

A Semantic Web-Based Approach for Building Personalized News Services

Flavius Frasincar, Jethro Borsje, and Leonard Levering
Erasmus University Rotterdam
PO Box 1738
NL-3000 DR Rotterdam
the Netherlands

Abstract. This article proposes Hermes, a Semantic Web-based framework for building personalized news services. It makes use of ontologies for knowledge representation, natural language processing techniques for semantic text analysis, and semantic query languages for specifying wanted information. Hermes is supported by an implementation of the framework, the Hermes News Portal, a tool which allows users to have a personalized online access to news items. The Hermes framework and its associated implementation aim at advancing the state-of-the-art of semantic approaches for personalized news services by employing Semantic Web standards, exploiting domain information, using a word sense disambiguation procedure, and being able to express temporal constraints for the desired news items.

Keywords. Personalization, News Services, Semantic Web, Ontologies, Natural Language Processing

INTRODUCTION

The simplicity, availability, reachability, and reduced exploitation costs have made the Web one of the most common platforms for information publishing and dissemination. This is particularly true for news agencies that use Web technologies to present emerging news regarding different types of events as for example business, cultural, sport, and weather events. Most of this information is published as unstructured text that is made available to a general audience by means of Web pages.

The heterogeneity of the Web audience and the diversity of the published information asks for more refined ways of delivering information that would enable users to access news items that interest them. For this purpose the Really Simple Syndication (RSS) (Winer, 2003) standard was developed that publishes information in a semi-structured format that supports machine processing. This format is based on metadata that (1) associates news items with channels (feeds) that have properties like categories (e.g., business, sport, politics, etc.), title, publication date, etc., and (2) describes news items by means of their properties as categories (e.g., online business, business system, Internet marketing, etc.), release time, title, abstract, link to the original published information, etc.

Most of the annotations supported by the RSS feeds are coarse-grained in the sense that they fail to identify the different topics addressed in a certain news item. Also, the current annotations are only partially processable by machines as the tags do not have

unique semantic meaning associated to them and thus have different interpretations. Being able to understand the semantic content of a news item would enable a fine-grained categorization of this information, thus better supporting the users (casual users, media analysts, stock brokers, etc.) information needs.

In order to make the Web data not only machine readable but also machine understandable the World Wide Web Consortium proposes the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001), a sequence of technologies that allow for self-describing content. On the Semantic Web metadata is defined using semantic information usually captured in ontologies. Some of the most popular formats to describe ontologies on the Semantic Web are RDF(S) (Klyne & Carroll, 2004) (Brickley & Guha, 2004) and OWL (Bechhofer et al., 2004).

A special class of users who make daily use of (emerging) news is that of stock brokers. Because news messages may have a strong impact on stock prices, stock brokers need to monitor these messages carefully. Due to the large amounts of news information published on a daily basis, the manual task of retrieving the most interesting news items with respect to a given portfolio is a challenging one. Existing approaches such as Google Finance or Yahoo! Finance are developed to meet these personalization needs by supporting automatic news filtering on the Web.

Current approaches to news filtering are able to retrieve only the news that explicitly mention the companies involved, failing to deliver indirect information which is also deemed relevant for the considered portfolio. For example, for a portfolio based on Google shares, such systems fail to deliver news items related to competitors of Google, such as Yahoo! or Microsoft, which might have an indirect influence on the share price of Google. Exploiting the semantic contextual information related to companies such as its competitors, CEO's, alliances, products, etc., enables a more comprehensive overview of relevant news with respect to a certain portfolio.

Another limitation of existing news filtering systems is their inability to cope with delivering news items satisfying temporal constraints. The time aspect is of utmost importance when, for example, one considers the fact that news items usually have an immediate impact on stock prices, or when one desires to do a historical analysis of past news and stock price evolutions. Being able to exploit the timestamps associated to news items enables retrieving only news that obey user-determined time-related constraints.

In this paper we propose the Hermes framework, a semantic-based approach for retrieving news items related, directly or indirectly, to the concepts of interests from a domain ontology. In addition these news items might need to satisfy temporal constraints. For illustration purposes we focus here on the NASDAQ stock market domain (Kandel & Marx, 1997), but the genericity of our approach makes it applicable also to other domains, as, e.g., tourism or scientific domains. The Hermes News Portal (HNP) is an implementation of the Hermes framework, which allows the user to specify queries for the concepts of interest and temporal constraints, and retrieve the corresponding news items.

For HNP we make use of Semantic Web technologies like OWL (Bechhofer et al., 2004) for formally defining the semantics of the concepts of interest in the ontology.

We employ natural language processing (NLP) technologies as, e.g., lexical analysis, gazetteering, word sense disambiguation, etc., for indexing news items based on ontology terms. The most popular Semantic Web query language SPARQL (Prud'hommeaux & Seaborne, 2008) is used for expressing queries using the previously identified concepts. In order to simplify the representation of temporal constraints we propose time-related extensions to SPARQL. HNP is a generic platform that could easily be applied to other domains than the financial one.

The structure of the paper is defined as follows. The first section discusses related approaches for personalized news services. The second section presents the Hermes framework identifying the proposed methodological steps. The third section describes HNP, an implementation of the proposed framework. The last section concludes the paper and discusses future work.

RELATED WORK

Among the methods that aim at personalizing news information we distinguish two types: non-semantic approaches and semantic approaches. In the followings we will present short descriptions of two non-semantic methods: Server for Adaptive News and YourNews, and two semantic methods: MyPlanet and SemNews. For each presented method we give the differences compared to our approach and at the end of this section we highlight the main contributions of the Hermes framework.

Server for Adaptive News (SeAN) (Ardissono, Console, & Torre, 2001) enables a personalized access to news servers on the Web. The generated views are composed of sections, as in newspapers, on which customized news items are embedded. The news items are viewed as complex entities in which attributes define different components, e.g., title, abstract, text, photos, videos, commentaries, etc. The system employs a user model initialized using orthogonal stereotypes (interests, domain expertise, cognitive characteristics, and life styles) for which the user is asked to provide input and is further updated using rules that exploit the user behavior with the application. Taking into account the user model, the system builds a presentation based on relevant news items, each news item being shown at an appropriate level-of-detail (based on the user model). Differently than SeAN, our framework uses standard Semantic Web technologies for representing knowledge and employs NLP techniques for automatic annotation of news items, instead of using a manual approach.

YourNews (Ahn, Brusilovsky, Grady, He, & Syn, 2007) proposes an open and editable user model for personalizing news items. The user model is open in the sense that users can view the list of keywords stored in the individual profiles. The user model is also editable as it allows users to add/delete keywords from their associated profiles. As an additional feature which also contributes to the transparency and control over adaptation, YourNews shows the key terms present in news items. The representation of news items is given by weighted vectors of terms (Salton, 1971), where the weights are computed using TF-IDF (Salton & McGill, 1983). The visited news items are used for building a weighted term vector which is the user model. The similarity between a news item and the user model is defined by the cosine metric between their associated vectors. This measure allows the system to recommend news items that are considered relevant for the user. Despite the users' interest to view and edit their profiles, there is a decrease in performance (e.g., precision, recall, etc.) for

recommended items compared to the same system using a closed user model (where the user is not able to view/edit the user model). While YourNews uses a keyword-based approach for modeling news items and user interests, Hermes employs a semantic approach based on ontology concepts for modeling similar aspects.

