Agent-based Semantic Search at motoso.de

Nils Weber[1], Lars Braubach[2], Alexander Pokahr[2], and Winfried Lamersdorf[2]

[1] motoso.de GmbH & Co. KG
nilsweber@gmx.de

[2] Distributed Systems and Information Systems
Computer Science Department, University of Hamburg
{braubach | pokahr | lamersdorf}@informatik.uni-hamburg.de

Abstract. Searching for information in large rather unstructured real-world data sets is a difficult task, because the user expects immediate responses as well as high-quality search results. Today, existing search engines, like Google, apply a keyword-based search, which is handled by indexed-based lookup and subsequent ranking algorithms. This kind of search is able to deliver many search results in a short time, but fails to guarantee that only relevant data is presented. The main reason for the low search precision is the lack of understanding of the system for the original user intention of the search. In the system presented in this paper, the search problem is tackled within a closed domain, which allows semantic technologies to be used. Concretely, a multi-agent system architecture is presented, which is capable of interpreting a key-words based search for the car component domain. Based on domain specific ontologies the search is analyzed and directed towards the interpreted intentions. Consequently, the search precision is increased leading to a substantial improvement of the user search experience. The system is currently in beta state and it is planned to roll out the functionality in near future at the car component online market-place motoso.de.

1 Introduction

Searching for information is a difficult task and is considered to be an important skill for internet users. Despite this difficulty, currently well known search engines like Google are mainly based on keyword-oriented search requests. A search process often involves several rounds in which the user has to iteratively refine the query according to the preliminary gathered results. One reason for this way of searching is that the request style is easy to employ for the users. Nevertheless, the user intentions cannot be understood well, because only the occurrence of certain keywords decides about the result relevance and its inclusion in the result set. Understanding user intentions in general is very hard but becomes easier when the search topic is constrained in beforehand e.g. by considering a closed domain only. In this case semantic technologies, like ontologies and corresponding reasoning mechanisms, can be applied to analyze and comprehend the query.

Such an approach has the potential to combine the ease of use of a keyword-oriented search request with the semantic expressiveness of a metadata based query.

In this article, semantic search techniques are applied to the car domain at the German Internet marketplace motoso.de [11]. The platform is mainly specialized in new and used replacement and tuning components, tires and wheels as well as garage services for all types of vehicles. Currently, the total number of components is about 5.8 Million, whereby 85% is located in the passenger car category. In addition, roundabout 400.000 complete automobiles are registered in the database. The adverts are primarily placed by commercial customers and can contain a lot of detail information that is possibly of interest for potential buyers. In order to manage the adverts they are grouped into a tree-like structure with main and subcategories (e.g. 16 main categories for components with circa 700 branches of varying depth), which also exhibit links to the car types as well as their model and variant refinements. In addition to the base information containing a short and long description, price, state, origin, etc., a lot of category specific attributes with different types and allowed values can be specified. This allows a very precise (semi-structured) description, which should enable potential customers to find adequate offers. In this context, one serious problem is that descriptions are entered by different service providers in very different ways leading to variable data quality. This variable data quality causes non optimal search results, because the search is based on full text indices and might not take into account relevant but poorly described adverts.

In the next section the background and related works concerning semantic search is shortly summarized. Thereafter, in Section 3 the system architecture of the agent-based semantic search engine is presented. Its realization within motoso.de is described in Section 4 and the paper closes with a summary and an outlook on planned future work.

2 Analysis of the Search Problem

The motoso.de portal can be seen as representative for a wide range of internet- or intranet-based systems, i.e. the class of systems, which provide a search function as the main entry point to access its contents. These kinds of systems share a number of common properties:

- a common theme or domain for the whole portal,
- huge data sets,
- large diversities of data contents,
- heterogeneity with respect to data quality and data completeness (e.g. due to heterogeneous data sources),
- multiple search alternatives including a 'flat' keyword-based quick search as well as extended search and/or browsing capabilities based on domain specific metadata and categorizations.

