**Deliverable 1.1**  
**Feasibility Study**

<table>
<thead>
<tr>
<th>Dissemination Level</th>
<th>Public</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due Date of Deliverable</td>
<td>Month 4, 31/12/2015</td>
</tr>
<tr>
<td>Actual Submission Date</td>
<td>Month 4, 31/12/2015</td>
</tr>
<tr>
<td>Work Package</td>
<td>WP1, Feasibility Study</td>
</tr>
<tr>
<td>Deliverable</td>
<td>D1.1</td>
</tr>
<tr>
<td>Type</td>
<td>Report</td>
</tr>
<tr>
<td>Approval Status</td>
<td>Work in progress</td>
</tr>
<tr>
<td>Version</td>
<td>0.4</td>
</tr>
<tr>
<td>Number of Pages</td>
<td>13</td>
</tr>
</tbody>
</table>

**Abstract:**  
This report describes first the industrial use cases from Metaphacts and Ontos concerning the enterprise search challenges. Further, it analyses the state-of-the-art technologies in science and industry, and states the feasibility of the DIESEL project.

The information in this document reflects only the author’s views and Eurostars is not liable for any use that may be made of the information contained therein. The information in this document is provided "as is" without guarantee or warranty of any kind, express or implied, including but not limited to the fitness of the information for a particular purpose. The user thereof uses the information at his/ her sole risk and liability.
History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Reason</th>
<th>Revised by</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>15/11/2015</td>
<td>Initial Template</td>
<td>Axel-Cyrille Ngonga Ngomo</td>
</tr>
<tr>
<td>0.2</td>
<td>30/11/2015</td>
<td>Initial Draft</td>
<td>Ricardo Usbeck</td>
</tr>
<tr>
<td>0.3</td>
<td>11/12/2015</td>
<td>Improved Draft</td>
<td>Ricardo Usbeck</td>
</tr>
<tr>
<td>0.3</td>
<td>14/12/2015</td>
<td>Improved Draft and Action Items for partners</td>
<td>Ricardo Usbeck</td>
</tr>
<tr>
<td>0.4</td>
<td>15/12/2015</td>
<td>Finished ULEI Action Items</td>
<td>Ricardo Usbeck</td>
</tr>
</tbody>
</table>

Author List

<table>
<thead>
<tr>
<th>Organization</th>
<th>Name</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>ULEI</td>
<td>Axel-Cyrille Ngonga Ngomo</td>
<td><a href="mailto:ngonga@informatik.uni-leipzig.de">ngonga@informatik.uni-leipzig.de</a></td>
</tr>
<tr>
<td>ULEI</td>
<td>Ricardo Usbeck</td>
<td><a href="mailto:usbeck@informatik.uni-leipzig.de">usbeck@informatik.uni-leipzig.de</a></td>
</tr>
<tr>
<td>Ontos</td>
<td>Alejandra Garcia-Rojas M.</td>
<td><a href="mailto:alejandra.garciarojas@ontos.com">alejandra.garciarojas@ontos.com</a></td>
</tr>
</tbody>
</table>
Contents

1 Introduction 3

2 Use Case Specifications 3

2.1 Use Case I: Medium–Large Enterprise Search and Knowledge Graph 3

2.1.1 Characteristics and Market 3

2.1.2 Elicitation approach 4

2.1.3 Common pain points and customer characteristics 4

2.1.4 Infrastructure at customers 5

2.1.5 Structured/Unstructured Data 5

2.1.6 Outlook and Gains 5

2.2 Use Case II:Wikidata 6

2.2.1 Characteristics and Market 6

2.2.2 Common pain points and customer characteristics 7

2.2.3 Structured/Unstructured Data 7

2.2.4 Infrastructure at customers 7

2.2.5 Outlook and Gain 7

3 State Of The Art 8

3.1 Keyword Search 8

3.2 Hybrid Question Answering 9

4 Feasibility of DIESEL Engine 11

5 Conclusions 11

References 11
1 Introduction

Enterprise search at its current state limits itself to aspects of collecting and storing big structured data volumes. However, state-of-the-art solutions either cannot access large amounts of unstructured or semistructured data (e.g., text, news, social media, web tables) or miss the opportunity to truly understand the users search intention by applying traditional search paradigms.