MyPlanet (Kalfoglou, Domingue, Motta, Vargas-Vera, & Shum, 2001) aims at providing users with news items relevant for their topics of interest. MyPlanet is an extension of PlanetOnto, an integrated suite of tools used to create, deliver, and query internal newsletters of the Knowledge Media Institute (KMi). Similar to our approach an ontology is used for classifying news items and allowing the user select his topics of interest. Nevertheless, the classification process is based on the heuristics of cue phrases attached to ontology concepts, while we have a more systematic approach to classification by employing NLP techniques (e.g., exploiting the WordNet term synonyms, performing word sense disambiguation, etc.) that improve classification results. In addition, our implementation is based on the standard ontology language OWL instead of the specific ontology language, OCML (Motta, 1999), used in myPlanet. We also did choose to present the ontology as a graph instead of a tree as it allows the user to have a more comprehensive overview of the ontology structure.

SemNews (Java, Finin, & Nirenburg, 2006) proposes a framework for understanding and querying news items. As in Hermes, the monitored news are made available by RSS feeds. The news items are analyzed by OntoSem (Nirenburg & Raskin, 2001), SemNews' natural language processing engine. OntoSem converts the textual representation of news into Text Meaning Representation (TMR), a specific format for knowledge representation. The TMR descriptions are subsequently converted to OWL and published on the Web. The OWL news representation can be used for querying using RDQL (Seaborne, 2004), one of the precursors of the SPARQL query language. Differently than SemNews, Hermes uses a semantic lexicon (e.g., WordNet) for performing word sense disambiguation, and allows for a more intuitive way of building queries by letting the user make his selections in a graphical way.

The contributions that Hermes brings to building personalized news services compared to existing approaches are fivefold. First, Hermes makes a strict distinction between the framework (Hermes framework) and its implementation (HNP), allowing for possible different technologies (as these evolve) to be used with the same framework. Second, Hermes uses an advanced NLP methodology (e.g., tokenization, part-of-speech tagging, word sense disambiguation, etc.) for news understanding employing a semantic lexicon (e.g., WordNet). Third, the implementation is based on the most up-to-date Semantic Web standards (OWL and SPARQL). Fourth, we allow news querying using temporal constraints by providing temporal extensions to SPARQL. Fifth, and the last contribution, the user is provided with a graphical query interface to specify the concepts of interest in a direct (using concept selections) or indirect manner (using relationship selections).

HERMES FRAMEWORK

The Hermes framework proposes a sequence of steps to be followed in order to build a personalized news service. The input for the constructed system comprises RSS news feeds and the output consists of news items fulfilling user needs. The Hermes framework is centered around a domain ontology which is used for indexing news

items and helping the user formulate queries based on his concepts of interest. In addition, the user can specify temporal constraints that news items need to satisfy. The resulting news items are sorted based on their relevance for the user queries.

For illustrative purposes we chose a financial domain example, i.e., a personalized news service which can help the stock brokers in their daily decisions. More precisely we opted for portfolios based on stocks of NASDAQ companies. For this purpose we developed a domain ontology, which captures companies, products, competitors, CEO's, etc. In addition we have developed a news ontology able to store news items and their metadata such as title, abstract, time stamp, etc.

The domain ontologies are developed by domain experts. The process of developing the ontology is an incremental middle-out approach. First the most salient concepts are defined and then these are refined using generalization/specialization towards the top/bottom of the ontology. As the news information can contain additional concepts not a priori known, the ontology needs to be regularly maintained by the domain experts. We validated our domain ontology using the OntoClean methodology (Guarino & Welty, 2002).

As news items might come from different RSS feeds, it is possible that the same news item has been published through different channels. After aggregating the news items one needs to remove the duplicate ones. In order to speed-up this process we employed the heuristics to use only the title for identifying news items, even so we acknowledge that in few cases news items might have the same title and still have different content, i.e., they represent different news items.

The architecture of the Hermes framework is described in Figure 1 and is composed of three main steps: *news classification*, *news querying*, and *results presentation*. *News classification* is responsible for indexing the news items based on ontology concepts. *News querying* consists of two substeps: *query formulation*, i.e., helping the user build the query that expresses the items of interest, and *query execution*, i.e., computing the results of query evaluation. In the last step, *results presentation* the computed news items are presented based on their relevance to the user interests.

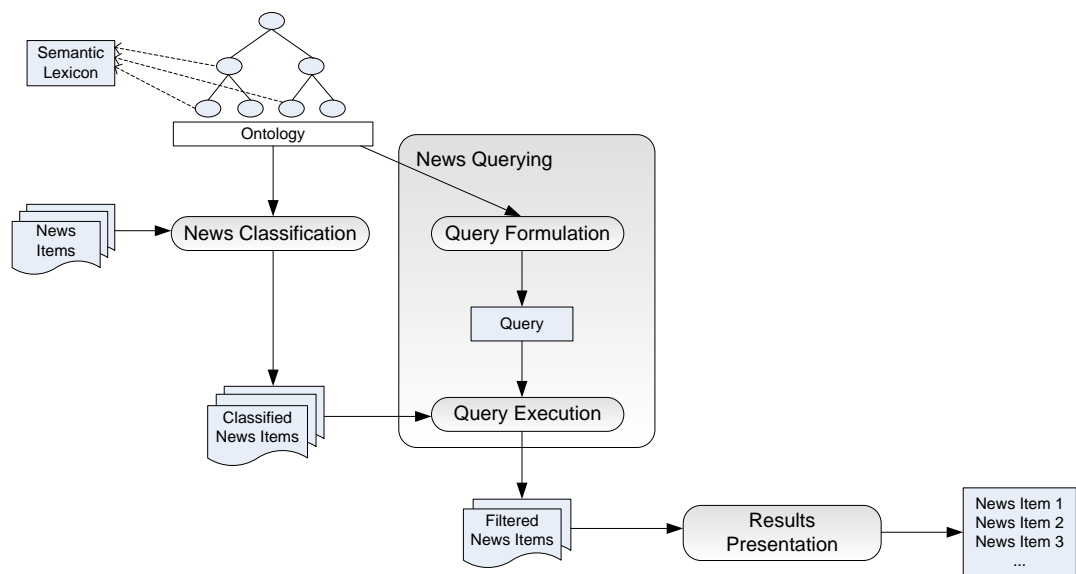


Figure 1 Hermes architecture

NEWS CLASSIFICATION

The ontology concepts used for news items classification are classes and individuals from the domain ontology. Concepts are linked to synsets, i.e., sets of synonyms, from a semantic lexicon, which identify their unique meaning. The synonyms stored in a synset in the semantic lexicon are used as lexical representations of the associated concept. In this case a lexical representation has a sense associated with it, i.e., the one given by the corresponding synset. As the ontology is specific to a domain, while the semantic lexicon is domain independent, we associate additional domain specific lexical representations to the concepts in our ontology. The lexical representations are composed only of word lemmas (the canonical word form appearing in dictionaries).

