Semantic Web 0 (0) 1
IOS Press

SINFIO – A User-Friendly Hybrid Semantic Search Engine

Kinga Schumacher [a,*], Harald Sack [b]
[a] *German Research Center for Artificial Intelligence, Germany*
E-mail: kinga.schumacher@dfki.de
[b] *FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany*
E-mail: harald.sack@fiz-karlsruhe.de

Abstract. Hybrid semantic search aims to close the gap between fact retrieval and semantic document retrieval. The combination of facts and documents throughout the entire search process is required in order to exploit available information in various representation forms to full extent. This leads to numerous challenges such as combining fact and document retrieval, merging facts and documents into hybrid results, ranking of differently structured results, etc. Moreover, to achieve user acceptance, the complexity of the system must be hidden from the user. Especially in the case of novel systems, an understandable presentation plays a key role.

SINFIO, the hybrid semantic search engine described in this paper, offers a solution to the challenges mentioned above. Besides proving the gain on effectivity over a hybrid semantic search solution which does not combine facts and documents throughout the entire search process, this paper also presents the results of evaluations with respect to ranking and the user interface. The user studies show a clear acceptance of SINFIO. Despite novel hybrid results, users prefer SINFIO over the solution that does not combine facts and documents across the entire search process, over standalone fact retrieval as well as standalone document retrieval.

Keywords: Hybrid Semantic Search, Search Result Presentation, Hybrid Semantic Search Evaluation

1. Introduction

Today's information management systems like intranets, Web 3.0 applications, as well as many web portals, contain information in heterogeneous formats and structures. They provide structured data as well as documents that are related to this data. These documents are usually only partially structured or completely unstructured. For example, travel portals describe the period, destination, and cost of travel using structured data, while additional information, such as descriptions of the hotel, destination, excursions, *etc.* is provided in unstructured form.

The focus of today's semantic search engines is to find information either in a structured form, in formal knowledge bases (*fact retrieval*), or in semi- or unstructured form, usually mapped to a document index (*semantic document retrieval*)[1] [2]. Only a few semantic search engines try to close the gap between these two approaches and answer queries with facts as well as with documents. Although they search simultaneously for structured and unstructured data, the results are either analyzed independently, or the search possibilities are highly limited (see section 2). Accordingly, the information available in the system is not exploited, and, simultaneously, the relationships between individual pieces of content in the respective information systems and complementary information cannot reach the user.

*Corresponding author. E-mail: kinga.schumacher@dfki.de.

[1]The degree of formality is determined by the strength of formalization trough formal representation languages of the structure. Facts represented in RDF/S or OWL are structured and formal. If no formal representation languages are used, such as for free text, the contents are unstructured and informal [1].

In order to close this gap, the hybrid semantic search approach SINFIO combines structured and semi- or unstructured content throughout the entire search process. This approach not only finds facts and documents, it also uses relationships that exist between the different items of data at every stage of the search process, and integrates them into the search results. If the answer to a query is not completely structured (fact), or unstructured (document), this approach provides a query-specific combination of both (hybrid result). Moreover, SINFIO supports structured, unstructured and hybrid queries while the user formulates the information need using natural language queries. However, consideration of structured as well as semi- or unstructured content by the information system throughout the entire search process poses a special challenge to the search engine. With respect to the backend, this engine must be able

- to browse facts and documents,
- to combine them during the search process
- to rank the different results (fact, document and hybrid) in an appropriate order.

Furthermore, the complexity of the data should not be apparent to the end user. Rather, the presentation of the contents must be understandable and easy to interpret, both in the query request and the presentation of results. In terms of the user interface, the main challenges are:

- to support the user in query formulation (structured, unstructured and hybrid),
- to represent the results itself (facts, document and hybrid) understandable and
- to represent the result list that contains facts, document and hybrid results, in a way that it is easy to interpret to the user.

The most important **questions** of this work are:

Is a hybrid semantic search approach, which combines structured and semi- or unstructured content throughout the entire search process more effective than hybrid approaches that do not combine them?

Can the user interface be designed in such a way that the complexity of the search process remains hidden to the users while the presentation of the (novel) content is understood?

This paper is structured as follows: Chapter 2 introduces related works on hybrid semantic search. Chapter 3 describes the approach SINFIO including the architecture (Section 3.1), the search process (Section 3.2), the ranking method (Section 3.3) and the result presentation (Section 3.4). Chapter 4 presents the evaluation results with respect to effectivity (Section 4.1), ranking (Section 4.2) and the user interface (Section 4.3) as well as a correlation analysis among the results (Section 4.4). We conclude this work in Chapter 5.

2. Related Work

Commercial web search engines, like Google or Bing, are already adressing the issue of structured, semi-structured and unstructured data. They provide additional structured information in a box at the right side of the screen for queries that ask for one specific concept like a person or city. Google also performs question answering based on structured data providing the answer itself as one single factoid result at the top of the result list. Thus, the facts and documents are presented independent from each other and facts are only provided for concept-queries or questions that Google can answer with a fact. These solutions approves the demand on hybrid search but doesn't combine the available data throughout the entire search process.

Considering semantic search enignes, early (almost) hybrid approaches combine keyword-based search in the document index and retrieval in formal knowledge bases in the backend, but they don't support both document as well as fact retrieval simultaneously. TAP, KIM, and the web search engine described in [3] accomplish fact retrieval in addition to a search in the document index. However, both tasks are performed independently. TAP focuses on entity search and presents, additionally to the document list, the found entity and its semantic properties (with predefined path length) in a box at the right side of the result page [4]. KIM enables users to search the knowledge base as an alternative to searching the document index, but it uses the matched entities as a filter on the document set, *i. e.*, the results are documents that are annotated with query matching entities [5]. The search engine introduced in [3] maps web sites to instances of an ontology and executes keyword-based semantic document retrieval. Thus, the search results are instances that represent web pages, *i. e.*, documents. HyKSS and Mímir search the knowledge base as well as the document index. However, HyKSS applies found facts only to adjust the order of the result documents [6] while Mímir uses matching entities and facts to filter the document set [7]. HS^3 creates a so called index graph in order to interlink the knowledge base and document index based on the semantic annotations of the documents. The index graph is used to identify documents that don't contain all query terms but are connected to at least one of the entities matched by the query [8]. All of these approaches belong to the category of semantic document retrieval: they use formal semantics in order to enhance document retrieval.

To clearly distinguish it from semantic document retrieval, we define **hybrid semantic search** *as a process that combines fact retrieval with simple or semantic document retrieval in a way that both are executed interdependently while the search engine is able to find both facts as well as documents*.

In terms of this definition, PowerAqua, K-Search and CE^2 are hybrid semantic search engines. PowerAqua [9] extends the question answering machine AquaLog [10] with a component for document retrieval, where the document index also includes the semantic metadata of the documents. PowerAqua performs fact retrieval and, in addition, queries the document index by the names of the found entities. In the result list, facts take precedence over the documents, *i. e.*, documents are considered as provider of additional information. If no facts are found, the search engine performs document retrieval. Thus, search queries can be answered even if there are no relevant facts available [11–13].

K-Search doesn't combine fact and document retrieval in terms of including the partial results of one retrieval method in the other one. It searches the knowledge base and document index with the same query and reconciles the matched facts and documents. If a document is annotated with all entities that occur in a fact, the document and the fact are considered to be related. Facts without a related document are deleted from the result set. For result presentation users can choose between the list of documents and the list of facts. K-Search supports three kinds of queries: formal, *i. e.*, instances and properties from the knowledge base (*e. g.*, $\langle Person \rangle$); informal, *i. e.*, natural language keywords (*e. g.*, "Michael"); and hybrid, *i. e.*, queries composed of formal and informal parts (*e. g.*, $\langle Person \rangle$ with the $\langle surname \rangle$ "Michael"). The formal parts of hybrid queries are used as filter, they determine the relevant subset of concepts in the formal knowledge base. The informal parts, *i. e.*, the keywords in natural language, are used to search in this subset, where only the literals are considered. Thus, K-Search supports only a conjunction of the formal and informal parts of the query [14].

The search engine CE^2 instantiates the documents and explores the graph representation of the knowledge base including the document instances. Documents are linked to the other instances of the knowledge base based on their semantic annotations. For an efficient keyword search, this graph is converted into an KB-, a document- and an annotation- text index. In order to perform hybrid search, CE^2 transforms the user query into a SPARQL-query which maps to a graph in a tree structure. The keyword search in the document index serves to identify instances of the knowledge base, which form parts of the tree structure. The search results are subgraphs of the knowledge base that satisfy the query graph [15]. Thus, CE^2 is limited to conjunctive queries that map a contiguous graph in tree structure. Connections over more than one unknown edges between two nodes can not be found.

Since this work focuses on the combination of facts and documents throughout the entire search process, the supported type of search queries and results as well as the way how to combine fact and document retrieval are of particular interest. Table 1 summarizes these characteristics of the introduced hybrid search engines.