The first goal of the DIESEL project is to carry out a feasible study which will point out the practicability of the approach. This study focus especially on the scientific aspects of the DIESEL project to gather insights of the research effort needed to realise the DIESEL vision. This vision consist in lowering barriers of data accessibility and data integration of distributed enterprise data. DIESEL will overcome this by integrating diverse data sources and enabling semantic search, both keyword and natural language driven, over distributed large scale knowledge sources. Furthermore, the DIESEL search engine will transform its structured output, as well as what DIESEL understood, to natural language to better inform the user.

First, the partners provide use case descriptions which are used to drive high-level requirements. These use cases will be later described in deliverable D1.3 in depth and provide a fixed specification for the development and research within the DIESEL project. Second, we present an analysis of the state-of-the-art technologies with respect to existing semantic search approaches, from industry as well as from science. The advantages and disadvantages of the existing solutions are discussed. Finally, we conclude with some feasibility statements on the scientific aspects of the project.

2 Use Case Specifications

In this section, we present preliminary use case descriptions from the industrial partners. These use cases aim to show that there is a need in the industry for improving the search of information. A more exhaustive description will be reported in D3.1.¹ in M6.

2.1 Use Case I: Medium–Large Enterprise Search and Knowledge Graph

The following section describes the potential use case for Ontos. It is based on discussions and presentations given to existing and future customers in Switzerland and Germany. The definition of this use case is currently under elicitation with a small set of Ontos potential customers.

2.1.1 Characteristics and Market

The smallest company (Avicom Controls) has 120 employees and operates on a world wide base out of Germany. The key challenge is to find relevant data related to previous offers in a multilingual natural language text and interlink it with user manuals, wiki data and information stored in the ERP and CRM system. The other potential customers are from Switzerland and represent the segment of large companies (>10.000 employees) operating mainly in Switzerland. One company is from the transportation industry and the other company from the telecommunication market. The common characteristics is that they have offices spread around Switzerland, people working mobile and information available in 3-4 different languages. Another commonality is the view on an electronic workplace for all knowledge workers. The main goal is to provide relevant data on all devices around the clock.

¹https://github.com/diesel-project/deliverables/
2.1.2 Elicitation approach

![Mockup Enhanced Search with Enterprise Knowledge Graph](image.png)

Figure 1: Mockup Enhanced Search with Enterprise Knowledge Graph

The elicitation process for the Ontos use cases is based on customer interviews and presentations. The presentation is a pitch of the current Ontos product, the Linked Data Suite (LDS) with mock-ups of the solution vision derived from the DIESEL project. Main focus is on explaining how existing search functions can be enhanced with the DIESEL approach and how an Enterprise Knowledge Graph is supporting the search results. Figure 1 depicts the general idea where a classical keyword search is enhanced with semantic search and additional data from aggregated data stored in the Enterprise Knowledge Graph.

Within the presentation the Ontos team is collecting feedback from the customers and potential customers in relation to the proposed DIESEL vision. A more exhaustive description of the use case will be reported in D1.3.

2.1.3 Common pain points and customer characteristics

Based on the interviews we can provide a high level view on the pain points that have been elaborated. First, we describe the typical user within the organisation that needs access to the data. Jobs / Workplace description (excerpt):

- Sales: Working on proposals and new customer deals.
- Engineers: Solving support cases, developing products, first level support/field support with desktop or mobile device.
- Marketeer: Analysing campaigns and spendings to improve ROI.
- Managers: Better business insights by finding faster status about ongoing activities.

Second, to avoid duplicate work and increase productivity the common pain points can be summarised as follows.
Data is stored in many places without a common access interface. Thus, search does not show all data from the various silos. Furthermore, there is no common data formats which leads to hard integration issues and incomprehensability.