In addition, for classes without subclasses and individuals we decided to consider the hyponyms associated to their corresponding semantic lexicon synset. For these classes and individuals, the domain expert, who devised the ontology, is probably not interested in more refined definitions of these. However, the lexical representations of these concepts can be enlarged by considering also the corresponding hyponyms synsets from the semantic lexicon.

The classification approach is *ontology-centric*, in the sense that the ontology concepts are loaded one at-a-time and their lexical representations are matched against the news items. This approach is opposed to a *news items-centric* one where the words in the news items are matched against the lexical representations of the concepts from the ontology. We opted for an ontology-centric approach in order to speed-up the classification process as in this case we need to traverse the ontology only once. The number of concepts in the ontology is considerably larger than the number of words in the news items.

First the tokenization, sentence splitting, part-of-speech tagging, and morphological analysis are performed. The tokenization precedes sentence splitting as sentence splitting needs the identification of the punctuation signs from tokenization (as “.”, “;”, etc.). Morphological analysis follows part-of-speech tagging because the lemma of a word depends on its part-of-speech tag. For example “reading” as a verb has lemma “read” but as noun it has the lemma “reading”. In this way, all words in a news item are reduced to their canonical form, a form shared also by the lexical representations of concepts stored in the domain ontology. For lexical representation identification we use the maximal group of words (compound words) found in sequence in a news item that stand for a concept’s lexical representation. For example “European Central Bank” would be identified as a compound word representing the ECB concept, e.g., a longer match supersedes a shorter match.

Each time a lexical representation of a concept is matched a *word sense disambiguation* procedure takes place. As the same lexical representation can belong to different concepts (present or not in the ontology) this procedure checks if the match indeed corresponds to the meaning of the found ontology concept. If the check is positive a *hit* is stored in the ontology, i.e., a link between the news item and the corresponding concept is defined. The hit also stores the found lexical representation, as classification evidence.

For word sense disambiguation we use a variant of the SSI algorithm (Navigli & Velardi, 2005). In this process we also consider lexical representations for concepts that are not stored in the ontology but are present in the semantic lexicon as these are relevant when computing the sense of a found lexical representation from the ontology. These lexical representations help in better determining the context of a sentence and thus computing the sense of an ontology-based lexical representation.

The algorithm determines, per news sentence, the sense of a lexical representation (*lex*) by computing the sum of the distances between one of the senses of the considered lexical representation (s_j) and the senses of the previously disambiguated lexical representations from the context sentence (sc_i). The sense corresponding to the smallest sum is the chosen one (*selectedSense*). The algorithm starts with monosemous lexical representations (i.e., lexical representations which correspond to only one concept) and in case that such representations do not exist a guess is made by picking the most common sense for one of the found lexical representations. These senses are added to the context (I) of the sentence. For each remaining polysemous lexical representations two steps take place: a disambiguation step to find the correct sense and an insertion step which adds the newly computed sense to the context. In this way the context gets enlarged with new senses that model the meaning of the sentence. Formula (1) specifies what sense is selected for each lexical representation.

$$selectedSense(lex) = \arg \min_{s_j \in senses(lex)} \sum_{sc_i \in I} d(s_j, sc_i) \quad (1)$$

The distance between senses is defined as being inverse proportional with the length of the shortest path between the corresponding synsets ($sense_i$ and $sense_j$) in the semantic lexicon graph. The graph is based on the hypo/hypernyms relationships stored in the semantic lexicon. Path lengths larger than a predefined constant (e.g., 4 or 5) are not used (for these cases the distance is considered infinite). In this way we employ only semantically close concepts that truly help in the disambiguation procedure and also improve the speed of the process by not using arbitrarily large paths. Formula (2) shows how to compute the distance between two senses.

$$d(sense_i, sense_j) = length(shortestPath(synset(sense_i), synset(sense_j))) \quad (2)$$

As the distances between synsets are not changing, one can pre-compute these, thus further reducing the time needed for the disambiguation step. The Hermes framework can be used with other methods for computing the similarity between concepts as for example string metrics (e.g., Levenshtein, Editex, etc.) or lexical co-occurrences in a corpus (e.g., pairwise mutual information, Google distance, etc.). Nevertheless, many of these methods are less precise than the graph-based method used here, as they do not take in account senses, comparing only the lexical representations of concepts.

NEWS QUERYING

The user expresses the topics of interests by posing queries using concepts from the domain ontology. In addition the user can express constraints that the timestamps associated to news items need to satisfy. The news querying step consists of two substeps: query formulation, i.e., supporting the user to build queries, and query

execution, i.e., computing the results of query evaluation. Below we present in detail each of these substeps.

Query Formulation

In order to assist the query construction process we present to the user the ontology graph. We decided to show a graph-based representation of the ontology instead of a tree-based representation, as it gives more insight in the overall structure of our domain. For example a graph representation captures more relationship types instead of a singular relationship type, often the subsumption relationship, from a tree-based representation. The user needs to understand the different relationship types in order to be able to build his query.

By using the ontology graph the user can select the *direct concepts of interest*. In addition he is able to specify concepts of interest using a keyword search facility through the graph. For this purpose the input keyword is checked for possible inclusion in the lexical representations of concepts. If such inclusions are found the corresponding concepts are being returned as possible direct concepts of interest. It is the task of the user to accept these as direct concepts of interest or to reject them.

One of the important functionalities of the Hermes approach is that it allows for the selection of concepts indirectly linked to the selected ones, concepts which are not a priori known to the user. We call these concepts *indirect concepts of interest*. For the selection of the indirect concepts of interest the user can state the type of the relationships that links the direct concepts of interest to the indirect concepts of interests or leave this undefined in which case all relationship types are being considered.

Suppose that the user has selected the direct concept of interest `Google` from the NASDAQ domain ontology. The user also specifies that he is interested in news related to the competitors of Google by selecting the `hasCompetitor` relationship. That means that `Yahoo!`, `Microsoft`, and `EBay` will be selected as indirect concepts of interests, without the user having to know the exact names of Google's competitors. All this background information is being extracted from the ontology.

The direct, indirect, keyword-based search concepts of interest, and the other concepts from the ontology need to be emphasized in a graph by using for example different colors. In this way the user is able to know, by analyzing the graph, why a certain concept is being highlighted. As the size of the graphs can be very large the user is provided with zooming/panning facilities for visualizing the ontology.

The original graph of the domain ontology is also called the *conceptual graph*. Based on the user selection, a new graph is generated, the so-called *search graph* that contains only the concepts and concept relationships relevant for the query. The user can go back and forth between the two graphs performing new selections and thus updating the search graph with new concepts.

