatch was addressed by deductive inference, realised as traversals over *ISA* relationships. Vocabulary mismatch was addressed through the use of the concept-based representation. Inferences of similarity was addressed by the diffusion factor, which controlled the uncertainty of the inference.

6.5 Unaffected Queries

Some queries exhibited a near constant performance for different depth settings (two examples are shown in Figure 15). Unaffected queries were those that: (i) were particularly challenging, such as query 137, which had very poor performance for term, concept, GIN and TREC systems (where the median bpref was 0.000 for all automated systems); (ii) had little or no information attached to the query concepts in SNOMED CT (Figure 16); thus, there were no documents attached to lvl1 nodes and the GIN was essentially behaving as the concept baseline model (lvl0).

Depth Approach	**All Queries**		**Hard (TREC Median)**	
	Bpref	P@10	Bpref	P@10
Fixed — lvl0	0.4290	0.5123	0.1985	0.2800
Fixed — lvl1	0.4229	0.4481	0.2024	0.2425
Fixed — lvl2	0.4138	0.4259	0.2072	0.2275
Adaptive Depth, 0–10 (Oracle)	0.4731 (+10%)†	0.5741 (+12%)†	0.2572 (+30%)†	0.3475 (+24%)†

Table 5 Graph Inference model retrieval results using the best depth setting per-query. This represents an oracle upper bound for an adaptive depth method. The percentages show the improvements of this method against the lvl0 baseline. † indicates statistical significant differences with fixed approaches (paired t-test, $p < 0.05$).

6.6 Selectively applying Inference

The analysis so far highlights that inference is required for some queries but not for others (or varying degrees are required). Practically, this equates to adaptively controlling the depth of traversal on a per-query basis. To understand the potential gains that this might provide, we selected the bpref value for the best depth setting for each query and averaged this across all queries; this represents an oracle upper bound for an adaptive depth method. The results are shown in Table 5, along with the fixed depth approaches for comparison.

As suspected, the adaptive method demonstrates the best performance. More important though is what characteristics or conditions might indicate the optimal depth setting. We have already commented that hard queries required inference and that the Graph Inference model was more effective for these. (Indeed, Table 5 shows that large gains were made for the adapative approach on hard queries.) In contrast, easy queries do not require inference. Therefore, a query performance predictor might inform whether it is worth traversing beyond level 0.

Inference can be risky. For hard queries, there is nothing to lose and adding domain knowledge can bring substantial benefits. For easy queries, adding domain knowledge is not required and can introduce noise. The analysis provided here points to an adaptive approach, where inference is applied on a per-query basis, as more appropriate. Future work can be directed toward the development such an an adaptive depth method.

6.7 Computational Complexity

The computational complexity of the GIN retrieval (Algorithm 2) is based on the number of documents scored each time a node is visited (score function on line 8). At each depth level $l = [0, .., k]$, there are e^l nodes, where e is the average number of edges (degree) for nodes in the graph G. Assuming an average of $\hat{d}$ documents are attached to each node, then $e^l\hat{d}$ documents are

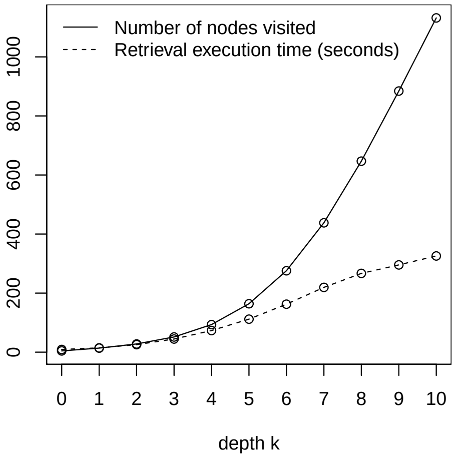


Fig. 17 The number of nodes traversed and the retrieval time in seconds for each depth level k, calculated across the full 81 queries. The number of nodes traversed increases exponentially with the depth k. However, the execution time degrades at a much slower rate.

processed at each depth level. When traversing multiple levels for a *single* query concept, the number of documents processed is $\sum_{l=0}^{k} e^l \hat{d}$. For a query of size $|Q|$ concepts, the number of documents processed is $|Q| \sum_{l=0}^{k} e^l \hat{d}$.