- Relevant information should be available on any device at any time.
- Switching between various systems/apps is very time consuming.
- Multilingual information are hard to search.
- Due to no common vocabulary, customers loose connection within there data.
- How to filter relevant data including external data sources.

In a simple statement: 'If we only knew what we know, we would be 30 percent more productive'.

2.1.4 Infrastructure at customers

Most infrastructure is based on the bring-your-own-device (BYOD) approach. Thus, we have identified a huge variety on mobile devices and operating systems, e.g., Android, iOS and Windows on mobile devices as well as on common desktops Windows, Linux and OSX.

2.1.5 Structured/Unstructured Data

Industrial customers keep their structured data separated in the following stores:

- ERP Data: Mainly SAP and Microsoft Dynamics (Structured data based on RDBMS/SQL)
- CRM Data: Mainly SAP and Microsoft Dynamics (Structured data based on RDBMS/SQL)
- Legacy Data: Own applications based on Relational Database Systems (RDBMS/SQL).

Moreover, unstructured data can be found in a variety of storage silos:

- Enterprise Content Management/Demand Chain management: Many customers use Microsoft Sharepoint but other document stores are in place and include for example OpenText or just File Servers based on Windows Server.
- Existing Search Engines: To our surprise, we have seen that the customers are using (testing) many different approaches including Microsoft FAST, Google Search for Work and Apache lucene (Solr, ElasticSearch). All implementations are based on keyword search and provide only filters by document type (e.g. Word, Excel, PDF etc).

2.1.6 Outlook and Gains

The Ontos use case is to solve the above dilemma of pains. Thus, we elaborated with the customer the idea of integrating different data silos (see figure 2). The goal is, to enhance the existing search interfaces with functions from DIESEL and provide a kind of enterprise knowledge graph as infobox (similar to the google Knowledge Graph). The system shall provide support for natural language processing in different languages (specially English, German and French), extract entities and build
the corresponding query. The federated search should support configurable data sources and allow to identify the origin of the data.

The major benefits (gains) would be that people are more productive/efficient and therefore can also make better business decisions.

![Diagram of different data silos (internal and external) using the Linked Data paradigm](image)

Figure 2: Integration of different data silos (internal and external) using the Linked Data paradigm

## 2.2 Use Case II: Wikidata

This use case focuses on how enterprises can support search utilizing Wikidata². It is not an enterprise search use case per se, instead it aims at enriching, contextualizing and integrating enterprise data with an open knowledge graph.

### 2.2.1 Characteristics and Market

Wikidata is a community-created knowledge graph that acts as the central store of structured data for its Wikimedia sister projects such as Wikipedia. It is free to use for anyone to build own knowledge-driven applications. Since its official launch in 2012, the Wikidata community has gathered and stored several hundred millions of cross-domain knowledge facts about person, places, artifacts, terms, and other entities. These facts include both temporal as well as spatial information, are annotated with provenance information and may be represented in multiple languages. Built on top of the Wikidata Knowledge Graph, the Wikidata Query Service exposes this knowledge to the community and third-party developers through a scalable Web-based SPARQL endpoint, enabling queries such as "How

²[https://www.wikidata.org/](https://www.wikidata.org/)
did the population of Berlin develop over time", "Which countries are run by a female president", or "What are the most notable works displayed in the British Museum".

Wikidata is a cross-domain, general purpose knowledge graph, as such it is applicable to numerous application domains. Potential clients range from cultural heritage (such as museums) to the pharmaceutical industry. Requirements have been elicited from current customers as well as the larger Wikidata community, which discusses use cases and needs in a community process.

2.2.2 Common pain points and customer characteristics

A common characteristic is that the clients actually rely on open data to make their data more useful, or at least that they perceive that their own data is more useful if it is contextualized, enriched and interlinked with open data sources. In particular when relying in a variety of data sources, which may not actually be under the control of the enterprise, data quality is an important issue.

Another characteristic is further the support for complex information needs: Across all domains, information needs are complex in the structure of the queries need to express them, in the number of data sources required to answer them, and in the data modalities involved (data types, structured vs unstructured etc.) At the same time – while the information needs are complex – the interfaces to construct queries need to be usable by experts that understand their domain, but are not able to express queries in a formal query language.