The search graph is given by the subgraph of the conceptual graph that models the user interests which is equivalent to the answer of a conjunctive query based on selected binary predicates and concepts (that form graph patterns) while keeping the

selected relationships between the concepts in the result set. The search graph has disjunctive semantics with respect to the included concepts which means that the user is interested in *any* of the search graph concepts.

Another crucial functionality of the Hermes approach is that it allows the specification of temporal constraints for news items. As news items appear at a certain moment in time and have certain time validity, it is important to be able to restrict the timestamps associated with the news. For this purpose the user can employ time comparison/arithmetic operators and retrieve the current time in order to build complex time expressions. In addition the system provides predefined temporal constraints such as: last day, last week, last two weeks, last three months, last quarter, last half year, and last year. The temporal conditions that model these constraints have conjunctive semantics as they need to be fulfilled in the same time.

Query Execution

Based on the previously selected concepts and specified temporal constraints the system can support the generation of the corresponding query in a semantic query language. This translation process involves mapping concepts and temporal constraints to query restrictions. After that, the user can trigger the query evaluation and the relevant news items are retrieved. The order of the retrieved news items is not relevant, at this stage.

RESULTS PRESENTATION

The results returned from query evaluation are presented in the order of their relevance for the user query. For this purpose, for each returned news item a *relevance degree* is computed based on all the hits between the news item and the query concepts. News items with high relevance degree are placed at the top of the retrieved news items list.

Based on previous work (Micu, Mast, Milea, Frasinca, & Kaymak, 2008) the relevance degree is defined as a weighted sum of the number of hits ($n(c_i)$), where the weights (w) depend on hits location (*title* or *body* of a news item). From our experimental results we have determined as acceptable values for w_{title} to be 2 and for w_{body} to be 1. Formula (3) presents how to compute the relevance degree.

$$relevanceDegree(news\ item) = \sum_{\substack{c_i\ found\ in\ title \\ c_i \in O \cap newsitem}} w_{title} n(c_i) + \sum_{\substack{c_i\ found\ in\ body \\ c_i \in O \cap newsitem}} w_{body} n(c_i) \quad (3)$$

News items that have the same relevance degree are sorted in descending order based on the associated timestamps (the most recent news items are presented first).

In addition to presenting the relevant news items, the system shows the query concepts in order to provide cues of the current query for which the results are computed. Also, for each returned news item, the found lexical representations stored in the hits are emphasized in the news item text, thereby offering to the user an explanation of why a certain news item is considered to be relevant.

HERMES NEWS PORTAL

The Hermes News Portal (HNP) is an implementation of the Hermes framework, which allows the user to specify queries on the considered domain using temporal constraints and subsequently retrieve the relevant news items. The presentation of HNP follows closely the steps proposed by the Hermes framework. Note that HNP is one of the possible implementations of the Hermes framework, with specific design choices, query/programming languages, and libraries used.

Operating on the Semantic Web we chose as ontology language OWL due to its expressivity and standard status. We did not opt for RDFS because OWL specific features, as for example relationships inverses (`hasCompetitor` has as inverse `isCompetitorOf`), are exploited in the conceptual graph. Lacking a true OWL query language we used SPARQL as the query language, an RDF query language that we extended with time-related functionality. This functionality is provided by implementing comparison/arithmetic time operators and functions for retrieving current time information.

The chosen implementation language is Java due to the availability of powerful libraries for manipulating, reasoning with, querying, and visualizing OWL ontologies. For manipulating and reasoning with OWL ontologies we used Jena (Jena Development Team, 2008a). For querying we employed ARQ (Jena Development Team, 2008b), the SPARQL implementation available in Jena. For visualizing ontologies we adapted the generic graph visualization library Prefuse (The Berkeley Institute of Design, 2008) for visualizing OWL graphs (Borsje & Giles, 2008). For part-of-speech tagging we used the Stanford parser (The Stanford Natural Language Processing Group, 2008). As a semantic lexicon we employed WordNet (Princeton Cognitive Science Laboratory, 2008), the largest database available online for the English language. JW1 (Finlayson, 2008) was used for the morphological analysis (finding lemmas of words) and the communication with WordNet.

We illustrate the HNP by means of the following user query: *retrieve all news items related to Google or one of its competitors that appeared in the last three months*. For this query we will go through all the different phases of Hermes: news classification, news querying, and results presentation. In the current HNP the news items duplicates removal has not been yet implemented.

NEWS CLASSIFICATION

The news classification step is responsible for indexing news items based on the domain ontology. We present this process by means of the news item example depicted in Figure 2. The first line describes the title, the second line shows the timestamp, and the remaining text represents the content of the news item.

Google to broker print ads in newspapers
6 November 2006 17:41 CET

SAN FRANCISCO (Reuters) - Google Inc. is set to begin helping customers buy advertisements in 50 U.S. newspapers in a test of how the Web search leader can extend its business into offline media, the company said on Sunday.

Figure 2 News item example

The news classification step starts by identifying lexical representations of the ontology concepts in the news item. First the tokenization, sentence splitting, part-of-speech tagging, and morphological analysis take place. Then, the concepts from the ontology are traversed once and their lexical representations are matched against the content of the news item. The following lexical representations “Google”, “extend”, and “company” are found. Next, per sentence, for each of the lexical representations with multiple senses, the word sense disambiguation procedure takes place in order to identify the used senses. The noun “Google”, having only one sense, doesn’t undergo the word sense disambiguation procedure and is mapped to the `Google` concept.

For “extend” and “company” a word sense disambiguation procedure is needed. In this process we do consider also lexical representations of concepts outside the ontology, as for example the nouns “customer” and “business”, or the verb “buy” that do appear in the news item. We select the sense that yields the smallest sum of similarities to previously disambiguated lexical representations. For “extend” it is determined as representing the `extend-verb-#1` concept with lemma `extend`, part-of-speech tag `verb`, and the first sense from WordNet. For “company” the corresponding concept is `company-noun-#1`. “customer”, a lexical representation outside the ontology, is determined as having the sense `customer-noun-#1`.

After identifying an ontology concept in a news item, a hit is stored. This hit is defined as a link between the news item and the concept together with the found lexical representation. For this purpose we decided to model a hit as an instance of the `Relation` class, which uses different properties for storing the involved news item, concept, and found lexical representation. This modeling choice is based on a best practice for modeling N-ary relations on the Semantic Web (Noy & Rector, 2006).

NEWS QUERYING

Query Formulation

Figure 3 shows the conceptual graph from which the user can select concepts of interest. Once a user selects a concept, the control panel gets activated by means of which the user can add to the search graph his concepts of interest.

The concepts of interest can only be the current concept, all related concepts including the current one, all related concepts excluding the current one, or all related concepts by means of specified relationships including/excluding the current concept. The node info panel shows information regarding the selected concept.

The local name of the different concepts is displayed using ovals. In order to emphasize the different types of concepts we use the following coloring scheme for ovals. The selected concepts are displayed in red, the concepts related to the selected one are shown in green, and the ones returned by the keyword search are presented in pink. The other concepts are displayed in yellow for classes and magenta for individuals.