The assets of such a kind of portal are the stored entries, where entries can be offered goods or services in marketplace like eBay (or motoso.de), but also e.g., publications in a scientific research portal. The main purpose of the portal's web interface is to provide an interested user access to those (and only those) entries that match its current interest. The different search alternatives provide the means for the user to express this interest to the system. As an example, two search alternatives are described in the following and discussed with respect to their advantages and disadvantages.

Quick Search: A single search phrase can be entered that (by default) leads to a keyword search matching those entries where all words appear in some part of the entry.

Advanced (or extended) Search The user is offered detailed control over which parts of the entries data should be searched (e.g. title, text, specific attributes). Moreover, queries a not entered as 'flat' text but follow domain specific representations (e.g. two number fields for entering a price range, a drop down list or a set of check boxes for choosing among predefined categories).

Quick search is the kind of search that most search engines like Google offer and that is therefore well known by virtually every internet user. Among the biggest advantages of the quick search is its intuitivity and ease of use. The disadvantage is the generally poor result quality of purely keyword-based search [1,10]. Therefore, most search engines improve the *perceived* result quality by applying stemming, and removing stop words from the search phrase and by sorting the results using complex ranking algorithms. Nevertheless, this approach can not incorporate domain knowledge for interpreting search requests.

The advanced search approach requires that knowledge about the domain of the portal is explicitly incorporated into the system (for providing differentiated search options), but also assumes that the user is knowledgeable in the domain itself (to make use of the advanced search options). The disadvantage of this approach is therefore the complicated query formulation that might deter inexperienced users. On the other hand, experienced users benefit from the ability of formulating very precise queries with respect to the relevant properties of the considered domain.

The two described search alternatives form extremes with respect to the trade-off between usability and result quality. Recently, semantic approaches have been introduced that try to combine both advantages. The general idea of these approaches is to offer a simple free form query, but use semantic technologies for interpreting the query and thus improving search results. An overview of several of these approaches is given in the rest of this section. The overview is coarsely divided into semantics for broad-based search engines vs. semantically enriched vertical, i.e. domain-specific, engines.

Introducing semantics is typically much more ambitious for broad-based search engines, because they cannot easily draw initial implications from the query. An elegant solution for this problem are semantic web search engines that are capable of performing an attribute-based retrieval process on typical semantic web resources like ontology A-boxes and RDF resources. These engines can directly make use of semantic technologies like ontological reasoners but currently can operate on a small database only. Regrettably, they are subject to the basic chicken and egg problem of the semantic web, i.e. people don't see the reason to be one of the early adopters for meta tagging their sites because the merits of this overhead remain unclear as long as no critical mass has been achieved [5]. Examples of this category are the SHOE[3] and Swoogle[4] [4] search engines. Other approaches like Powerset[5] and Cognition[6] (e.g. compared in [7]) try to semantically interpret the query using natural language processing (NLP) techniques. Basically, these approaches translate the user query into a canonical query over the database of indexed documents [6]. Some NLP search approaches also classify the interpreted query according to several predefined areas of expertise. In a second step these approaches can then use domain-specific for interpreting the categorized query similar to vertical search engines (see next paragraph). An example of this approach is the Hakia search portal[7], which supports queries in multiple languages and covers already a diversity of different domains.

Vertical search engines are developed for specific domains and include domain knowledge for retrieving optimized search results. Classical vertical search engines operate similar to standard search engines and use indices for looking up search results. These engines have clear advantages in the crawling and indexing process, because the spiders only have to be sensitive for web pages that contain keywords from the considered domain and the index is also constrained by the domain vocabulary [12]. The quality of search can be further enhanced when vertical search engines are enriched with semantic technologies. An ontology for the target domain can be developed by experts from the specific area. The ontology can then be used to understand terms and possibly also combinations of terms in the query leading to a (more or less vague) interpretation of the user intention. As described above, for this interpretation, sophisticated NLP algorithms can be applied, though, at least some indication exist [3] that a vague but simple ontology-based interpretation of a query can for some application cases be more appropriate than a thorough NLP analysis. One example is the vertical semantic search engine UpTake[8] for holiday planning. It tries to extract the general intention of the trip planning such as "family holidays" versus a "romantic trip". On basis of this broad categorization the search is directed to the right data by knowing ontologically, what makes up the interpreted type of trip. This information can then be used to rank the results accordingly.