3. SINFIO

3.1. Architecture

Figure 1 shows the architecture of SINFIO, sketching the two major phases of the underlying process: offline and online. According to our focus on the search process itself, *i. e.*, the online part of the architecture, the offline process is only described briefly.

The offline part deals with the integration of the information that is stored in different data sources and in heterogeneous formats and structures. It connects the data sources by processing the contents to convert them into the knowledge base (ontologies and instances) as well as the text index. Depending on the structure and formality of the data, *i. e.*, structured, semi-structured or unstructured, different processing steps like annotation, mapping, and natural language processing (NLP) are required to fill the knowledge base. Statistical analysis is typically applied to create the text index. The information pool, composed of the knowledge base and the text index, integrates and enables access to information from various data sources, potentially covering the whole spectrum from simple text documents over linked data to formally described

Table 1
Supported query and result types of the hybrid semantic search approaches

Approach	Query	Fact and Document Retrieval	Result
PowerAqua	informal	applies found entities for document search	facts, in addition documents
K-Search	formal, informal and hybrid (restricted)	independent	documents or facts
CE^2	formal, informal and hybrid (restricted)	includes document instances in fact retrieval	graphs of facts (incl. doc. instances)
SINFIO	*formal, informal and hybrid*	*interdependent*	*facts, documents and hybrid results*

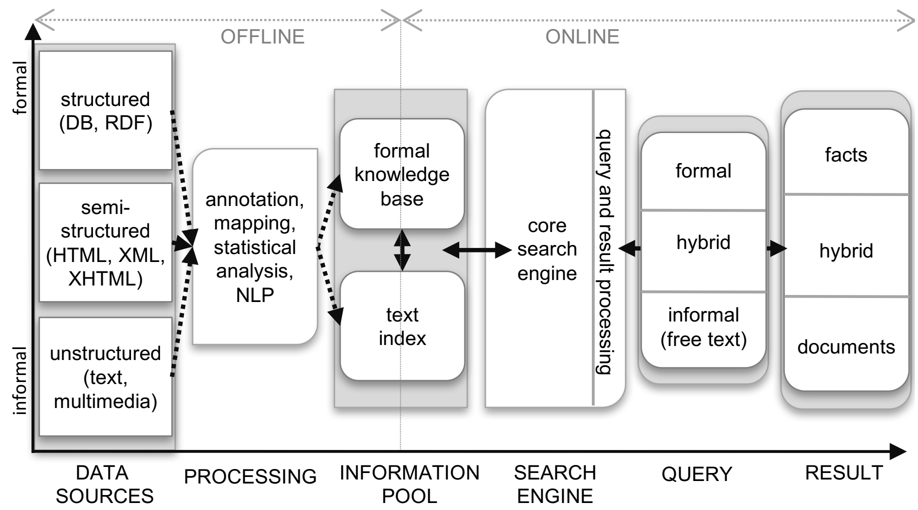

Fig. 1. Architecture of the hybrid search engine SINFIO

knowledge. Documents and facts are interlinked based on the semantic annotations of the documents [16].

The online part of the architecture depicts the search process including queries and results. As indicated in table 1, SINFIO performs fact and semantic document retrieval interdependently, it supports formal, informal and hybrid queries, and answers the queries with facts, documents, and hybrid results. The search process and the user interface are described in the following sections. The user interface has been developed based on the user-centered design [17]: design, testing and customization of the user interface took place in multiple iterations.

3.2. The search process

Given that methods to search a knowledge base are different from methods for searching in a text index, SINFIO combines a fact retrieval approach with semantic document retrieval. Both approaches have to work together in a way that the engine is able to exploit and combine both facts and documents meaningfully. Further requirements are the ability to resolve ambiguities and to apply an adequate ranking function which is suitable for mixed result lists, where a single result can be one or more facts[2], a document as well as a hybrid result. Based on an analysis of fact and semantic document retrieval approaches with respect to the requirements, a triple-based fact retrieval and a graph-traversal-based semantic document retrieval algorithm have been chosen for the following reasons: triple-based search provides the resolution of ambiguities, and the graph traversal algorithm Spreading Activation enables an effective combination of fact retrieval and document retrieval [16].

Figure 2 illustrates the hybrid semantic search process. After preprocessing the query, fact retrieval is carried out and results in a set of matched resources respectively facts. Labels and synonyms of the matched resources are applied for query expansion before querying the text index. The set of ranked resources and documents constitute the starting point for the hybrid search algorithm.

[2]one triple or a set of coherent triples

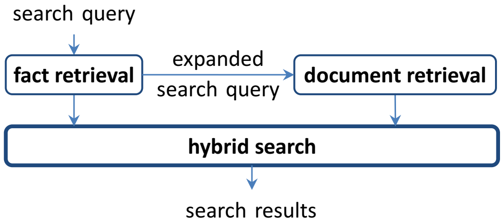

Fig. 2. Overview of the search process

If the fact retrieval is successful, a hybrid semantic search is performed. If no facts are found, the search engine can only perform a semantic document retrieval.

3.2.1. Query preprocessing

On the one hand, search engine users have a clear preference for natural language input in form of keywords and questions [18, 19]. On the other hand, formal structured parts are precise, unlike free text, they mitigate the problem of resolving syntactic and structural ambiguities[3]. Therefore SINFIO is designed to handle structured, unstructured and hybrid queries by natural language input from the user. To close the gap between system and user, a *semantic autocompletion* component aims to support the user to create queries with as many formal parts as possible. Semantic autocompletion proposes suitable knowledge-base concepts, but the label is not necessarily similar to the user's input, unlike traditional autocomplete, at the lexical level [20, 21]. Synonyms, abbreviations, homonyms, variants in spelling, singular/plural, and phrases must be recognized in order to identify concepts. However, if the user enters a term that can not be found in the suggestion list, the cognitive load increases because the information must be merged and the association established (cf. [22]). For this reason, SINFIO displays the synonym which the user is currently typing and the suggestion for it is highlighted in bold as in popular search engines. Attached with a dash, the most specific class of a concept is displayed. Property names are italicized, matched classes are written in capital letters. This visual distinction is applied in the results representation as well. Visual support aims to establish a link between the proposals and their role in the results in order to facilitate a selection from the proposal list. Figure 3 shows examples of semantic autocompletion with SINFIO.

The semantic autocompletion is realised based on an autocompletion index and prefix-search where a

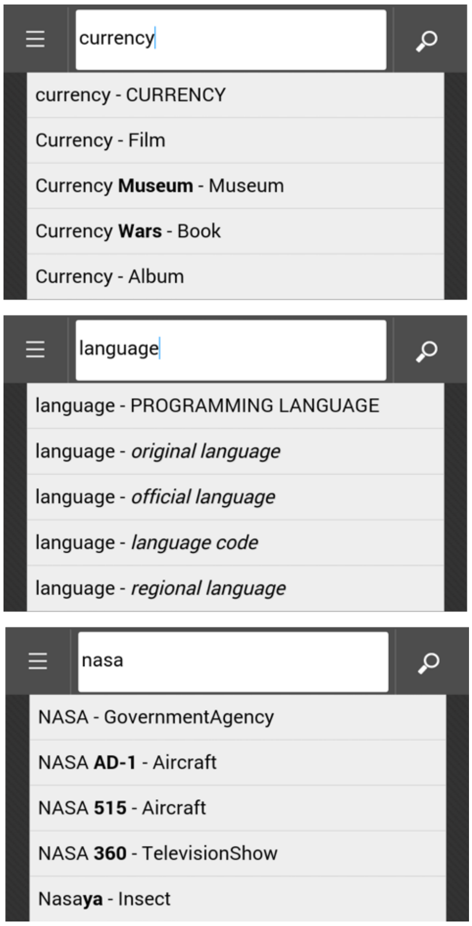

Fig. 3. Semantic autocompletion in SINFIO

component for the recognition of plural/singular forms is integrated. The formal knowledge base is extended with synonyms, abbreviations and homonyms based on the online thesaurus WordNet[4].

3.2.2. Fact retrieval and document retrieval

The triple-based fact retrieval approach maps query terms to literals in the knowledge base resulting in URIs of matching properties (p_i) and non-properties (n_j). These resources are used as subject, predicate or object to generate RDF queries, *e. g.*, $\langle n_1, p_1, ?\rangle$, $\langle ?, p_1, n_2\rangle$. In [23], the process of creating and applying queries is repeated for each new resource detected but it is limited to only two hops in the knowledge base in order to avoid irrelevant inferences. We adopted and extended this basic idea in order to enhance capabil-

[3] Ambiguity of the underlying structure of complex expressions

[4] https://wordnet.princeton.edu

ities by providing more hops in the knowledge base guided by the query.