In the example from Figure 3 the user has directly selected the Google concept. In the node info panel all the information related to the Google concept is displayed: name, competitors, CEO, etc. Then, the user can select the indirect concepts by specifying the hasCompetitor relationship between the direct concept and the indirect ones. The user also specifies that the Google concept should be kept in the search graph.

After selecting the concepts of interest the user can visualize the search graph. The user can refine its query by deleting some of the concepts, resetting the search, or adding new concepts from the conceptual graph to the search graph. After a number of iterations the user has added all the concepts of interests to the search graph.

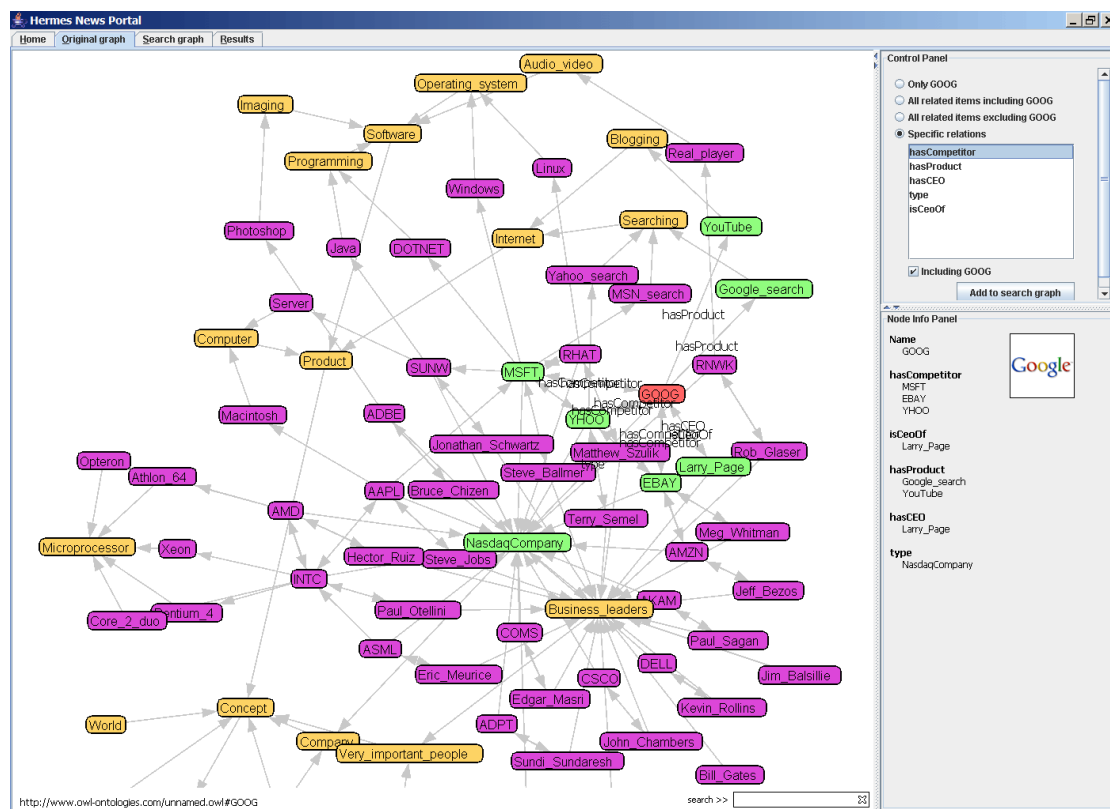


Figure 3 Conceptual graph example

Figure 4 shows the search graph, which presents the selected concepts, in this case Google and its competitors: Microsoft, Yahoo!, and EBay. In the time constraint panel the user can specify the desired time restriction. The user can choose between predefined temporal constraints as the past hour, past day, past week, past two weeks, past three months, past quarter, past half year, and past year, and specific time/date constraints.

As before the node info panel displays information about the currently selected node. However, differently than in the previous situation, only the information given by the specified relationships, in this case the `hasCompetitor` relationship, is displayed.

In this example the user has specified that the news items have to be between 1st of September 2006 and 1st of December 2006, where the current day is 1st of December 2006 (the *last three months* in the user query). Alternatively the user could have selected the past three months option from the predefined temporal constraints.

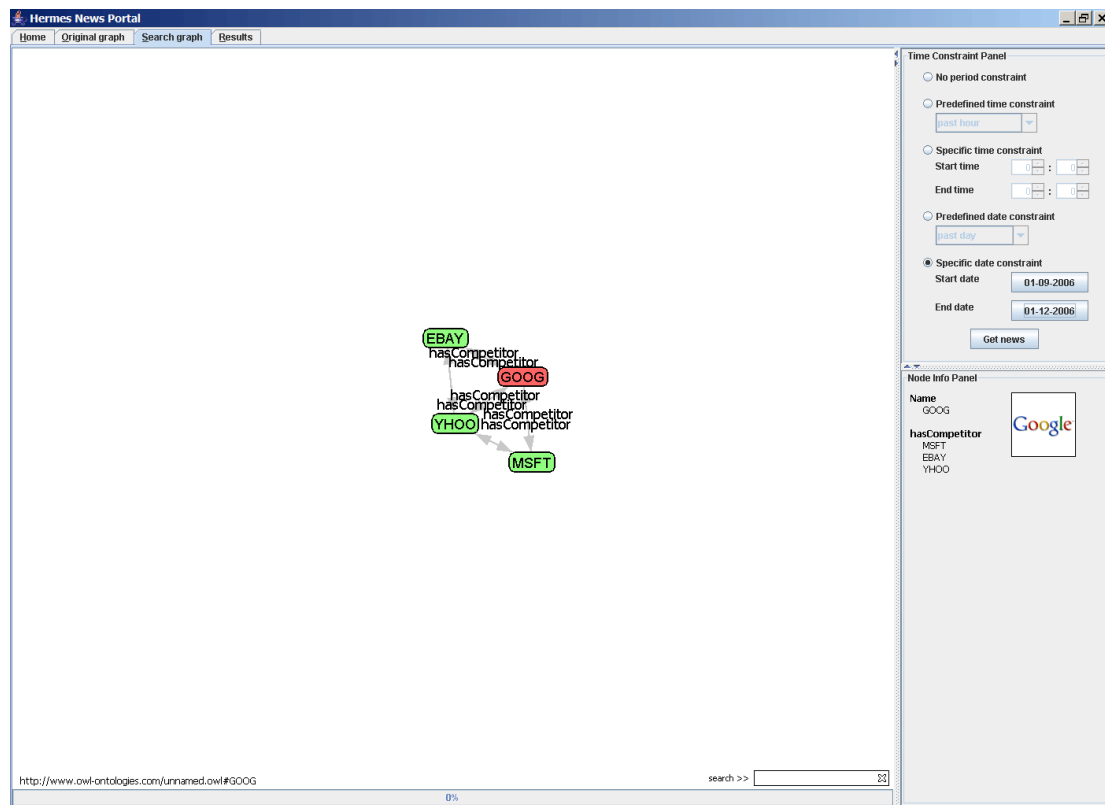


Figure 4 Search graph example

Query Execution

For each search graph a SPARQL query is generated. This query is a SELECT query as it retrieves the news items in which any of the search graph concepts are present. The disjunctive semantics of the search graph with respect to its embedded concepts is naturally specified as an 'or' filter in the SPARQL query. Also, we have decided to use filters to specify the time restrictions that news timestamps need to satisfy. Due to the conjunctive semantics of the time restrictions we modeled them as an 'and' filter in the SPARQL query.