The system described in this paper uses ontology-based reasoning for improving the search in a domain-specific portal. It therefore follows a similar approach like UpTake. The main reasons for this choice are its relative simplicity and effectiveness compared to other (e.g. NLP) approaches.

---

[3] `http://www.cs.umd.edu/projects/plus/SHOE/search/`
[4] `http://swoogle.umbc.edu/`
[5] `http://www.powerset.com/`
[6] `http://www.cognition.com/`
[7] `http://www.hakia.com/`
[8] `http://www.uptake.com/`

3 Semantic Search System Architecture

In this section the general architecture of the semantic search system and its implementation will be introduced. Its main purpose is to transform an incoming quick search query to an advanced query with explicit structure via interpretation of the formerly unstructured search phrase. In addition, the system shall also be capable of improving its behavior constantly by incorporating user feedback on the search quality.

3.1 System Design

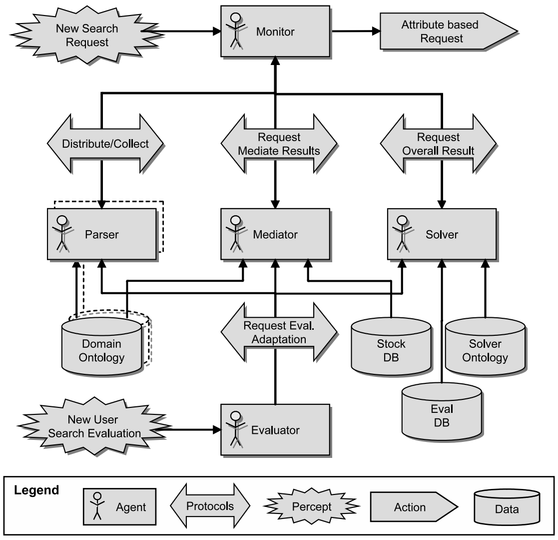

Fig. 1. ASQP system architecture

The main architecture of the ASQP (agent-based semantic query processing) system is depicted as Prometheus [8] system overview diagram in Fig. 1. An incoming quick search request is directed towards the *monitor* component that is responsible for its complete processing and therefore can be considered as the control unit of the system. In a first step the control unit decomposes the request into a list of single words and word chunks up to a predefined length and distributes it together with the original request to all active *parsers*. Each parser analyzes the request according to its domain knowledge, whereby the system can provide an arbitrary number of differently specialized parsers. Hence, a parser tries to identify the semantics of the single tokens and small token chunks of the request by operating on a specific ontology or taxonomy and sends its interpretation back to the monitor. The monitor collects the answers and in case the last answer arrived or a defined timeout occurs preevaluates the results. If there are conflicting interpretations of words or chunks from different parsers the results are forwarded to a *mediator*. The mediator has access to all ontologies, previous evaluations and probabilistic observation data and decides for each conflicting word or token which decision to follow. The arbitrated results are then forwarded to a *solver,* which has the purpose to generate the final solution. Therefore, it mainly resorts to an ontology of search pattern behavior, in which search intentions are stored for groups of semantic components. The search intention together with the semantically analyzed request is then handed over to the monitor, which redirects the request as attribute-based search to other established components outside of the ASQP system.

The incorporation of user feedback in the system is realized via a dedicated *evaluator* component. This component is fed with user evaluation data concerning single requests and calculates evaluations for the parsers, mediator and solver. These evaluations are stored in a specific evaluation database, and allow estimating the system's standard of performance. Furthermore, the specific evaluations are also forwarded to the corresponding components, which in this way get a chance to adapt their behavior accordingly.