For the formal description of SINFIO's fact retrieval approach, G_Σ denotes our knowledge base as graph including the ontology and instances, q the query as an ordered list of terms $q = (t_1, \ldots, t_n), n \in \mathbb{N}$. To explain the syntactic matching, we define the textual content L_{G_Σ} of G_Σ as

$$L_{G_\Sigma} = \{l |\ \exists \langle r, p, l \rangle \in G_\Sigma,\ l \text{ is a Literal}\}. \tag{1}$$

Furthermore, we define the set of resources without literals and statements[5] R_{G_Σ} and divide it in the set of properties P_{G_Σ} and the set of non-properties N_{G_Σ}:

$$R_{G_\Sigma} = \{r |\ \exists \langle r, p, o \rangle \in G_\Sigma,\ r \notin L_{G_\Sigma}\} \tag{2}$$

$$P_{G_\Sigma} = \{p | \\ \exists \langle p, \texttt{rdf:type}, \texttt{rdf:Property} \rangle \in G_\Sigma\} \tag{3}$$

$$N_{G_\Sigma} = R_{G_\Sigma} \backslash P_{G_\Sigma} \tag{4}$$

where $\langle s, p, o \rangle$ is an RDF triple that consists of subject s, predicate p and object o.

As described in section 3.2.1, our knowledge base supports the search engine with synonyms of the elements (classes, properties, and instances) which are used for the syntactic matching step. If the user does not choose one of the recommendations of the autocompletion component we apply the n-gram matching method to match a query term against the knowledge base [24, 25]. This method has been chosen for it's performance and a combination of 2-grams for terms up to 5 letters while 3-grams for all longer terms has been applied [24, 26–28]. If the user has chosen a concept from the semantic autocompletion component, the similarity value is 1.0. Otherwise the Dice-similarity is computed from the n-gram values. Thus, the syntactic matching returns for one query term $t_i \in q$ a set of 3-tuples $(t_i, r, w_{t_i r})$ including the query term, the matched resource and the dice-measure of the term and the resource's label or synonym as similarity-weight:

$$M(t_i, G_\Sigma) = \{(t_i, r, w_{t_i r}) |$$

$$t_i \in q,\ r \in R_{G_\Sigma},\ \exists \langle r, p, l_j \rangle \in G_\Sigma,\ l_j \in L_{G_\Sigma},$$

$$w_{t_i r} = \ dice - sim(ngram(t_i), (ngram(l_j)), \\ w_{t_i r} > H, \in [0,1] \subset \mathbb{R}\}. \tag{5}$$

Only resources r were included in the result set which are matched with a $w_{t_i r} > H$ where $H \in [0,1] \subset \mathbb{R}$ is a predefined similarity threshold. The recognition of phrases requires a combination of the entered terms in different orders. For example, the query "semantic web applications" can also cover the concepts "semantic", "web", "application", "semantic web" and "web application". This process is driven by the n-gram-method which computes the rank of the single concepts and also the possible phrases (processed as one term). If the weight of the phrase is higher than the average weight of its terms then we assume that the phrase is intended.

The total result of the matching process on the lexical level between the query and the knowledge base $M(q, G_\Sigma)$ is the union of the results per query term:

$$M(q, G_\Sigma) = \{\bigcup_{i=1}^{n} M(t_i, G_\Sigma)\}. \tag{6}$$

The matching on the semantic level use the structure of the formal knowledge base. To describe this process, we designate the graph of the instance base, *i. e.*, the instances with G_I. Analogously to R_{G_Σ}, P_{G_Σ} and N_{G_Σ}, we define the set of resources (again without literals and statements) R_{G_I}, properties P_{G_I} and non-properties N_{G_I} in G_I. In addition, we designate the set of literals with L_{G_I}. When the query is composed of only one term the semantic matching step returns resources and triples of the graph G_I, dependent on what kind of resources the term t_1 has matched. When the query is composed of more than one term, we iterate over every two adjacent terms t_i, t_{i+1} and consider their results from the syntactic matching $(M(t_i, G_I), M(t_{i+1}, G_I))$ as specified in figure 4. We create and apply possible SPARQL queries with the matched resources in order to find suited triples in G_I. We create query templates with these terms and the subjects/objects of the triples which were found based on the adjacent query terms as specified in figure 5.

Figure 7 exemplifies the process based on the query "In wich films directed by Garry Marshall was Julia Roberts starring?".

[5] RDFS defines rdfs:Literal and rdf:Statement as Resources.

function $semMatchFirst(q, G_I)$
$result = \emptyset$
$matched = \emptyset$
if $|q| > 1, q = t_1, \ldots, t_n$
$/ *$ for each pair of adjacent query terms $* /$
for each $(t_i, t_{i+1}) \in q,\ 1 \leqslant i \leqslant n-1$
    $result(t_i, t_{i+1}) = \emptyset$
    for each $r_j : (t_i, r_j, w_{t_i r_j}) \in M(t_i, G_\Sigma)$
        for each $r_k : (t_{i+1}, r_k, w_{t_{i+1} r_k}) \in M(t_{i+1}, G_\Sigma)$
            $/ *$ find suited triples in G_I $* /$
            $result(t_i, t_{i+1}) =$
                $result(t_i, t_{i+1}) \cup findStatements(r_j, r_k)$
    if $result(t_i, t_{i+1}) \neq \emptyset$
        $matched = matched \cup \{t_i, t_{i+1}\}$
        $result = result \cup result(t_i, t_{i+1})$
return $(result, matched)$

Fig. 4. Semantic matching

function $semMatchSecond(q, G_I, result, matched)$
$/ *$ iterate until all query terms are matched or no new triples found $* /$
$new = true$
while $new \wedge (|matched| < |q|)$
  $new = false$
    for each $t_i \in q,\ t_i \notin matched$
        $/ *$ for each adjacent term pair in q
        where one of them is not yet matched $* /$
        for each $t_j,\ j = i-1 \vee j = i+1$
            $part\_result = \emptyset$
            for each $r_i \in \{r_i | \exists (t_i, r_i, w_{t_i r_i}) \in M(t_i, G_\Sigma)\}$
                for each $r_j \in \{r_j | \exists (t_j, r_j, w_{t_j r_j}) \in M(t_j, G_\Sigma)\}$
                    $part\_result = findStatements(r_j, r_i)$
                    if $part\_result \neq \emptyset$
                        $result = result \cup part\_result$
                        $matched = matched \cup t_i$
                        $new = true$
return $result$

Fig. 5. Semantic matching, iterations on the results of figure 4

The ordered processing of unmatched terms enables to handle also enumerations of instances, properties or classes. The process is iterated and stops when either all query terms are matched or it is not possible to include all terms since some terms do not match existing triples in G_I. We also make as many hops in the knowledge base as possible guided by the query. The result consists of a set of instances and a set of triples. At least, we identify triple sets in the result which build a coherent subgraph of G_I. To avoid the merging of instances to a big subgraph in order to deliver well arranged results, each triple of a subgraph is connected to another by the same subject or the same object but we exclude the connection by the same class. For this,

function $findStatements(r_j, r_k)$
$/ *$ find RDF triples with r_j and r_k $* /$
if $r_j, r_k \in P_{G_I}$
    return $\{\langle s_1, r_j, o_1\rangle, \langle s_2, r_k, o_2\rangle \in G_I | o_1 = s_2 \vee s_1 = o_2 \vee s_1 = s_2\}$
else
    return $\{\langle s, p, o\rangle \in G_I | \{r_j, r_k\} \subseteq \{s, p, o\}\}$

Fig. 6. Semantic matching, $findStatements(r_j, r_k)$

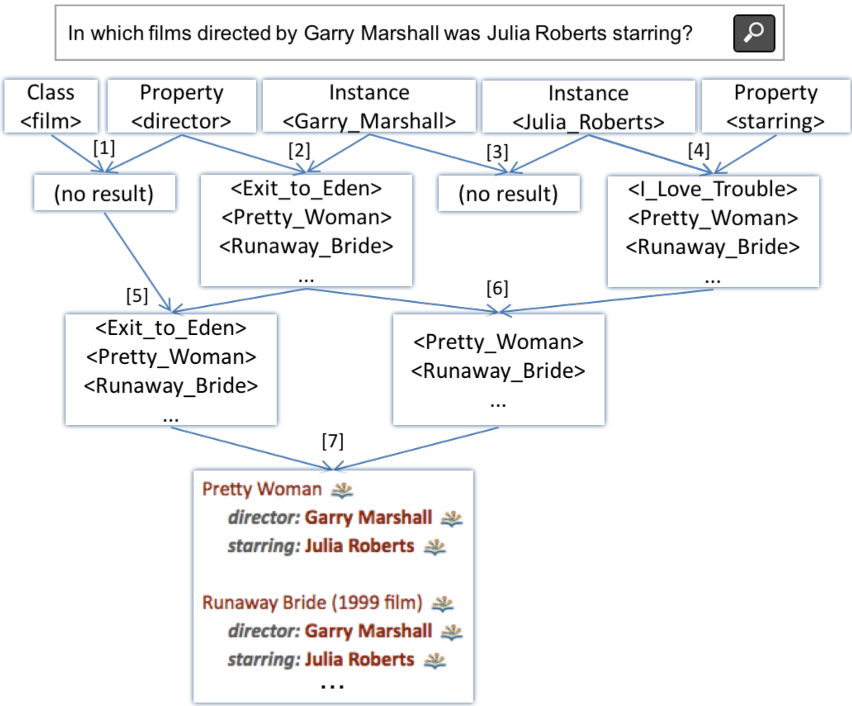

Fig. 7. Example for the triple-based fact retrieval process

we pick a triple $\langle s_i, p_i, o_i\rangle$ from the result and add all other triples $\langle s_j, p_j, o_j\rangle$ to the subgraph where

$$s_i = s_j \text{ or } s_i = o_j \text{ or } s_j = o_i \text{ or } o_i = o_j,\ o_i \notin C_{G_i}. \tag{7}$$

One subgraph is also a set of connected triples where the connection by class is excluded or it is one triple if there are no connected triples in the result set. The ranking is based on the $w_{t_ir_j}$ which are computed in the syntactic matching step by exact (user's choice from autocompletion) or n-gram match. The weight of a matched triple is the sum of the $w_{t_ir_j}$ of participating ontological elements r_j. Each expansion with new triples increases the weight of the partial result by the appropriate $w_{t_ir_j}$ value. Finally, the rank of a result in S_{G_I} is the sum of participating elements' weights divided by the number of query terms. Resolving ambiguities is supported by triple-based processing. Since we do not directly transform the user query to an RDF query, the triples found step by step lead to possible interpretations based on the existing triples in G_I.