Figure 5 shows the SPARQL query corresponding to the search graph given in Figure 4. This query is hard-coded with XML Schema `dateTime`s specifying the desired temporal boundaries of the interval in which the timestamps of the desired news items need to be contained.

The first part of the SPARQL query defines that the returned news items should be related to the concepts of interest. The second part of the query is composed of two filters. The first SPARQL filter defines the concepts of interest to be `Google`,

Microsoft, Ebay, and Yahoo!. The second SPARQL filter specifies that the timestamp of news items should be between 1st of September 2006 and 1st of December 2006, where 1st of December 2006 is the current day. Both dates are specified using XML Schema `dateTime` format.

```
PREFIX hermes: <http://hermes-news.org/news.owl#>
SELECT ?title
WHERE {
  ?news hermes:title ?title .
  ?news hermes:time ?date .
  ?news hermes:relation ?relation .
  ?relation hermes:relatedTo ?concept .
  FILTER (
    ?concept = hermes:Google ||
    ?concept = hermes:Microsoft ||
    ?concept = hermes:Ebay ||
    ?concept = hermes:Yahoo
  ) .
  FILTER (
    ?date > "2006-09-01T00:00:00.000+01:00" &&
    ?date < "2006-12-01T00:00:00.000+01:00"
  )
}
```

Figure 5 SPARQL query example

In order to ease the specification of temporal constraints in queries we have extended SPARQL with custom functions. We call the SPARQL language extended with time functions `tSPARQL`. Please note that SPARQL does naturally support such extensions, `tSPARQL` being backwards compatible with SPARQL. Figure 6 shows the signature of the time functions that we have added. These functions relate to retrieving the current date and time, the current `dateTime` instance, adding/substrating to a `dateTime` instance a duration, and subtracting two `dateTime` instances (`date`, `time`, `dateTime`, and `duration` are defined by XML Schema).

```
xsd:date currentDate()
xsd:time currentTime()
xsd:dateTime dateTime-add(xsd:dateTime A, xsd:duration B)
xsd:dateTime dateTime-substract(xsd:dateTime A, xsd:duration B)
xsd:duration dateTime-substract(xsd:dateTime A, xsd:dateTime B)
```

Figure 6 Custom time functions

Figure 7 depicts the same query as in Figure 5 but now written in `tSPARQL`. Different than in the previous case the `tSPARQL` query is not hard-coded with times and dates, but makes use of custom functions and durations. The semantics of the query is closer to its representation, in our current example that is retrieving the news items *that appeared in the last three months*.

The `tSPARQL` query uses the `dateTime-substract()` to determine the `dateTime` of three months ago, `now()` is used to obtain the current `dateTime`, and it specifies that the news timestamps should be between these two `dateTimes`. `P0Y3M` is an XML Schema duration constant that specifies a period with 0 number of years and 3 months.

After its creation, the tSPARQL query is executed. As a result, the news items that match the query constraints (concepts of interest and temporal constraints) are being returned. The order of the results is not relevant here.

```
PREFIX hermes: <http://hermes-news.org/news.owl#>
SELECT ?title
WHERE {
  ?news hermes:title ?title .
  ?news hermes:time ?date .
  ?news hermes:relation ?relation .
  ?relation hermes:relatedTo ?concept .
  FILTER (
    ?concept = hermes:Google ||
    ?concept = hermes:Microsoft ||
    ?concept = hermes:Ebay ||
    ?concept = hermes:Yahoo
  ) .
  FILTER (
    ?date > hermes:dateTime-substract(hermes:now(), P0Y3M) &&
    ?date < hermes:now()
  )
}
```

Figure 7 tSPARQL query example

RESULTS PRESENTATION

Figure 8 shows the results after the query execution. It lists the concepts of interest from the search graph (top) and the retrieved news items (bottom). In addition, the system shows the found lexical representations of the concepts of interests in the returned news items using different colors. The user is able to deselect some of the concepts of interest in order to refine his query and thus limit the result set. At the current moment the relevance degree and timestamps-based sorting of relevant news items as proposed in the Hermes framework have not been yet implemented in the HNP.