2.2.3 Structured/Unstructured Data

Wikidata is tightly bound to its Wikimedia sister projects, in particular the Wikipedias. Many uses (and potentials) of Wikidata involve bringing together structured and unstructured data (e.g. Wikidata with the Wikipedia articles). Also for the enterprise data sources, there are both structured and unstructured source, such that typical use cases often require a combination along both dimensions: open data and enterprise data, as well as structured and unstructured.

2.2.4 Infrastructure at customers

Due to the important role of entity search and the requirement to bridge with unstructured data, rich keyword search is essential. Of particular interest is support for Elastic Search\(^3\), as this is the search engine of choice of the Wikimedia projects.

2.2.5 Outlook and Gain

Wikidata serves as an entity hub by providing entity descriptions for ca. 20 million entities covering a variety of domains. To this end, entity search is a predominant use case for search over Wikidata.

Wikidata itself has a language-neutral representation. Every item can carry and provide labels in many languages, which also feed the corresponding local Wikipedias. To this end, Wikidata is a useful resource for enabling multilingual search.

Furthermore, customers want to contextualize enterprise data. By linking enterprise data sources with Wikidata, that data can be effectively contextualized and enriched. Wikidata can thus serve as

\(^3\)https://www.elastic.co

Page 7
bridge to the open data world. Searches can then cover information needs that require the combination of enterprise data with open data.

Wikidata develops more and more towards a hub of identifiers for any kind if entities. By linking entities in enterprise knowledge graphs to Wikidata identifiers, not only the knowledge graph of Wikidata itself becomes queryable, but also other data sources that use Wikidata identifiers become accessible and amendable for search that involves previously disparate, isolated data sources. Links can be established by identifying identical instances (represented as e.g. owl:sameAs) or more relationships with related entities.

The ontology of Wikidata is special in the sense that it is very large and comprehensive (>200 thousand concepts), but also not very formal and prone to quality issues, as it is curated by the community rather than ontology engineers. These specifics of the Wikidata ontology need to be considered when developing search interfaces.

Additionally, the search over Wikidata is very diverse, e.g., involving the following:

- Entity search.
- Natural language queries (What are the largest cities with a female mayor?).
- Property paths of unknown length (e.g. ancestor relations, territorial structures, taxonomic structures, part of relationships).
- Discovery of links / paths between entities.
- Temporal and spatial data.

The potential gains are very much in line with the objectives, in particular with regards to the ability to support complex information needs through simple interfaces, the ability to federate over diverse sources, as well as the ability to extract and provide structured knowledge from unstructured content.

3 State Of The Art

The DIESEL project aims at improving enterprise search by a better understanding of the user’s information need and the integration of distributed data sources. This state of the art review focuses on (1) semantic based keyword search as well as hybrid question answering. Both methodologies are the most common information search strategies respectively will drive future innovation.

3.1 Keyword Search

This related work section is based on former publications of the project consortium, namely SINA [21] and SESSA [18].

We analyze semantic search approaches in five dimensions, i.e., i) input query format, ii) disambiguation, iii) expansion, iv) data distribution and v) query transformation. With respect to the first dimension, there are two common types of input query, i.e., natural language query and keyword query. There is a contradiction in usability studies of these two types of input queries. While [13] shows that users prefer using natural language queries to keywords, [19] presents that students prefer keyword query. Second dimension is using a disambiguation approach which selects the best interpretation of
the input query. Third dimension is query expansion like taking into account synonyms in order to improve retrieval performance. Fourth dimension is related to the number of the underlying knowledge bases, whether the search engine runs on either a single knowledge base or multiple interlinked knowledge bases. The last dimension refers on how to transform the input query to a formal query.

Semplore [28] is the first known hybrid search engine by IBM. It combines existing information retrieval index structures and functions to index RDF data as well as textual data. Semplore focuses on scalable algorithms and is evaluated on an early Question Answering over Linked Data (QALD) dataset.