As mentioned above, we use the matched resources to expand the query and perform a keyword search in the document index.

3.2.3. Hybrid search

Spreading Activation (SA) is applied to carry out hybrid semantic search. The idea of using SA for information retrieval is to find more relevant information based on retrieved information elements by exploiting associations represented by semantic networks (as graphs). The ontological concepts are the nodes, the properties the edges of the network, usually directed and weighted. SA starts with the initial incoming activation of nodes which propagate the activation along the edges activating the connected nodes. This process is iterated, *i. e.*, the activation spreads through the network, until the stop condition is fulfilled. Result is the activation level of each node at termination time [29].

The process can be devided into 3 steps:

1. Activation network set up: For SA, we extend the formal knowledge base by document nodes, *i. e.*, we instantiate our documents. To setup the activation network, the set of matched resources of the fact retrieval result set is to extract where we differentiate between instances, properties and classes. For the extraction of the classes and the properties only the results of the syntactic matching are relevant. This restriction helps to avoid noisy results by to wide spreading and spreading independently from the subject asked by the query (instances) respectively. To get the matched instances, we extract not only the resources found by the syntactic matching, but also subjects and objects from the triples found by the semantic matching. These resources and the instantiations of documents (that were found by keyword search in the text index) are our activation nodes, their initial weight is given by their ranks. We assign all edges with the matched properties their rank and a default weight to all other edges.
2. Spreading: To spread the network in an iterative procedure, we apply the activation function

$$I_j = \sum_i O_i w_{ij}(1-\alpha) \tag{8}$$

   where I_i is the incoming activation of nodes, O_i the outgoing activation (determined by an output function), $w_{ij} \in [0,1] \subset \mathbb{R}$ is the weight of the edge from n_i to n_j, and α is an attenuation factor which decreases the activation strength with each propagation [29]. Since both α and the edge weights already decrease the activation level we use the simple output function $O_i = I_i$. We apply an activation constraint which stops spreading at a node when its activation level does not exceed a defined threshold. Furthermore, a fan-out constraint averts the danger of a too wide spreading through nodes with high connectivity, thus to become noise in results. Both constraints are parameterizable in order to be able to adjust them to data sets. It is important to assure that each directed edge is processed only once in order to avoid endless spreading in cycles. In each iteration, the node with the highest activation level and pending edges is processed. SA stops when no more nodes have an activation level above the defined threshold or the nodes above the threshold have no pending edges.
3. Result determination: The result of the activation process is a set of weighted nodes. The last step of the hybrid approach is to extract the single search results from the activation network, thus, to determine the coherent subgraphs. Since facts found by the fact retrieval are part of the set of starting points for spreading, they are also part of the results and offer a starting point for an effective subgraph extraction. Therefore, the result determination process starts by the fact retrieval results and add connected documents and resources from the results of the SA process. Note that the subgraphs also contain the properties which are not spread since they have a literal value but are matched by the fact retrieval. For all other found documents, the results of SA define a set of facts which describes the answer and which is also part of the result. We only collect the high ranked resources as facts to avoid confusing results. The rank of one object from the result set is the average rank of the contained information elements.

Semantic document retrieval, which is performed when no facts were found, applies the same procedure. The difference is, that only document nodes are initially activated and every edge has the same default weight. The results are the documents that belong to weighted document-nodes of the activation network.

3.3. Ranking

In general, semantic document retrieval and fact retrieval applies different ranking strategies. Thus, for a hybrid semantic search which supports hybrid results and result lists, a ranking model is needed, which can combine different ranking functions. The range of the single ranking functions have to be the same or at least adjustable to each other. The vector space model with $tfidf$ with the Cosine-similarity (cf. [25]) and a fact ranking based on nGram-similarity with Dice-measure (cf. [24]) fulfill this requirement. The range of both functions is restricted to $[0, 1]$ and they are very similar vector-similarity measures. Though, facts tend to be ranked slightly higher due to the differences in the lengths and structure of the compared vectors. The vectors of a query term and a fact label (fact retrieval) are often more similar in length and have more probably more matching parts than the vectors of a query compared to a document vector (document retrieval). However, given that formal structured information like facts are precise, unlike unstructured information like documents, we assume that facts more likely satisfy the user' information needs than documents.

The combination of facts and documents is given by the spreading activation, which ranks all results by the same function. In detail, the weights computed by fact matching and document retrieval flow into the spreading activation as initial weights. Furthermore, based on the edge weights, it is possible to boost documents or resources when it is necessary due to the underlying information pool and search goals. For the evaluations described in this paper, neither the documents nor the facts were boosted. The rank of final results is computed as the average of the weights of the participating nodes.

In summary, SINFIO performs a semantic ranking[6] that includes the lexical similarity, the rank of the documents, the distance of concepts (by spreading activation), and the coverage of the search query.

3.4. Result presentation

The aim of the result presentation design is to support the user by an interpretable representation of facts, documents and hybrid results, and also result list which can be a mix of them.

For text documents, the title and a document snippet containing the search terms within their textual context in the document (keywords in context) is displayed [31]. The keywords are highlighted by bold formatting. This presentation is based on several user studies, *e. g.*, [32–34], and it is applied by popular search engines, *i. e.*, it is well-known to the users.

[6]Formal knowledge bases enable semantic search engines to carry out the ranking on a semantic basis instead of calculating lexical similarities. The so-called semantic ranking reverts to the structure of the knowledge base [30].

Graphs (*e. g.*, [14, 35]) or a structured textual representation (*e. g.*, [4, 10, 36]) are used to represent facts. However, the presentation of document, fact, and hybrid results should rely on an abstraction of the different result types into a similar form (cf. [22]). In order to hide the complexity from the user and to support result interpretation, facts are presented as structured text.

Furthermore, *documents and facts* are presented in different colors and properties are italicized in the result presentation too (like by the autocompletion component). Figures 8, 9 and 10 illustrate the document, fact and the hybrid result representation.

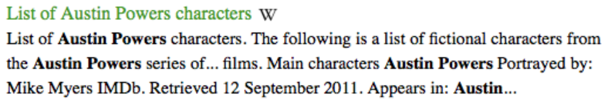

Fig. 8. Document presentation example.

Fig. 9. Document (top) and fact (bottom) presentation.

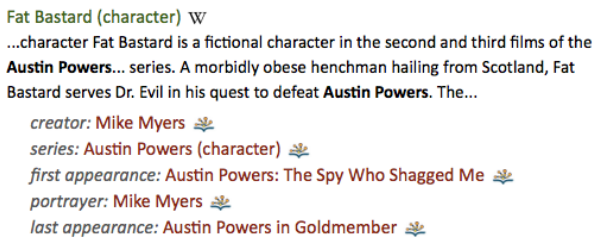

Fig. 10. Document (top) and fact (bottom) presentation.

Since this principle can not be transferred directly to the results lists that contain all kinds of result types, SINFIO offers two alternatives to the user: a result list ordered by rank and a list which is grouped by result types. The latter allows quick access to results, if users know exactly what they are looking for or are interested in a particular type of results. Figures 11 and 12 shows the two variants of result organizations.

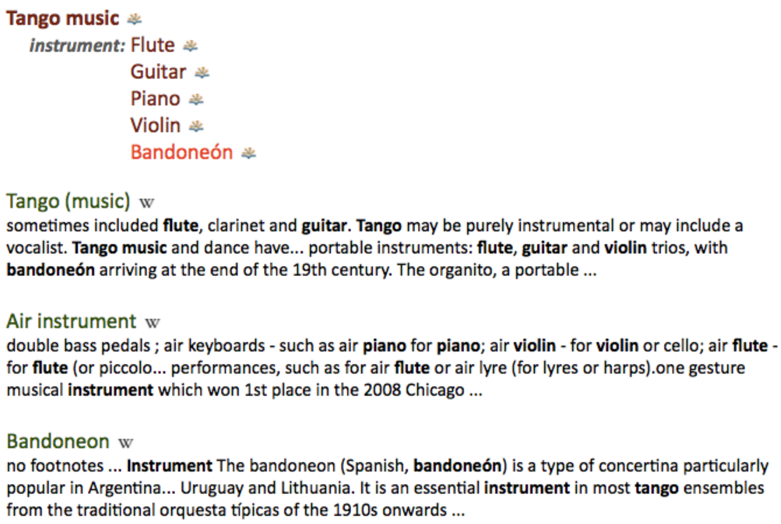

Fig. 11. The result list is ordered by rank.