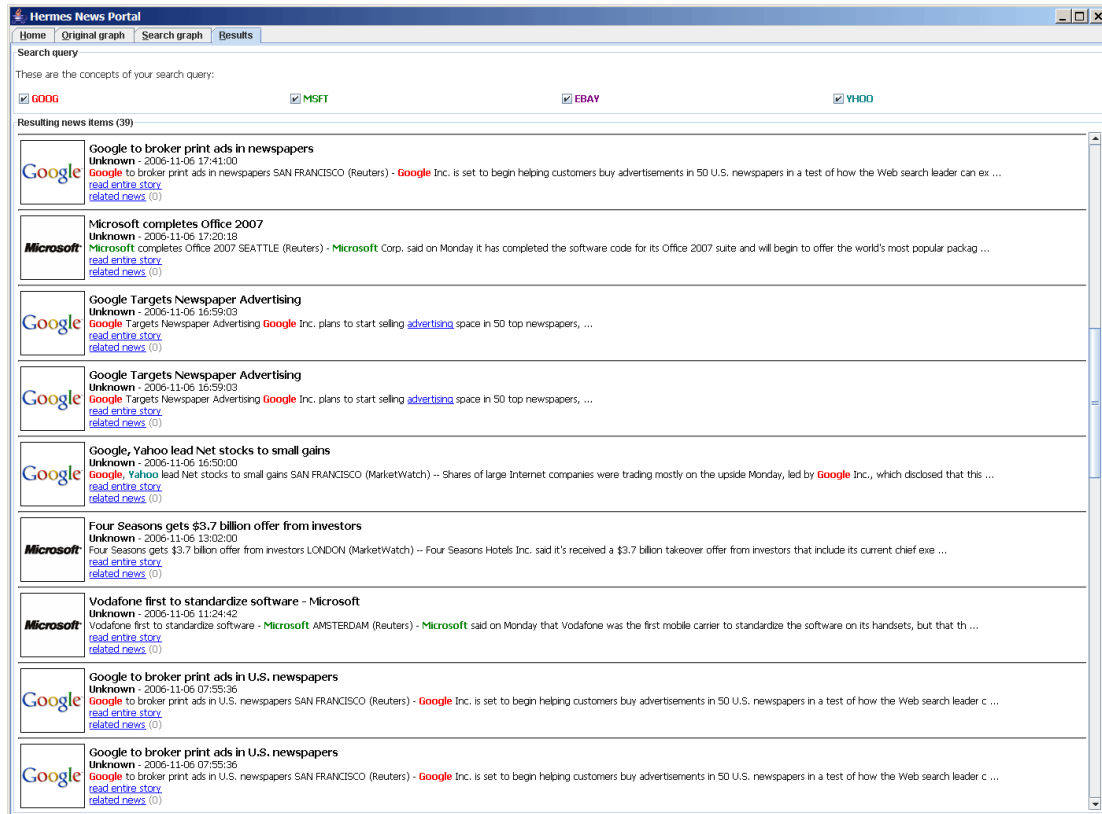


Figure 8 Results presentation example

EVALUATION

In order to evaluate the performance of the implementation we measured the concept identification precision. Precision was defined as the number of concepts correctly identified in the news items divided by the number of concepts identified in news items. If we define recall as the number of concepts correctly identified in the news items divided by the number of concepts that should have been identified in the news item, one can notice that for our application precision and recall are the same. The reason is that we are only looking for given concept lexical representation in news items which means that we identify all the concepts that should be recognized possibly with some errors. For our current implementation the precision (recall) is 85% for a given repository of around 200 news items.

Precision (recall) are based on cumulative errors through our application pipeline based on the HNP's part-of-speech recognition, morphological analysis, and word sense disambiguation algorithm. Despite using only WordNet as a semantic lexicon (compared to other approaches which combine several semantic lexicons, some domain specific (Navigli & Velardi, 2005)) we obtained high values for precision as many of our concepts' lexical representations are named entities (names of companies, CEO's, locations, etc.) that usually have only one meaning. The high value of recall can be explained by the fact that we do not aim at providing meaning for each (compound) word in news messages but only for the ones that correspond to lexical representations of ontology concepts. The meanings of the news' (compound) words that are not present in the ontology are used only to help the disambiguation process of found concept lexical representations.

A different metric for the performance evaluation is the latency of a news item in the concept identification phase. The obtained average latency time is around 30s which represents the time needed to process a news item from tokenization to concept recognition. The bottleneck lies in the disambiguation step for which distances between synsets need to be computed. As identified in the Hermes framework these distances can be pre-computed (given a certain limit for the shortest path length between synsets) which would further reduce the disambiguation time.

Regarding usability we have asked 9 users (students at Erasmus University Rotterdam following a course on Semantic Web technologies including RDF(S), OWL, and SPARQL) to find news items for 3 given natural language queries in two ways: (1) using the Hermes implementation and (2) a SPARQL engine. Most students were able to correctly build the search graph and specify the temporal constraints, as well as the corresponding SPARQL query. All queries were faster specified using the Hermes framework than using SPARQL. Note that we do not claim that it is easier to use Hermes instead of SPARQL for querying RDF graphs, but for expressing a certain set of RDF queries (the ones supported by the search graph with temporal restrictions, which we consider typical for news querying) Hermes seems to be easier to use than SPARQL. Among the features mostly appreciated by students in Hermes were the graphical representations of the conceptual graph, the predefined time functions, and the visual cues employed for emphasizing concepts in returned news items.

Compared with traditional keyword-based search engines for news items (e.g., YourNews, SeAN, Google News, Yahoo! News, etc.) our semantic approach benefits from better precision, as it is able to disambiguate (compound) words, query expressive power, because it allows the selection of indirect concepts (i.e., concepts not directly related to the items of interests), and the support for temporal constraints. A quantitative comparison with non-semantic based approaches is difficult to achieve due to the query limitations that these systems have and the impossibility of using the same news items as inputs in the compared systems.

In the tradeoff between expressivity and usability we decided to keep our queries simple with intuitive semantics so that a broad range of users (casual users, media analysts, stock brokers, etc.) should be able to use Hermes. Nevertheless, we acknowledge that an expert user might need more query expressivity (e.g., optional graph patterns, disjunctive semantics for temporal constraints, etc.) which contributes to the increase in complexity (and thus to the decrease in usability) of the framework. For this purpose we plan to extend the Hermes framework in the future with additional powerful functionality that would enable the generation of a news personalization service family (services targeting novice, average, or expert users).

CONCLUSION

The Hermes framework proposes a sequence of steps to be followed for building personalized news services. The input for these systems comprises RSS news feeds and the output are news items fulfilling user needs. The Hermes approach is based on a domain ontology used for classifying news items and to support the user define his concepts of interest. In addition the user can specify temporal constraints that the news item needs to obey. The Hermes News Portal (HNP) is an implementation of the Hermes framework. The domain ontology is specified in OWL and as a query

language we used SPARQL. As a semantic lexicon we employed WordNet, one of the most popular English dictionaries available online. For representing temporal constraints we have extended the SPARQL language with temporal functions.

Differently than Google News and Yahoo! News, Hermes is able to exploit the background information stored in ontologies for retrieving user's items of interest. In this way the user doesn't need to explicitly define all the instances involved in the query by making use of the concept relationships for specifying his concepts of interest. In addition to the concepts of interest, the user is able to specify temporal constraints in his query. Another key feature of Hermes is the word sense disambiguation procedure, which is not used in related approaches as SeAN, YourNews, MyPlanet, or SemNews. The word sense disambiguation step increases the accuracy of news classifications, by making sure that the found lexical representations indeed correspond to the meaning of the domain ontology concepts.

As future work, we would like to extend the Hermes framework by employing multiple semantic lexicons and adding specific concepts to the ontology that are not captured in existing semantic lexicons. Some domain specific concepts (e.g., domain neologisms) are used in news items while current semantic lexicons, which are not up-to-date, do not include them. We also plan to exploit the structure of the domain ontology in order to compute the similarity between concepts. In this way we enrich our knowledge base and thus are better equipped in determining concept (synset) similarity.

In addition, we would like to introduce a learning step in which new ontology instances and relations are extracted from news items. For this purpose we envisage the use of lexico-semantic rules that would extract the relevant information from news items. For example a proper noun followed by "Inc." in a news item as in "ClearForest Inc." would indicate that the proper noun is a company which needs to be inserted in the ontology if it is not already present there. In addition we would like to explore the possibilities to redefine the domain ontology as a time-based representation, where instances have a certain time validity associated with them (Milea, Frasinca, & Kaymak, 2008). These temporal extensions to the ontology would enable us to better reason with the temporal contextual information available for our domain.

Another direction that we would like to pursue is that of semantic adaptation of news items based on a user model. The user preferences now represented in the (temporary) search graph would be represented in a (stored) user model which is continuously adapted based on user behavior. In order not to bother the user with already seen content, we would like to be able to filter news items that provide new information by using a novelty control mechanism (Gabrilovich, Dumais, & Horvitz, 2004). We believe that our semantic approach can be successfully applied for modeling dissimilarities between news items and thus be able to recommend only news carrying novel content. In a different scenario, by measuring the similarities between news items, we would be able to recommend news items related to the same story but issued at different moments in time and/or by different institutions.