As stated previously, at a certain depth the diffusion factor becomes so small that documents scored at this level will not change the overall ranking; thus, we need consider only the documents k edges away from the query node.[12] The size of $\hat{d}$ is determined by the average inverse document frequency of the collection. The size of e (average number of edges per node) is the average degree of G (for SNOMED CT the average degree is 2.02). The size of the query, $|Q|$, is typically small for a retrieval scenario. With e, l and $|Q|$ all small, the retrieval method is computationally feasible.

The computational complexity analysis shows that the most influential factor is the depth level l (where $l = [0, .., k]$). We empirically investigate this by measuring the number of nodes traversed and the retrieval time in seconds for each depth level k; this is shown in Figure 17.[13] The number of nodes traversed increases exponentially with the depth k. However, the execution time did not increase exponentially, with the rate of increase tapering off for $k > 7$. This was because less documents were scored at these greater depth levels and because of caching that avoids recomputing statistics for nodes that have already been visited.

[12] The empirical evaluation revealed $k = [0 - 3]$ was preferred.

[13] Experiments were conducted on a Dell PowerEdge R710 Rack Mount Server with Dual Intel 3.33GHz processors, 96GB RAM and running Ubuntu 10.04 (64-bit).

7 Understanding when Inference Works

A number of issues arose from the underlying representation (SNOMED CT). The analysis of the SNOMED CT Relationship Types showed that the GIN traversed far more ISA relationships than any other (Figure 4). These relationships are valuable for overcoming granularity mismatch (Zuccon et al, 2012) but do not help address the aspects underlying the other semantic gap problems. For these, different types of relationships are required, such as *treatment* → *disease* and *organism* → *disease*. The former relationships are not modelled in SNOMED CT as they are not definitional.[14] For the latter, the coverage in SNOMED CT is lacking (Spackman, 2008). In addition, coverage may also vary considerably for ISA relationships: some concepts may inherit from very specific parent concepts (for example, "Right ventricle" $\xrightarrow{\text{ISA}}$ "Cardiac ventricle"), while others may inherit from very general parent concepts (for example, "Vertebral Unit" $\xrightarrow{\text{ISA}}$ "Body Structure"). This affects the GIN as some ISA relationships may provide valuable information, while others are too general for inference that promotes effective retrieval. Section 6.2 showed that performance degraded when very general concepts were traversed. To address this, work by Boudin et al (2012), which attempts to identify the granularity of concepts in a medical query, might be applied. More generally, poor performance in the GIN was found in queries where there was little valuable information at levels greater than 0 (Sections 6.2 & 6.5). These issues highlight that SNOMED CT as the underlying representation, rather than the traversal mechanism, is a limiting factor for the GIN.

The effect of the underlying representation raises the wider issue of using for retrieval an ontology originally designed for knowledge representation. The purpose of SNOMED CT (or many other such domain knowledge resources) is to represent the concepts belonging to that domain; the information regarding these concepts is *definitional* (Spackman, 2008). The conclusions possible using this definitional information are valid from a conceptual point of view; however, these conclusions may not be valuable from an IR perspective. For example, it is logically true that "Vertebral Unit" is indeed a "Body Structure" but this is unlikely to be of any value when encountering "Vertebral Unit" in a retrieval scenario. Two types of inference are at play here: *definitional inference*, used in knowledge representation to understand the concepts belonging to that domain, and *retrieval inference*, used to determine whether some evidence (e.g. found in a document) may entail relevance to a statement (e.g., a query). A consequence of the differing requirements between these two types of inference is that many relationships that are definitional are not useful for retrieval. The strain between definitional and retrieval inference has been highlighted as one of the challenges in utilising conceptual representations and alternative representations are currently under investigation (Frixione and Lieto, 2012).

In the GIN, inference is realised as a traversal over the graph. The depth parameter k controls how many edges are traversed from the query node and

[14] Opinions may differ on the best treatment for a disease and may change over time.

reflects how much additional information the model draws on (or how inference is applied) to score documents. Section 6 highlighted that the best performance was achieved for depth 1–3 (Figure 7). Beyond this, the related concepts were too peripheral to the query concepts and often introduced noise. (For some cases, this was mitigated by the diffusion factor, which decreases rapidly the further the concept is from the query concept.) The analysis also showed that different amounts of inference are required for different queries and that a static setting of the depth parameter k may not be optimal. An adaptive approach that determines the depth on a per-query basis would be more appropriate. This is left to future work.