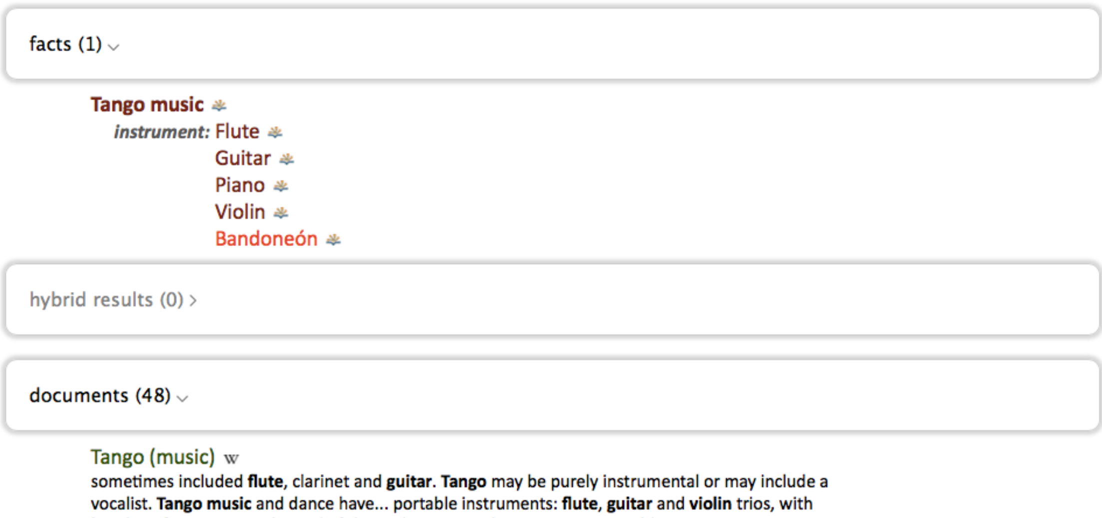

Fig. 12. The grouped view of the results.

4. Evaluations

The evaluations are based on a **data set** including the English version of the online encyclopedia Wikipedia and its corresponding Linked Data version DBpedia[7].

The 20 natural language **queries** are randomly selected from DBpedia's query logs [37] in a way that the distribution of the 20 queries by length correlates to the distribution in query logs and the features of the compared search engines are covered to the same extend (cf. [38]). For example "austin powers" is a simple query while "Give me all actors starring in movies directed by and starring William Shatner" is a complex one. The collection of queries in [37] covers both fact retrieval and semantic document retrieval and it is suitable for the comparison of the evaluated search engines.

With respect to **relevance and IR-measures**, comparing the hybrid search SINFIO with stand-alone fact

[7]DBpedia v3.9 covers about 4 million concepts, 470 million triples. The corresponding Wikipedia contains about 4.2 million articles.

retrieval, stand-alone semantic document retrieval and a hybrid search that does not combine structured and un-structured content throughout the entire search process presents special challenges, since:

- The search engines deliver result lists of different lengths. In particular, the number of results of the fact search can often be less than k and several correct results are possible. In this case, only precision and recall are suitable for measuring the retrieval effectiveness. They are based on the number of relevant documents found in the first k results and the number of relevant documents in the document pool[8].
- Binary relevance judgements are less suitable for different types of results than for homogeneous result sets, as they do not allow the differentiation of relevance and thus the quality of results. In particular, if the results can be facts, documents and hybrid, the question of whether a search result completely or only partially answers the search query is significant. From the point of view of the assessors, the possibility of a differentiation of relevance levels is essential, in many cases a “yes / no” decision is difficult to make [41, 42]. Therefore, the question was to what extent a search result answers the search query with the 5 relevance levels answer set "yes", "partial", "neither yes nor no", "rather no" and "no". The level "neither yes nor no" is used to express that, although a result has something to do with the search query, a relevance decision can not be made.

Due to the application of graded relevance and different lengths of the result lists[9], the standard precision and recall cannot be applied but the generalized variations are suitable. The *generalized Precision gP and Recall gR* of n results d with relevance-score $r(d)$ are defined as follows [43]:

$$gP = \frac{\sum_{d \in R} r(d)}{n}, \; gR = \frac{\sum_{d \in R} r(d)}{\sum_{d \in D} r(d)}. \tag{9}$$

[8] MRR looks at the position of the first correct answer, it is aimed at evaluating search queries with a well-known answer. MAP is applicable if several correct answers are possible but all relevant documents must be known and k is to be chosen as large that for each search query all relevant results are included. Therefore, MAP can only be calculated using a Gold Standard that has relevance to all (or a sufficiently large number) of results [39, 40].

[9] In particular, the fact retrieval can yield less than 10 results (*e. g.*, there are only 7 main Canary Islands).

We have applied the relevance-scores 1.0, 0.75, 0.5, 0.25 and 0.0 and the evaluations have been carried out according to the pooling method with the top 10 results[10].

The **gold standard** is created from the top 10 search results of fact retrieval, semantic document retrieval, the hybrid search SINFIO and the hybrid search that does not combine structured and un-structured content throughout the entire search process. Altogether 14 information scientists and professionals, who are able to carry out a relevance assessment due to their professional background (cf. [39]), are asked to rate the extent to which a search result answers the query.

The overall Kappa value[11] amounts to 0.593 and achieved with a standard error of 0.007 a high statistical significance (cf. [44]). It is also located at the upper limit of the interval of fair match ([0.41-0.6], cf. [45]). An evaluation of the interim result (7 assessors) and its comparison with the final result (14 assessors) explains why a doubling of the number of participants doesn’t remarkable increase the kappa value: While some of the participants expect enumeration, *e. g.*, companies in Munich, in one result, others expect one result per company.

The following sections present the results of the evaluations with respect to effectivity, ranking and the user interface. We have tested the statistical significance of the achieved results using Fisher’s randomization test (cf. [46, 47]) with $\alpha = 0.5$ and applied the Standard Pearson coefficient for correlation analysis (cf. [48, 49]). Since the Standard Pearson correlation is based on the assumption that there is a linear correlation, we have also cross-checked all results closed-by 0.0 using the Spearman correlation.

4.1. Effectivity

The effectivity is *evaluated using IR-measures as well as user studies*. The evaluations compare SINFIO to a hybrid approach which do not combine structured and semi- or unstructured content throughout the entire search process, in the following referred to as FSDR. The **hypothesis** is:

[10] Note that the values of gP and gR are dependent on the relevance level-values, *i. e.*, they can only be interpreted in relation to the search methods considered in this study. For the chosen relevance values and top 10 results, the maximum gP/gR is 1.0.

[11] The Kappa-statistic calculates the rate of agreement among relevance judgments (cf. [39]).

H1: The hybrid semantic search SINFIO achieves a higher level of retrieval effectiveness than FSDR, *i. e.*, combining formal and informal content throughout the entire search process achieves higher level of retrieval effectiveness than accomplishing semantic document retrieval and fact retrieval without a combination of formal and informal content during the search process.

Furthermore, SINFIO has been compared to the standalone semantic document retrieval as well as fact retrieval with the following *hypotheses*:

H2: Compared to semantic document retrieval, users are able to meet their information requirements more quickly by means of the hybrid semantic search.
H3: Compared to fact retrieval, the hybrid semantic search contributes more to the satisfaction of the users' information needs.

FSDR is realized as an alternative implementation of SINFIO without the combination of facts and documents throughout the entire search process but answering queries with facts and documents. The fact retrieval and the semantic document retrieval are evaluated as they are implemented for SINFIO.

The interpolated generalized **precision at the standard recall-levels** (cf. [40]) with respect to hypothesis H1 is depicted figure 13. SINFIO achieves the highest precision on all recall levels. The advantage over FSDR increases with every recall level, *i. e.*, the more of the relevant results are found, the higher is the precision of SINFIO in comparison to FSDR. Furthermore, for every query with hybrid results (3 of 20) SINFIO achieves better results. The difference of the F-measure ($\beta = 1.0$, *i. e.*, precision and recall count equally) is statistically not significant ($p = 0.10$, $\alpha = 0.05$): The probability that the difference of 10 percentage points will occur even if both systems perform equally well is 10%. In detail, the highest precisions occur for queries with mostly facts or hybrid results only, and the lowest values for queries without facts or exactly one fact and only a few relevant documents.