Bhagdev et al. [1] describe an approach to hybrid search combining keyword searches, Semantic Web inferencing and querying. The proposed K-Search outperforms both keyword search and pure semantic search strategies. Additionally, a user study reveals the acceptance of the Hybrid Search paradigm by end users.

A personalized hybrid search implementing a hotel search service as use case is presented in [27]. By combining rule-based personal knowledge inference over subjective data, such as expensive locations, and reasoning, the personalized hybrid search has been proven to return a smaller amount of data thus resulting in more precise answers.

SINA [21] aims at answering a keyword question using different datasets. First, simultaneous disambiguation and segmentation is performed using Hidden Markov Models (HMM) and the Hyperlink-Induced Topic Search (HITS) algorithm. The resources found are used to construct an Incomplete Query Graph (IQG) consisting of disjoint sub-graphs. To build the federated SPARQL query that retrieves the results, the IQG’s are connected using a Minimum Spanning Tree approach inspired by Prim’s algorithm.

The work of Tran et al. [23] tackles the problem of keyword search over RDF data. More specifically, their work is concerned with mapping keywords to a list of ranked conjunctive queries, with a special focus on efficient inference of implied connections. To accomplish this, a top-k algorithm is proposed that computes the best query interpretations of the keyword query using bidirectional graph exploration. The interpretations are then scored and mapped to conjunctive queries.

All presented approaches fail to answer natural-language questions. Besides keyword-based search queries, some search engines already understand natural language questions. Question answering is more difficult than keyword-based searches since retrieval algorithms need to understand complex grammatical constructs.

The project partners analyse the search application perspective of their customer within the Use Case Section.

### 3.2 Hybrid Question Answering

Several question answering (QA) systems have been introduced and taken part in the QALD challenge, we briefly highlight some systems below.

One of the first systems was AquaLog [16], an ontology-driven QA system for the Semantic Web. AquaLog uses linguistic analysis to transform the input query to a set of query-triples. Then, these query triples are interpreted using lexical resources and the given ontology. The interpreted query-triples are sent to an inference engine to find the answer. One major drawback of AquaLog is that it is limited to one ontology at a time. To address this and other drawbacks of AquaLog, PowerAqua [15]

---

The project partners analyse the search application perspective of their customer within the Use Case Section.

### 3.2 Hybrid Question Answering

Several question answering (QA) systems have been introduced and taken part in the QALD challenge, we briefly highlight some systems below.

One of the first systems was AquaLog [16], an ontology-driven QA system for the Semantic Web. AquaLog uses linguistic analysis to transform the input query to a set of query-triples. Then, these query triples are interpreted using lexical resources and the given ontology. The interpreted query-triples are sent to an inference engine to find the answer. One major drawback of AquaLog is that it is limited to one ontology at a time. To address this and other drawbacks of AquaLog, PowerAqua [15]
was developed. PowerAqua can automatically combine information from multiple knowledge bases at runtime. The input is a natural language query and the output is a list of relevant entities. PowerAqua lacks a deep linguistic analysis and can not handle complex queries.

Schlaefer et al. [20] describe Ephyra, an open-source question answering system and its extension with factoid and list questions via semantic technologies. Using Wordnet as well as an answer type classifier to combine statistical, fuzzy models and previously developed, manually refined rules. The disadvantage of this system lies in the hand-coded answer type hierarchy.

Cimiano et al. [4] developed ORAKEL to work on structured knowledge bases. The system is capable of adjusting its natural language interface using a refinement process on unanswered questions. Using F-logic and SPARQL as transformation objects for natural language user queries it fails to make use of Semantic Web technologies such as entity disambiguation.

Damljanovic et al. [5] present FREyA to tackle ambiguity problems when using natural language interfaces. Many ontologies in the Semantic Web contain hard to map relations, e.g., questions starting with 'How long...' can be disambiguated to a time or a distance. By incorporating user feedback and syntactic analysis FREyA is able to learn the users query formulation preferences increasing the systems question answering precision.