8 Related Work

The theoretical inspiration for the GIN comes from previous work in logic-based IR (Van Rijsbergen, 1986), where relevance is modelled as $P(d \rightarrow q)$, i.e., the likelihood that a document implies the query. The Logical Uncertainty Principle (Van Rijsbergen, 1986) provides a means of evaluating $P(d \rightarrow q)$: if $d \rightarrow q$ cannot be immediately evaluated, e.g. not all query terms appear in d, then some other document d' is considered, such that $d' \rightarrow q$ is true. The measure of the uncertainty is determined by the *distance* between d and d'. Nie (1989) used a graph analogy to describe the distance measure as a sequence of steps from d to d'. The distance measure is akin to the diffusion factor in the GIN: the combination of a sequence of transitions from document nodes to query nodes. A key difference in the GIN is that the diffusion factor is determined between Information Units rather than documents and that the diffusion factor is informed by both a similarity (i.e., distance) and domain knowledge. The GIN also bears a resemblance to the Logical Imaging technique for IR (Crestani and van Rijsbergen, 1995), where the truth of the logical implication, $P(d \rightarrow q)$, is evaluated as a function of the expected mutual information between terms. Similar to the GIN, at retrieval time, Logical Imaging scores a document by producing a probability kinematics that moves probability mass from terms that are not in the document to (query) terms that are in the document. Unlike the GIN, in Logical Imaging the probability kinematics is driven solely by statistical similarity (expected mutual information) and there are no multiple levels of transfers (i.e., levels of inference in the GIN).

The GIN makes use of structured knowledge resources. Early work by Voorhees (1994) used WordNet for query expansion, while Ravindran and Gauch (2004) developed a conceptual search engine based on a manually constructed concept hierarchy and Egozi et al (2011) developed the ESA model, which used Wikipedia articles as concepts. Empirically, these general concept-based approaches struggled to outperform keyword-based systems; however, biomedical applications — which use domain specific ontologies — do demonstrate consistent improvements (Zhou et al, 2007; Liu and Chu, 2007; Koopman et al, 2012b; Limsopatham et al, 2013a). Contrary to the GIN, most of these approaches only use concepts for augmenting the query (often in query expansion (Liu and Chu, 2007)); those that do use concepts for document representation, do not take advantage of relationships between concepts (Koopman et al, 2012b; Limsopatham et al, 2013a).

The GIN shares the same intuition as Turtle and Croft (1991) in that effective IR systems have to "infer probable relationships between documents and queries". The proposal of Turtle and Croft (1991) also realises inference through a graph traversal. However, the GIN differs in that it uses a unified graph representation for documents and queries, whilst Turtle and Croft (1991) construct separate graphs for each.

Within the area of genomic information retrieval there has been a number of lines of relevant research. A number of specific query expansion methods are proposed (Stokes et al, 2008; Dinh and Tamine, 2011), some that exploit concept-based representations (Trieschnigg, 2010); however, few exploit the relationships between concepts to drive any inference mechanism. An exception is work by Zhou et al (2007) that infers concepts "implicitly related" to the query, via domain knowledge, and thus incorporates inference into a retrieval method. (This method proves effective on the TREC Genomics test collection.) However, the genomic domain in which all these methods are applied is a very specific and constrained IR scenario: all queries adhere to a <biological object, relationship, biological process> template, where an object could be a gene and a process could be a disease. This clearly identifies the type of inferences that are required (e.g., the relationships between genes and diseases). Therefore, many of the methods proposed, including the inference mechanism of Zhou et al (2007), cannot be applied outside of genomics domain to medical IR in general.

State-of-the-art approaches used in TREC MedTrack feature well explored statistical IR models (e.g. Mixture of Relevance Models (Zhu and Carterette, 2012), Divergence from Randomness and voting models (Limsopatham et al, 2013a,b)) along with thorough engineering tailored to the MedTrack task (e.g., age and gender processing, document type, etc.). While these approaches do provide strong empirical results compared to the GIN, a key difference is that the GIN retrieves relevant documents that are not findable with these models; this was shown in Experiment 2 where many documents retrieved by the GIN but not included in the MedTrack pool turned out to be relevant.

The GIN is proposed to address the four semantic gap problems: vocabulary mismatch, granularity mismatch, conceptual implication and inferences of similarity; these problem were outlined in Section 2. There is empirical evidence showing that IR systems are hampered by these problems. Edinger et al (2012) conducted a failure analysis of the teams participating in TREC Medical Records tracks. Their categorisation of IR system failures revealed the same issues around vocabulary, granularity and contextually already highlighted in this paper. In addition, they show many cases where the query terms "must be inferred", highlighting the requirement for an inference mechanism advocated in our paper.