Table 2 lists the average gP, gR and F-measure over all queries for all approaches. The fact retrieval achieves the highest precision while SINFIO achieves the highest recall. Overall, the best F-Measure is achieved by SINFIO with statistically significant differences to the fact retrieval ($p < 0,004$) and as well as to the semantic document retrieval ($p < 0,0006$).

Table 2
Average gPrecision and gRecall

Approach	gPrecision	gRecall	F-measure
SINFIO	0.4951	0.9398	0.6485
FSDR	0.4421	0.7059	0.5437
Fact Retrieval	0.6550	0.3737	0.4785
Sem. Doc. Retrieval	0.2913	0.4459	0.3523

The **user-based evaluation** was performed using side-by-side panels that display the results of SINFIO and one of the other approaches in the same style and random order. The twenty participants (12 female, 8 male) had different professions such as, e.g., psychologist, cartographer, clerk, artist, craftsman, journalist, economist, lawyer, educator, housewife *etc.* and were of age between 20 and 60 years. In order to test our hypotheses, we asked:

Q1: Which result set they assess to be better;
Q2: If the answer was clear based on the displayed result set itself or they needed to read linked Wikipedia and DBpedia articles;
Q3: Based on which results set they found the answer faster.

The participants had also the possibility to give a statement of reasons for each answer.

Q1 – As shown in figure 14, the user rated the result set of SINFIO as the best. All differences are statistically significant whit $p <= 0.0001$. The reason for the high proportion of "equals" is that for many queries the top 10 answers of two search approaches are the same. For example, SINFIO and FSDR answer 7 of the 20 queries with identical results. Considering only queries with different result lists, SINFIO achieved 68.46% while FSDR 5.77% and 30.77% of the answers were "equals".

Altogether, this user-based evaluation and the evaluations based on F-measure *confirm our hypothesis H1*, but only the results of the user study are statistically significant.

Q2 – Table 3 summarizes the results of the question, if the answer was clear based on the displayed result set or the participants had to read the linked articles. The results suggest that hybrid search in general enables the users to find the answers by looking at the result sets more often then for fact retrieval and semantic document retrieval. Furthermore, this holds for

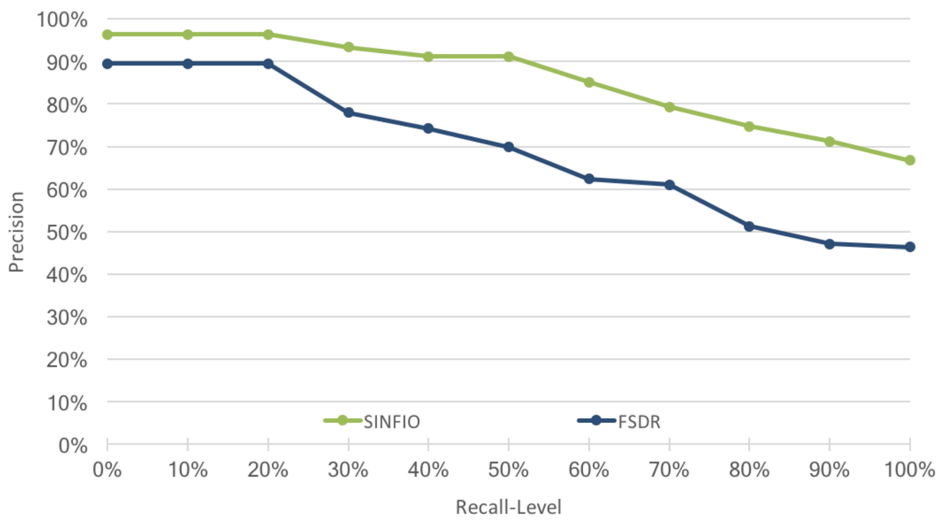

Fig. 13. Generalized precision/recall diagram

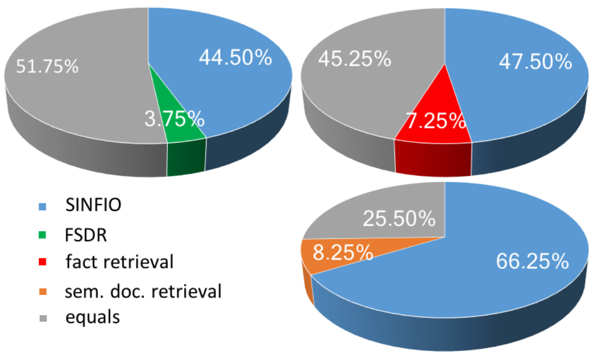

Fig. 14. Results of the user-based effectivity evaluation

Table 3

Results of the question, if the answer was clear based on the displayed result set or the user had to read the linked articles

Approach	results set	links opened
SINFIO	81.15%	18.85%
FSDR	62.31%	37.69%
SINFIO	79.67%	20.33%
Fact Retrieval	51.00%	49.00%
SINFIO	84.17%	15.83%
Sem. Doc. Retrieval	36.94%	63.06%

SINFIO more often compared to FSDR. All achieved results are statistically significant ($p <= 0.049$).

Q3 – Figure 15 gives an overview on the question in which result set the users were able to determine the answer faster. Consistent with the results of the previous question, SINFIO was rated best. The difference between SINFIO and all other search approaches is statistically significant ($p <= 0.0068$).

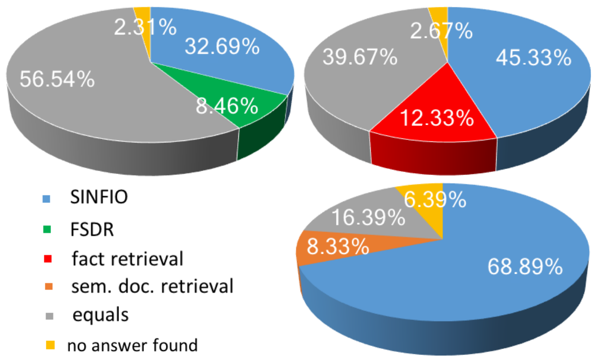

Fig. 15. Results of the question where the answer can be found faster

The results of the last two questions and the F-measures suggest that the hypotheses H2 and H3 are also true. A confirmation is only possible based on he analysis of the participants' explanatory statements.

With respect to H2: Compared to the semantic document retrieval, SINFIO is preferred because the answer is clearly and precisely at the top of the result list and it is highlighted by the red color. In addition, documents found by SINFIO are more relevant. With respect to H3: The users mainly prefer SINFIO against fact retrieval because of the additional information provided with the retrieved documents. Furthermore, in comparison to FSDR, users prefer SINFIO because they value its achieved results as more detailed and meaningful. Altogether, *hypotheses H2 and H3 are confirmed*.

Two adverse points have also been noted: The sorting and completeness of the answers is not always clear. These comments relate to SINFIO as well as to the fact retrieval approach and have been noted in case of the first ten results being facts. Since only the top 10 results are displayed, the completeness cannot be correctly considered from the answers only. The sorting, *i. e.*, the rank of the achieved results in those cases are the same. The order follows the random order given by the result set of the SPARQL-queries applied for fact retrieval.

4.2. Ranking

The ranking method is evaluated based on its Spearman correlation to the "ideal" ranking which is determined by the gold standard[12]. The ranking of SINFIO shows a very high correlation to the "ideal" ranking with a Spearman coefficient of 0.9922. This result suggests that the slightly higher ranking of facts than documents meets the users' needs (cf. section 3.3).

4.3. User Interface

According to the challenges formulated in section 1, we have evaluated the usage and advantages of the *autocompletion* feature, the intelligibility of the *result presentation*, and asked the users which *result list presentation* they prefer.

The **semantic autocompletion** has been developed in order to support users in creating queries with as many formal parts as possible. At the same time, users should not need to learn a specific query language or be familiar with the underlying knowledge base. To evaluate our component, we performed a task-based user study with overall 20 participants: 9 female, 11 male, all between 21-35 years old, with different professions. The tasks were to find the answer to 5 predefined questions that were selected from the DBpedia-query log. The participants had no knowledge about the underlying knowledge base and they were not familiar with search engine research. We did only inform them that this test is used to evaluate a novel search engine that searches DBpedia and Wikipedia and returns results from both. An example of a hybrid result has also been demonstrated to the users in advance. They then performed the tasks and were asked when and why they did not use the suggested autocompletion. We also logged the frequency of accepting suggested concepts as well as ignoring the suggestions, the number of queries per task, and asked the participants to press a button as soon as they found the answer to the question.

Figure 16 shows that the suggestion have beed accepted most times ($56 - 95\%$). The participants did not use the component when they were not able to decide whether one of the suggestions was related to what they were asking for, or where none of the suggestions did exactly fit. Some participants mentioned "found" as a negative example, where the first suggestions are "founded by", "foundation pace" and "founding person", since no one of these suggestions relates to time. In this case, the strategies of the participants were either not to accept any of the suggestions or to remove the corresponding word from the search query and then to continue with the results obtained.