3.2 System Implementation

The ASQP architecture has been implemented as multi-agent system using Jadex 0.96 [2]. Each of the roles, as defined in the design, has been mapped to a separate BDI agent type. The agents operate on the RDF and OWL ontologies using the semantic web framework Jena2[9]. In order to meet the performance criteria for a real-time search engine, most ontologies have been indexed using LARQ (Lucene + A SPARQL Processor for Jena). This allows a quick full text search on the ontologies with LARQ-extended SPARQL [13] queries. To reduce the number of matches in beforehand of the processing, additionally a Lucene-score can be used. Reasoning capabilities are currently only used in the solver agent. This agent instantiates an inference ontology consisting of the parser results and applies (currently quite simple) inference rules to deduce the user intention. Finally, the integration of the system with the existing search infrastructure and with the user interface had to be considered. The connection to the existing system infrastructure is minimal and is restricted to a search call on the traditional attribute-based search service. On the other hand, the user interacts via the browser with the system the Jadex webbridge framework [9] could used to simplify the delegation of user requests via Java servlets to the agent tier and back. Finally, the user interface of the portal itself was changed slightly. The user now has the possibility to explicitly turn on/off the semantic search facility. In addition, a new feedback dialog is generated, in which a user can state if the extracted intention was right or not.

4 Realization within motoso.de

To incorporate the ASQP system into the motoso.de portal, two distinct tasks had to be performed.[10] For the system to provide meaningful results with respect

[9] `http://jena.sourceforge.net`

[10] A test version of the motoso.de semantic search is available at: `http://semanticmotoso.mine.nu/`

to the automotive domain, specific ontologies and inference rules had to be provided. Additionally, different modes of operation were implemented, providing different strategies, how the system interacts with a user.

4.1 Domain-Specific Ontologies and Inference Rules

Domain knowledge for the motoso.de portal is reflected in three separate ontologies, which are used by the respective parser agents (cf. 3.1). The purpose of each of the parser agents is to assign meaning to single keywords or phrases. Each parser only considers a specific ontology and therefore only assigns meaning to those keywords that match some concept of this ontology. For the interpretation of the ontological concepts, all domain-specific ontologies refer to a common general-purpose ontology containing basic concepts and properties such as 'isPartOf' and 'equivalent'. The domain-specific ontologies, described below, all contain type as well as instance data.

Car Brands and Model Series Ontology On the type level, this ontology introduces broad concepts like 'Manufacturer' and 'VehicleType', but also very detailed conceptual structures for describing properties of model variants, etc. The instance level contains data about a wide range of car types and is taken from the corresponding German federal authority (Kraftfahrtbundesamt).

PARTS Ontology PARTS is an acronym for (replacement) parts, accessories, rims/tires, tuning parts, and services. The PARTS ontology therefore contains type and instance level data about specific parts or services. This ontology has been created by domain experts in a manual process by analyzing the existing offers. Special care has been taken to extract all the synonyms and abbreviations used for describing the same concept (e.g. 'window lifter' vs. 'window winder'), as these are important for improving the search result quality.

Geographic Locations Ontology This ontology contains information about German cities, districts, states, and regions as well as German postal codes.

To exemplify the usage of the domain ontologies, consider the search queries 'audi 100' and 'audi 22527'. Based on the information in the car brands ontology, the first query can be matched as a complete phrase to a specific car type (Audi 100). For the second query, no match for the phrase exists, but the separate keywords can be matched, i.e. 'audi' represents a manufacturer (car brands ontology) and '22527' is a postal code belonging to a district of Hamburg (geographic ontology).

As described in Section 3.1, the combined (and possibly conflicting) meanings provided by the three parsers are collected and interpreted by the solver agent. The solver agent uses domain specific reasoning rules that operate on the results from the parsers. The reasoning rules are responsible for assessing the query intention (e.g. is the search geared towards parts or a complete vehicle). The rules are manually derived by experts and represent their experience, how unstructured queries should be matched to the domain-specific attributes. In addition, the solver can use evaluation data of previous requests to resolve conflicts due to incompatible semantic interpretations from the different parsers.