9 Conclusion

Our implementation of the GIN addressed the four semantic gap problems. Regarding vocabulary mismatch, the GIN utilised the same concept-based representation as the concept baseline and thus inherited its benefits for overcoming vocabulary mismatch (Koopman et al, 2012b; Limsopatham et al, 2013a). The GIN addressed granularity mismatch by traversing parent-child (ISA) relationships. The semantic gap problem of Conceptual Implication is where the presence of certain terms in the document infer the query terms. Where these associations were encoded in SNOMED CT, the GIN addressed Conceptual Implication by traversing these types of relationships. Finally, the problem of Inference of Similarity, where the strength of association between two entities is critical, was addressed by the diffusion factor, which assigned a corpus-based measure of similarity to the domain knowledge-based relationship. The empirical results have shown that the inference mechanism promoted recall by retrieving new relevant documents not found by previous keyword-based approaches. In addition, it promoted precision by an effective reranking of documents. When inference is used, performance gains can generally be expected on hard queries. However, inference should not be applied universally: for easy, unambiguous queries and queries with few relevant documents, inference did adversely affect effectiveness. These conclusions reflect the fact that for retrieval as inference to be effective, a careful balancing act is involved. The need for this balancing fundamentally derives from the observation that inferences that may be valid at the level of concepts may not lead to the inference of relevant documents. For this reason, future research should be directed at query analysis which can reliably predict which queries are amenable to retrieval as inference.

Although developed and applied within medical search, the GIN is a general retrieval model. For example, the GIN can be applied to web search using Freebase as the structured domain knowledge resource; the GIN can be tested in this domain using the Freebase annotated version of the entire ClueWeb12 collection made available by Google[15]. These annotated resources are the mapping of the free-text web documents and queries to structured Freebase entities, and represent the companion of what MetaMap provided for the experiments of this paper. Compared to SNOMED CT, Freebase also provides a different type of underlying representation, one that is less definitional and more associational. Therefore, applying the GIN to web search also evaluates the model using a potentially more suited knowledge resource (Freebase). Preliminary work on exploiting Freebase annotations for web search have shown promise (Dalton et al, 2014). Applying the GIN to web search is, therefore, a natural avenue of future work.

Conflict of Interest

~~The authors declare that~~ they have no conflict of interest.

[15] ClueWeb12 Freebase Annotations: http://lemurproject.org/clueweb12/FACC1.

References

Aronson A, Lang FM (2010) An overview of MetaMap: historical perspective and recent advances. JAMIA 17(3):229–236

Bendersky M, Croft B (2008) Discovering key concepts in verbose queries. In: SIGIR, pp 491–498

Boudin F, Nie JY, Dawes M (2012) Using a medical thesaurus to predict query difficulty. In: ECIR, pp 480–484

Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG (2001) Evaluation of negation phrases in narrative clinical reports. In: Proceedings of the AMIA Symposium, American Medical Informatics Association, p 105

Crestani F, van Rijsbergen CJ (1995) Information retrieval by logical imaging. Journal of Documentation 51(1):3–17

Dalton J, Dietz L, Allan J (2014) Entity query feature expansion using knowledge base links. In: Proceedings of SIGIR, Gold Coast, Queensland, Australia, pp 365–374

Dinh D, Tamine L (2011) Combining global and local semantic contexts for improving biomedical information retrieval. In: Advances in Information Retrieval, Springer, pp 375–386

Edinger T, Cohen AM, Bedrick S, Ambert K, Hersh W (2012) Barriers to retrieving patient information from electronic health record data: failure analysis from the TREC Medical Records Track. In: AMIA, Washinton D.C., USA, vol 2012, pp 180–188

Egozi O, Markovitch S, Gabrilovich E (2011) Concept-Based Information Retrieval using Explicit Semantic Analysis. ACM Transactions on Information Systems 29(2):1–38

Ely J, Osheroff J, Gorman P, Ebell M, Chambliss M, Pifer E, Stavri P (2000) A taxonomy of generic clinical questions: classification study. British Medical Journal 321(7258):429–432

Frixione M, Lieto A (2012) Representing concepts in formal ontologies: Compositionality vs. typicality effects. Logic and Logical Philosophy 21(4):391–414