The task duration is depicted in figure 17. The first and the second task took the most time with 96.7 and 103.5 seconds. 60% of the participants succeeded to find the solution using 2 and 40% using 3 queries. In these cases the answer could not be read in the results list, the participants had to open Wikipedia article pages and to seek for the required information. The tasks 3, 4 and 5 were answered by all participants with only one query and the average task duration was between 32.0 and 56.3 seconds.

The achieved results show that *the semantic autocompletion is accepted by the users* and can be applied without knowledge of the underlying knowledge base. Using semantic autocompletion in SINFIO *leads to search queries with formal parts*. Furthermore, the statistically significant strong negative linear correlation between accepting suggestions and the task duration with a coefficient of -0.91 shows that the *queries with the most formal parts were answered most quickly*.

The **result presentation** has been evaluated in a comparative study with side-by-side panels and the 20 participants of the study on effectiveness. The participants were asked to indicate for each side of the panels whether they can understand the presentation of the search results or not. Figure 18 presents the results. The participants rated the result presentation of SINFIO in average of all comparisons to 91.38% as comprehensible. Despite the hybrid results, which we assume to be more complex, SINFIO had the most comprehensible results compared to FSDR, the fact retrieval and the semantic document retrieval approach.

[12]The range of the Spearman coefficient is $[-1.0,\ 1.0]$ where -1 means the maximum difference and $+1$ the perfect correlation.

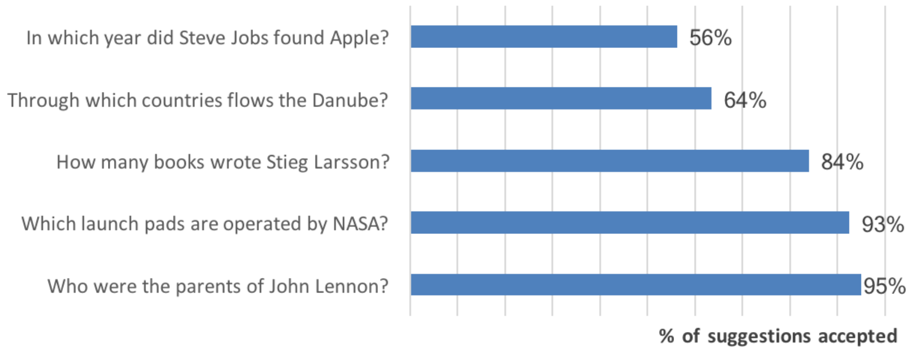

Fig. 16. Average frequency of using autocompletion

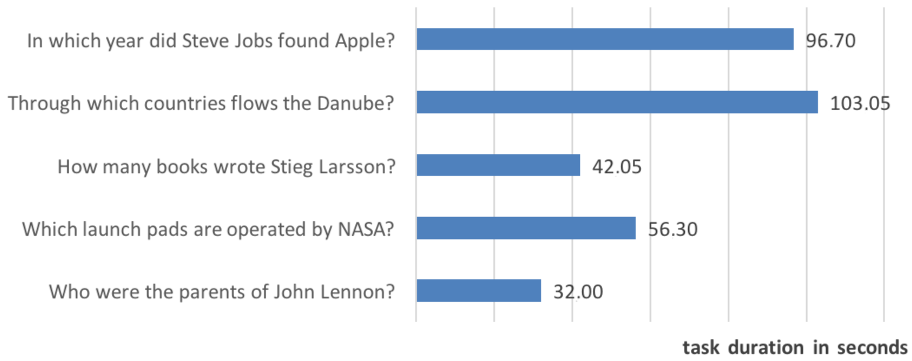

Fig. 17. Average task duration

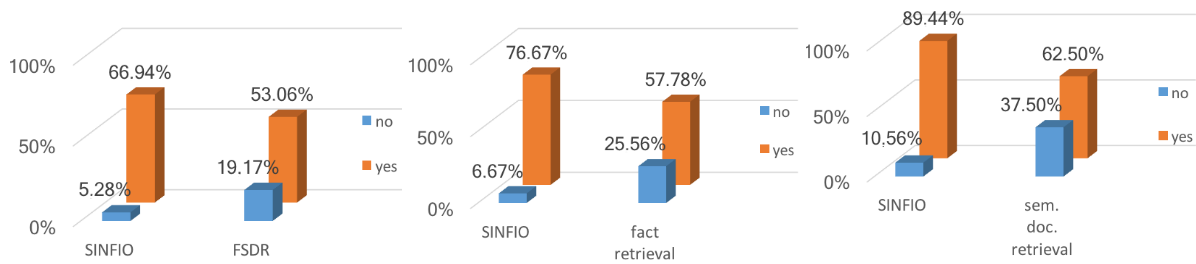

Fig. 18. Results of the survey on understanding the result presentation

The differences are statistically significant with p-values $<= 0.0068$. An analysis of the reasons implies that the participants took also the quality of the results into account. Participants mentioned the eye-catching red color of the “precise answers” as very useful. The users noted that facts and hybrid results are presented clearly arranged. The document presentation has not been commented. However, it follows the usual representation of the most commonly used web search engines. In detail, the three result sets with hybrid results were rated with 91.7%, 83.33% and 66.67% to be comprehensible. Two of these ratings are below the average. Following the participants’ notes, the completeness of the answers is unclear. This issue is caused by the cut off $k = 10$ as it is already discussed in section 4.1.

The survey on the two alternatives of the **result list presentation** was also carried out with side-by-side panels and 20 participants. The demographic data of the participants were similar to those in the study on effectiveness, whereby the age distribution has been shifted in favor of the > 40-year-olds. The panels contained the first 80 results of each approach. In average, participants preferred in 67.5% of the cases the result list ordered by rank, *i. e.*, it can contain a mix of facts, hybrid results and documents. They rated only in 16.25% of the cases for the result list grouped by the type of the results (and ordered by rank in each group) and rated them in the same amount of cases as "equals". All results are statistically significant with a p closed-by 0. In detail, the grouped view was never rated to be better than the ranked view and the highest score for being "equals" amounts to 30%.

4.4. Correlation analysis

The aim of this analysis was to find out which of the assessed factors has the greatest influence on judging SINFIO to be better than all other approaches. We considered the (positive) results of the effectivity evaluations and also the results of the search result presentation. The results of "which result set was rated to be better" show

- a mean linear correlation (0.54) with the results of "where the answer could be found faster",
- a weak linear correlation (0.23) to the comprehensibility of result presentation
- but, surprisingly, no correlation to the fact that the answer was clear based on the displayed results.

Correlation coefficients represent only a statistical but no causal relationship. Therefore, the measured correlations can only be interpreted assuming that the factors we asked play a role by the (comparative) positive assessments of the top 10 search results. Under this assumption, the correlations suggest that the speed of at which users were able to find a response to their query is most important among these three factors. A weak influence on a positive assessment of the search result quality had the comprehensibility of the result presentation, and it did not depend on the readability of the answer in the results list. This may mean that when evaluating the top 10 search results, users do not focus on finding the answer but will look at the results and evaluate their overall quality.

Furthermore, there is no noteworthy linear correlation among the F-measures and the results of the user-studies. This shows the difference between objective and subjective judgment on search approaches: users seem to consider more factors than only the (subjectively perceived) relevance of the results.

5. Conclusions

The hybrid semantic search approach SINFIO, which combines structured and semi- or unstructured content throughout the entire search process, is more effective than a hybrid approach which searches for facts and documents without combining them. SINFIO demonstrates the possibility of combining facts and documents according to query and to rank the results in an appropriate way. The evaluations show that users accept, understand and prefer SINFIO to hybrid search, which does not combine facts and documents, to stand-alone fact retrieval as well as semantic document retrieval. Moreover, the answer to a query can be found more quickly by SINFIO. Thus, the hybrid semantic search is the best method for users to satisfy their information needs from differently structured data sets: it is more effective than pure fact retrieval, semantic document retrieval or fact and document retrieval without a combination of the contents.

According to the evaluations, we are planning to extend the fact retrieval's ranking method. In order to more meaningfully arrange the results with the same rank, we plan to include information about the general popularity of the resources. In future versions, we also foresee to expand our approach with contextualization and personalization. Context information as well as user-specific information can be involved in form of weights in our activation network. Depending on if/how this information is to be strengthened or weakened over time, predefined or dynamic weights can be applied. Using dynamic weights, for example earlier searches are retained in the memory of the network, but they have less influence the longer they date back.

References

[1] J. Bock, F. Chaves, B. Forcher, S. Grimm, J. Henß§, J. Kleb, M. Sintek, T. Tserendorj and R. Volz, System Overview, Technical Report, 2008, Project Deliverable.

[2] K. Schumacher, B. Forcher and D.T. Tran, Semantische Suche, in: *Semantische Technologien – Grundlagen. Konzepte. Anwendungen.*, Spektrum Verlag, 2011, pp. 227–252.

[3] C. Rocha, D. Schwabe and M.P. Aragao, A hybrid approach for searching in the semantic web, in: *Proceedings of the 13th International Conference on World Wide Web*, WWW, ACM, 2004, pp. 374–383. ISBN 1-58113-844-X.