4.2 Modes of operation

The ASQP system takes a user query, which is then augmented with semantic annotations. The system itself makes no assumptions, how these annotations should be reflected in a portals web presence. In the prototype developed at motoso.de, the user can choose from three different options of using the ASQP system:

Show In this mode, the semantic interpretation of the users search phrase is presented to the user without altering the search itself. This mode is helpful for inexperienced users to learn how their queries would be interpreted. Moreover, this mode is used during the evaluation and training phase of the system (see below).

Suggest Based on the semantic interpretation of the search phrase, this mode will offer alternatives these (e.g. synonymous or related concepts) in addition to the originally identified search terms.

Do This mode transforms the original user query into a new attribute-based search, that reflects the predicted user intention.

Ideally, the system would work only in 'do' mode, once it is productive, i.e. constantly providing an improved search result quality without bothering the user about the complex semantic internals that operate behind the scenes. Yet, as the dynamic content of the portal represents a moving target and the ontologies and reasoning rules are hand-crafted, it needs a certain amount of training for the system to produce stable results. Therefore a fourth mode ('eval') is implemented, that makes use of the evaluation feature of ASQP. In 'eval' mode, the semantic interpretation of an issued query is presented to the user, who can then rate each of the recognized intentions and concepts as being (in)correctly identified (see Fig. 2). The evaluation result is fed back into the ASQP system, which stores all evaluations to improve future query analysis (cf. Section 3.2).

5 Summary and Outlook

In this paper a general architecture and a concrete, domain-specific implementation of an agent-based semantic search system are presented. It is argued that semantically enriched search technology can fill a gap between an easy to use but limited 'quick search' and a powerful but complicated attribute-based 'advanced search'. An overview of existing approaches is given and related to the semantic approach used in this paper.

The proposed ASQP (agent-based semantic query processing) architecture combines semantic technology with a multi-agent system collaborative problem solving approach. The architecture is flexible and extensible in the sense that arbitrary agents, which are knowledgeable in a specific area, can easily be added and removed at any time. The different agents are thus able to interpret different portions of a user query and ultimately allow deriving user intentions from the query text. The architecture was developed in the context of a concrete application domain (the motoso.de portal) and also realized in this context. Different