Regarding HNP we would like to implement news items duplicates removal, and the relevance degree and timestamps-based sorting of relevant news items, as proposed in

the Hermes framework. Additionally, we also wish to implement the previously proposed extensions to the Hermes framework: enriching our knowledge sources with multiple lexicons and domain-specific concepts, updating the domain ontology based on news information, adding a user model and employing it to adapt system functionality, and filtering news that provide novel content. Also, we would like to test the usage of data structures for fast data access (e.g., hash maps) for ontology access in a news-centric approach where the concept lexical representations are identified during a single news item traversal. Having a constant access time to concept lexical representations and taking in consideration that the number of lexical representations in a news item is smaller than in an ontology might reduce the time needed for the news classification step.

Additionally, we would like to conduct a more extensive evaluation procedure of the Hermes implementation. Based on detailed questionnaires and measuring the time spent on building queries given in natural language, we can obtain more empirical evidence on the system usability. The accuracy of the proposed relevance degree (based on concept identification) could be determined by measuring the access order and reading time of news item in the result list (accessing the first items first and spending substantial time for reading them are good indications that the returned items are relevant). In addition, we want to experiment with other domains (e.g., politics, sports, etc.) and analyze the precision and latency of our implementation for these new fields. The genericity of our approach only asks for the definition of a new domain ontology and domain-specific news feeds that need to be plugged into our implementation.

ACKNOWLEDGEMENTS

The authors are supported by the EU funded IST-STREP Project FP6-26896: Time-Determined Ontology-Based Information System for Real Time Stock Market Analysis (TOWL). More information is available on the official website of the TOWL project (TOWL Consortium, 2008). Also, we would like to thank Wouter Rijnvordt, Maarten Mulders, and Hanno Embregts for their contribution to the Hermes framework.

REFERENCES

- Ahn, J.-w., Brusilovsky, P., Grady, J., He, D., & Syn, S. Y. (2007). Open User Profiles for Adaptive News Systems: Help or Harm? In *16th International Conference on World Wide Web (WWW 2007)* (pp. 11-20). New York, NY: ACM.
- Ardissono, L., Console, L., & Torre, I. (2001). An Adaptive System for the Personalized Access to News. *AI Communications*, 14(3).
- Bechhofer, S., Harmelen, F. v., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., et al. (2004). *OWL Web Ontology Language Reference W3C Recommendation 10 February 2004*.
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5), 34-43.
- Borsje, J., & Giles, J. (2008). OWL2Prefuse. from <http://owl2prefuse.sourceforge.net/index.php>

- Brickley, D., & Guha, R. V. (2004). *RDF Vocabulary Description Language 1.0: RDF Schema*: W3C Recommendation 10 February 2004.
- Finlayson, M. (2008). The MIT Java Wordnet Interface (JWI). from <http://www.mit.edu/~markaf/projects/wordnet/>
- Gabrilovich, E., Dumais, S., & Horvitz, E. (2004). Newsjunkie: Providing Personalized Newsfeeds via Analysis of Information Novelty. In *13th International Conference on World Wide Web (WWW2004)* (pp. 482-490). New York, NY: ACM.
- Guarino, N., & Welty, C. A. (2002). Evaluating Ontological Decisions with OntoClean. *Communications of the ACM* 45(1), 61-65.
- Java, A., Finin, T., & Nirenburg, S. (2006). Text Understanding Agents and the Semantic Web. In *39th Hawaii International Conference on Systems Science (HICSS 2006)* (Vol. 3, pp. 62.62). Washington, DC: IEEE Computer Society.
- Jena Development Team. (2008a). A Semantic Web Framework for Java (Jena). from <http://jena.sourceforge.net/>
- Jena Development Team. (2008b). A SPARQL Processor for Jena (ARQ). from <http://jena.sourceforge.net/ARQ/>
- Kalfoglou, Y., Domingue, J., Motta, E., Vargas-Vera, M., & Shum, S. B. (2001). *myPlanet: An Ontology-Driven Web-Based Personalized News Service*. Paper presented at the Workshop on Ontologies and Information Sharing (IJCAI 2001).
- Kandel, E., & Marx, L. M. (1997). NASDAQ Market Structure and Spread Patterns. *Journal of Financial Economics*, 45(1), 61-89.
- Klyne, G., & Carroll, J. J. (2004). *Resource Description Framework (RDF): Concepts and Abstract Syntax*: W3C Recommendation 10 February 2004.
- Micu, A., Mast, L., Milea, V., Frasincar, F., & Kaymak, U. (2008). Financial News Analysis Using a Semantic Web Approach. In A. Zilli, E. Damiani, P. Ceravolo, A. Corallo & G. Elia (Eds.), *Semantic Knowledge Management: An Ontology-Based Framework* (pp. 311-328). Hershey, Pennsylvania: IGI Global.
- Milea, V., Frasincar, F., & Kaymak, U. (2008). Knowledge Engineering in a Temporal Semantic Web Context. In *The Eighth International Conference on Web Engineering (ICWE 2008)* (pp. 65-74). Washington, DC: IEEE Computer Society Press.
- Motta, E. (1999). *Reusable Components for Knowledge Modelling: Case Studies in Parametric Design Problem Solving* (Vol. 53). Amsterdam, the Netherlands: IOS Press.
- Navigli, R., & Velardi, P. (2005). Structural Semantic Interconnections: a Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7), 1063-1074.
- Nirenburg, S., & Raskin, V. (2001). Ontological Semantics, Formal Ontology, and Ambiguity. In *Formal Ontology in Information Systems (FOIS 2001)* (pp. 151-161). New York, NY: ACM.
- Noy, N., & Rector, A. (2006). *Defining N-ary Relations on the Semantic Web*: W3C Working Group Note 12 April 2006.
- Princeton Cognitive Science Laboratory. (2008). A Lexical Database for the English Language (WordNet). from <http://wordnet.princeton.edu/>
- Prud'hommeaux, E., & Seaborne, A. (2008). *SPARQL Query Language for RDF*: W3C Recommendation 15 January 2008.

- Salton, G. (1971). *The SMART Retrieval System—Experiments in Automatic Document Processing*. Upper Saddle River, NJ: Prentice-Hall.
- Salton, G., & McGill, M. J. (1983). *Introduction to Modern Retrieval*. New York, NY: McGraw-Hill.
- Seaborne, A. (2004). *RDQL - A Query Language for RDF*: W3C Member Submission 9 January 2004.
- The Berkeley Institute of Design. (2008). The Prefuse Visualization Toolkit. from <http://prefuse.org/>
- The Stanford Natural Language Processing Group. (2008). The Stanford Parser: A Statistical Parser. from <http://nlp.stanford.edu/software/lex-parser.shtml>
- TOWL Consortium. (2008). Time-Determined Ontology-Based Information System for Real Time Stock Market Analysis (TOWL). from <http://www.towl.org/>
- Winer, D. (2003). *RSS 2.0 Specification*: Harvard Law School.