[4] R. Guha, R. McCool and E. Miller, Semantic Search, in: *Proceedings of the 12th International Conference on Word Wide Web*, 2003.

[5] A. Kiryakov, B. Popov, I. Terziev, D. Manov and D. Ognyanoff, Semantic annotation, indexing, and retrieval, *Journal of Web Semantics* **2** (2004), 49–79.

[6] A.J. Zitzelberger, D.W. Embley, S.W. Liddle and D.T. Scott, HyKSS: Hybrid keyword and semantic search, *Journal on Data Semantics* (2014), 1–17.

[7] V. Tablan, K. Bontcheva, I. Roberts and H. Cunningham, Mímir: An open-source semantic search framework for interactive information seeking and discovery, *Web Semantics: Science, Services and Agents on the World Wide Web* **30** (2015), 52–68.

[8] M. Gärtner, A. Rauber and H. Berger, Bridging structured and unstructured data via hybrid semantic search and interactive ontology-enhanced query formulation, *Knowledge and Information Systems* **41**(3) (2014), 761–792.

[9] V. Lopez, E. Motta and V.S. Uren, Poweraqua: Fishing the semantic web, in: *The Semantic Web: research and applications*, Springer, 2006, pp. 393–410.

[10] V. Lopez, M. Pasin and E. Motta, Aqualog: An ontology-portable question answering system for the semantic web, in: *The Semantic Web: Research and Applications*, Springer, 2005, pp. 546–562.

[11] V. Lopez, M. Fernández, E. Motta and N. Stieler, PowerAqua: Supporting users in querying and exploring the Semantic Web, *Semantic Web* **3**(3) (2012), 249–265.

[12] V.S. Uren, M. Sabou, E. Motta, M. Fernandez, V. Lopez and Y. Lei, Reflections on Five Years of Evaluating Semantic Search Systems, *International Journal of Metadata, Semantics and Ontologies* **5**(2) (2010), 87–98, ISSN 1744-2621.

[13] M. Fernandez, V. Lopez, M. Sabou, V.S. Uren, D. Vallet, E. Motta and P. Castells, Semantic Search Meets the Web, in: *Proceedings of the IEEE International Conference on Semantic Computing*, ICSC '08, IEEE Computer Society, 2008, pp. 253–260. ISBN 978-0-7695-3279-0.

[14] R. Bhagdev, S. Chapman, F. Ciravegna, V. Lanfranchi and D. Petrelli, Hybrid Search: effectively combining keywords and semantic searches, in: *Proceedings of the 5th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC08, Springer, 2008, pp. 554–568.

[15] H. Wang, D.T. Tran, C. Liu and L. Fu, Lightweight integration of IR and DB for scalable hybrid search with integrated ranking support, *Web Semantics: Science, Services and Agents on the World Wide Web* **9**(4) (2011), 490–503.

[16] K. Schumacher and M. Sintek, Searching Web 3.0 Content, in: *Proceedings of the 7th International Conference on Web Information Systems and Technologies.*, 2011.

[17] C. Abras, D. Maloney-Krichmar and J. Preece, User-Centered Design, in: *Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, Publications, 2004.

[18] E. Kaufmann and A. Bernstein, How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users?, in: *Proceedings of the ISWC/ASWC*, Lecture Notes in Computer Science, Vol. 4825, Springer, 2007, pp. 281–294. ISBN 978-3-540-76297-3.

[19] M. Reichert, S. Linckels, C. Meinel and T. Engel, Student's Perception of a Semantic Search Engine., in: *Proceedings of the CELDA*, 2005, pp. 139–147.

[20] J. Osterhoff, J. Waitelonis and H. Sack, Widen the Peepholes! Entity-based Auto-Suggestion as a rich and yet immediate Starting Point for Exploratory Search., in: *GI-Jahrestagung*, 2012, pp. 1039–1046.

[21] E. Hyvönen and E. Mäkelä, Semantic autocompletion, in: *The Semantic Web–ASWC 2006*, Springer, 2006, pp. 739–751.

[22] R.E. Mayer and R. Moreno, Nine ways to reduce cognitive load in multimedia learning, *Educational psychologist* **38**(1) (2003), 43–52.

[23] D.E. Goldschmidt and M. Krishnamoorthy, Architecting a search engine for the Semantic Web, in: *AAAI Workshop on Contexts and Ontologies: Theory, Practice and Applications*, 2005.

[24] G. Kondrak, N-gram similarity and distance, in: *Proceedings of the 12th International Conference on String Processing and Information Retrieval*, 2005, pp. 115–126.

[25] C.J. van Rijsbergen, *Information Retrieval*, 2nd edn, Butterworth-Heinemann, 1979. ISBN 0408709294.

[26] C.Y. Suen, N-gram statistics for natural language understanding and text processing, *IEEE transactions on pattern analysis and machine intelligence* (1979), 164–172.

[27] K. Kukich, Techniques for automatically correcting words in text, *ACM Computing Surveys (CSUR)* **24**(4) (1992), 377–439.

[28] B. van Berkel and K.D. Smedt, Triphone analysis: a combined method for the correction of orthographical and typographical errors, in: *Proceedings of the 2nd Conference on Applied Natural Language Processing*, Association for Computational Linguistics, 1988, pp. 77–83.

[29] F. Crestani, Application of Spreading Activation Techniques in Information Retrieval, *Artificial Intelligence Review* **11**(6) (1997), 453–482, ISSN 0269-2821.

[30] N. Stojanovic, A. Maedche, S. Staab, R. Studer and Y. Sure, SEAL: a framework for developing SEmantic PortALs, in: *Proceedings of the 1st International Conference on Knowledge Capture*, ACM, 2001, pp. 155–162.

[31] M. Hearst, *Search user interfaces*, Cambridge University Press, 2009.

[32] A. Tombros and M. Sanderson, Advantages of query biased summaries in information retrieval, in: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 1998, pp. 2–10.

[33] R.W. White, J.M. Jose and I. Ruthven, A task-oriented study on the influencing effects of query-biased summarisation in web searching, *Information Processing & Management* **39**(5) (2003), 707–733.

[34] H.P. Luhn, Key word-in-context index for technical literature (kwic index), *American Documentation* **11**(4) (1960), 288–295.

[35] B. Aleman-Meza, C. Halaschek-Weiner, I.B. Arpinar, C. Ramakrishnan and A.P. Sheth, Ranking complex relationships on the semantic web, *Internet Computing, IEEE* **9**(3) (2005), 37–44.

[36] Y. Lei, V.S. Uren and E. Motta, SemSearch: A Search Engine for the Semantic Web, in: *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management*, 2006, pp. 238–245.

[37] K. Balog and R. Neumayer, A Test Collection for Entity Retrieval in DBpedia, in: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, ACM, 2013, pp. 737–740.

[38] M. Gordon and P. Pathak, Finding Information on the World Wide Web: The Retrieval Effectiveness of Search Engines, *Information processing & management* **35**(2) (1999), 141–180, ISSN 0306-4573.

[39] C.D. Manning, P. Raghavan and H. Schütze, Evaluation in information retrieval, in: *Introduction to Information Retrieval*, Cambridge University Press, 2008, pp. 151–176. ISBN 0521865719, 9780521865715.

[40] R. Baeza-Yates and B. Ribeiro-Neto, Retrieval Evaluation, in: *Modern Information Retrieval*, 2nd edn, Addison Wesley, 2011, pp. 131–176.

[41] S.E. Robertson, The probability ranking principle in IR, *Journal of documentation* **33**(4) (1977), 294–304.

[42] S.E. Robertson and N.J. Belkin, Ranking in principle, *Journal of Documentation* **34**(2) (1978), 93–100.

[43] J. Kekäläinen and K. Järvelin, Using graded relevance assessments in IR evaluation, *Journal of the American Society for Information Science and Technology* **53**(13) (2002), 1120–1129.

[44] J.L. Fleiss, The measurement of interrater agreement, *Statistical methods for rates and proportions* **2** (1981), 212–236.

[45] A.J. Viera, J.M. Garrett et al., Understanding interobserver agreement: the kappa statistic, *Fam Med* **37**(5) (2005), 360–363.

[46] M.D. Smucker, J. Allan and B. Carterette, A comparison of statistical significance tests for information retrieval evaluation, in: *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, ACM, 2007, pp. 623–632.

[47] W.E. Webber, Measurement in information retrieval evaluation, PhD thesis, The University of Melbourne, Department of Computer Science and Software Engineering, 2010.

[48] R.A. Johnson, D.W. Wichern et al., *Applied multivariate statistical analysis*, Vol. 4, Prentice hall Englewood Cliffs, NJ, 1992.

[49] J.H. Steiger, Tests for comparing elements of a correlation matrix., *Psychological bulletin* **87**(2) (1980), 245.

[50] E. Kaufmann and A. Bernstein, How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users?, in: *Proceedings of the ISWC/ASWC*, 2007, pp. 281–294.

[51] K. Schumacher, Hybride semantische Suche - eine Kombination aus Fakten- und Dokumentretrieval, doctoralthesis, Universität Potsdam, 2017.