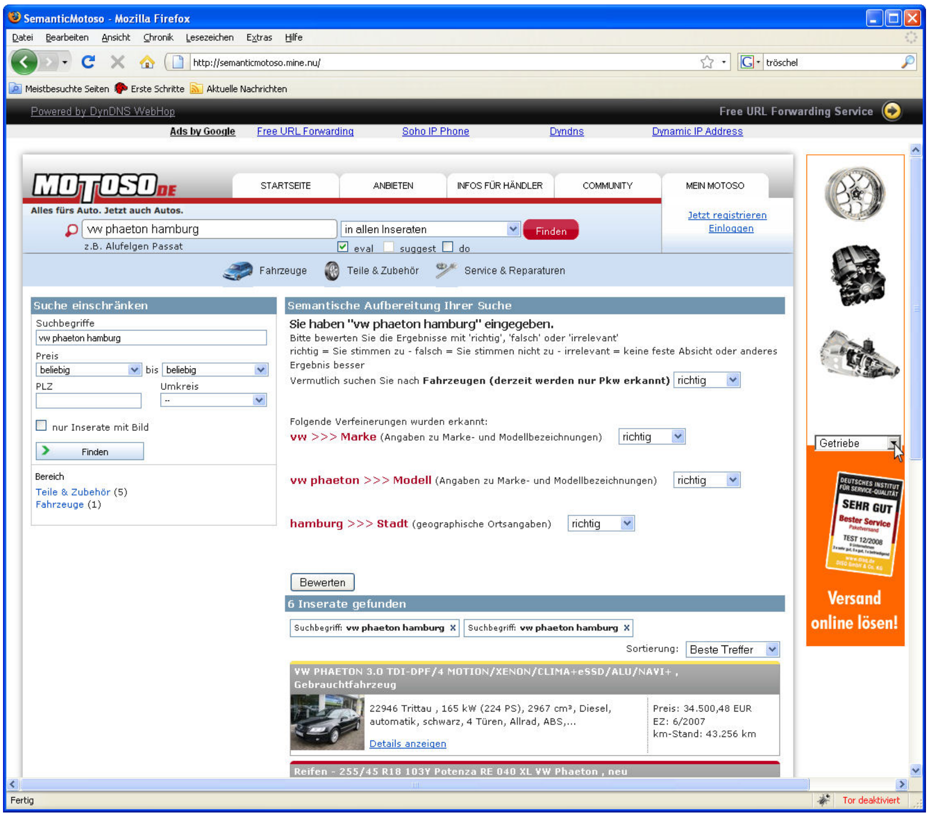

Fig. 2. Screenshot of the system in 'eval' mode

ontologies relevant for the car domain were devised and provided to separate agents. The results of the semantic query interpretation can be used in the portal to provide suggestions to users or directly transform the search request in accordance to the assumed user intention.

Currently, the system is running as a prototype and is used for internal evaluations of the technology at motoso.de. The evaluation is also performed for training of the system with respect to the domain in order to improve the overall query interpretation accuracy. On a technical level, the system runs stable with average response times below one second. If the ongoing evaluation ultimately confirms the initially promising results it is planned to integrate the system into the public motoso.de search interface.

References

1. D. Blair and M. Maron. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, 28(3):289–299, 1985.
2. L. Braubach, A. Pokahr, and W. Lamersdorf. Jadex: A BDI Agent System Combining Middleware and Reasoning. In R. Unland, M. Calisti, and M. Klusch, editors, *Software Agent-Based Applications, Platforms and Development Kits*, pages 143–168. Birkhäuser, 2005.

3. J. Catone. Semantic travel search engine uptake launches. *ReadWriteWeb*, 2008. http://www.readwriteweb.com/archives/semantic_travel_search_uptake.php.
4. L. Ding, T. Finin, A. Joshi, R. Pan, S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. In D. Grossman, L. Gravano, C. Zhai, O. Herzog, and D. Evans, editors, *Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, Washington, DC, USA, November 8-13, 2004*, pages 652–659. ACM, 2004.
5. J. Hendler. Web 3.0: Chicken farms on the semantic web. *IEEE Computer*, 41(1):106–108, Jan. 2008.
6. A. Iskold. Semantic search: The myth and reality. *ReadWriteWeb*, 2008. http://www.readwriteweb.com/archives/semantic_search_the_myth_and_reality.php.
7. N. Karandikar. Powerset vs. cognition: A semantic search shoot-out. *GigaOM Tech News and Views*, 2008. http://gigaom.com/2008/06/07/powerset-vs-cognition-a-semantic-search-shoot-out/.
8. L. Padgham and M. Winikoff. *Developing Intelligent Agent Systems: A Practical Guide*. John Wiley & Sons, 2004.
9. Alexander Pokahr and Lars Braubach. The webbridge framework for building web-based agent applications. In M. Dastani, A. El Fallah Segrouchni, J. Leite, and P. Torroni, editors, *First International Workshop on LAnguages, methodologies and Development tools for multi-agent systemS (LADS 2007)*, pages 173–190. Springer Verlag, 2008.
10. S.M. Shafi and R.A. Rather. Precision and recall of five search engines for retrieval of scholarly information in the field of biotechnology. *Webology*, 2(2). http://www.webology.ir/2005/v2n2/a12.html.
11. N. Weber. Ontologien zur multiagentengestützten Suche - Einsatz in der Automobildomäne unter Verwendung von Jadex. Diplomarbeit, Distributed Systems and Information Systems Group, Computer Science Department, University of Hamburg, 2009. (in German).
12. Wikipedia. Vertical search — wikipedia, the free encyclopedia, 2009. [Online; accessed 6-May-2009].
13. World Wide Web Consortium (W3C). *SPARQL Query Language for RDF*, January 2008.