Cabrio et al. [2] present a demo of QAKiS, an agnostic QA system grounded in ontology-relation matches. The relation matches are based on surface forms extracted from Wikipedia to enforce a wide variety of context matches, e.g., a relation birthplace(person, place) can be explicated by 'X was born in Y' or 'Y is the birthplace of X'. Unfortunately, QAKiS matches only one relation per query and moreover relies on basic heuristics which do not account for the variety of natural language in general.

CASIA [10] is the best-performing system on the QALD-3 benchmark dataset at the moment and relies on a three-step approach resembling AquaLog’s architecture. During the first step, the question type is determined and text triples are constructed from the dependency parse tree of the question sentence. In the second step, RDF resources which match phrases from the text triples are detected. In the final step, a SPARQL query is generated based on the question type and the RDF resources detected in the input question. CASIA achieves an F-score of 0.36 on the QALD-3 benchmark.

Pythia [25] is a question answering system that employs deep linguistic analysis. It can handle linguistically complex questions, but is highly dependent on a manually created lexicon. Therefore, it fails with datasets for which the lexicon was not designed.

Pythia was recently used as kernel for TBSL [24], a more flexible question-answering system that combines Pythia’s linguistic analysis and the BOA framework [9] for detecting properties to natural language patterns. Exploring schema from anchor points bound to input keywords is another approach discussed in [22]. Querying Linked datasets is addressed with the work mainly treat both the data and queries as bags of words [3, 26].

[11] presents a hybrid solution for querying linked datasets. It runs the input query against one particular dataset regarding the structure of data, then for candidate answers, it finds and ranks the linked entities from other datasets.

Treo [8] is a method for querying Linked Data that also relies on spreading activation. First, pivot entities in the query are identified. Then, from the dependency structure of the input sentence, Treo constructs a Partially Ordered Dependency Structure (PODS). The PODS is used to resolve the query in the spreading activation search step where semantic relatedness scores are used to rank candidates and subsequently spread activation. However, this approach is quite inefficient.

Several industry-driven QA-related projects have emerged over the last years. For example,
DeepQA of IBM Watson [7], which was able to win the Jeopardy! challenge against human experts. Further, KAIST’s Exobrain project aims to learn from large amounts of data while ensuring a natural interaction with end users. However, it is yet limited to Korean for the moment.

For further insights please refer to [6, 12, 17, 14] which present surveys on existing question answering approaches.

Overall, current research approaches tackle similar problems but do not covered the range of aspects to be considered within the DIESEL project.

4 Feasibility of DIESEL Engine

Here, we will look into each outlook of the respective use cases and will deduce required steps to confirm a feasible development here. However, this section will not go into great detail due to the ongoing user requirement elicitation, see later deliverable 1.3.

For the first use case the following points need to be considered:

- Integrating different data silos: DIESEL will use various technologies like FOX and SPARQLIFY to access non-RDF data sources.
- Enterprise knowledge graph as infobox: Providing an infobox based on the SemWeb2NL framework can be done. However, the framework needs to be extended to suffice the user requirements (see deliverable 1.3).
- Provide support for natural language processing in different languages: First, the University of Leipzig has rich experience in developing question answering systems and will thus be able to implement a natural language processing interface for English. However, porting the natural language system to other languages is not the focus of this project.
- Multi-lingual entity extraction: DIESEL can use the multilingual FOX framework to extract entities on unstructured texts as well as input queries.
- The federated search should support configurable data sources and allow to identify the origin of the data: DIESEL will use the federation engine QUETSAL and extend it to allow access to distributed data silos in RDF.

5 Conclusions

The goals, visions and gains of the DIESEL project presented in the short use case descriptions name various technologies. These technologies have been foreseen or slightly tackled with state-of-the-art approaches as pointed out in Section 2. However, the combination, scale and enterprise relevancy of the use cases demands novel approaches towards the implementation of the DIESEL search engine. Thus, we will further define the architecture (deliverable D1.2) and the exact specifications and requirements as well as benchmark data for the single workpackages (deliverable D1.3) to ensure a concise working goal.

Summarizing the preliminary approaches, the capabilities of the partners and the project goal, the DIESEL project is technically feasible.
References