Koopman B, Zuccon G (2014a) Document timespan normalisation and understanding temporality for clinical records search. In: Proceedings of the 19th Australasian Document Computing Symposium, Melbourne, Australia

Koopman B, Zuccon G (2014b) Understanding negation and family history to improve clinical information retrieval. In: Proceedings of the 37th annual international ACM SIGIR conference on research and development in information retrieval, ACM

Koopman B, Zuccon G (2014c) Why assessing relevance in medical IR is demanding. In: Proceedings of the SIGIR Workshop on Medical Information Retrieval (MedIR), Gold Coast, Australia

Koopman B, Bruza P, Sitbon L, Lawley M (2010) Analysis of the effect of negation on information retrieval of medical data. In: Proceedings of the Fifteenth Australasian Document Computing Symposium (ADCS), Melbourne, Australia, pp 89–92

Koopman B, Zuccon G, Bruza P, Sitbon L, Lawley M (2012a) An Evaluation of Corpus-driven Measures of Medical Concept Similarity for Information Retrieval. In: CIKM, pp 2439–2442

Koopman B, Zuccon G, Nguyen A, Vickers D, Butt L, Bruza P (2012b) Exploiting SNOMED CT Concepts & Relationships for Clinical Information Retrieval: AEHRC and QUT at the TREC Medical Track. In: TREC

Lancaster FW (1986) Vocabulary Control for Information Retrieval., 2nd edn. Information Resources Press, Washington, D.C

Limsopatham N, Macdonald C, McCreadie R, Ounis I (2012) Exploiting Term Dependence while Handling Negation in Medical Search. In: Proceedings of the 35th International Conference on Research and Development in Information Retrieval (SIGIR), ACM, Oregon, USA, pp 1065–1066

Limsopatham N, Macdonald C, Ounis I (2013a) A Task-Specific Query and Document Representation for Medical Records Search. In: ECIR, pp 747–751

Limsopatham N, Macdonald C, Ounis I (2013b) Aggregating Evidence from Hospital Departments to Improve Medical Records Search. In: ECIR, pp 279–291

Limsopatham N, Macdonald C, Ounis I (2013c) Inferring conceptual relationships to improve medical records search. In: OAIR, pp 1–8

Liu Z, Chu WW (2007) Knowledge-based query expansion to support scenario-specific retrieval of medical free text. Information Retrieval 10(2):173–202

Nadkarni PM, Ohno-Machado L, Chapman WW (2011) Natural language processing: an introduction. JAMIA 18(5):544–551

Nie J (1989) An information retrieval model based on modal logic. IP&M 25(5):477–491

Pratt W, Yetisgen-Yildiz M (2003) A study of biomedical concept identification: MetaMap vs. people. In: AMIA, pp 529–533

Ravindran D, Gauch S (2004) Exploiting hierarchical relationships in conceptual search. In: CIKM, pp 238–239

Sowa JF, et al (2000) Knowledge representation: logical, philosophical, and computational foundations, vol 13. MIT Press

Spackman K (2008) SNOMED Clinical Terms Basics, international Health Terminology Standards Development Organisation Technical Report

Stokes N, Li Y, Cavedon L, Zobel J (2008) Exploring criteria for successful query expansion in the genomic domain. Information Retrieval 12(1):17–50

Trieschnigg D (2010) Proof of concept: concept-based biomedical information retrieval. PhD thesis, University of Twente

Turtle H, Croft WB (1991) Evaluation of an inference network-based retrieval model. ACM Transactions on Information Systems 9(3):187–222

Van Rijsbergen CJ (1986) A non-classical logic for information retrieval. Computer Journal 29(6):481–485

Voorhees EM (1994) Query expansion using lexical-semantic relations. In: SIGIR, pp 61–69

Voorhees EM, Hersh W (2012) Overview of the TREC 2012 Medical Records Track. In: TREC

Voorhees EM, Tong RM (2011) Overview of the TREC 2011 Medical Records Track. In: TREC

Zhou W, Yu C, Smalheiser N, Torvik V, Hong J (2007) Knowledge-intensive conceptual retrieval and passage extraction of biomedical literature. In: SIGIR, pp 655–662

Zhu D, Carterette B (2012) Combining multi-level evidence for medical record retrieval. In: Workshop on Smart Health and Wellbeing, pp 49–56

Zuccon G, Koopman B, Nguyen A, Vickers D, Butt L (2012) Exploiting Medical Hierarchies for Concept-based Information Retrieval. In: ADCS, pp 111–114