A semantic-grained perspective of Latent Knowledge Modeling

Paola Della Rocca, Sabrina Senatore

Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata
Universitá degli Studi di Salerno - Fisciano, Italy

Vincenzo Loia

Dipartimento di Scienze Aziendali - Management & Innovation Systems
Universitá degli Studi di Salerno - Fisciano, Italy

Abstract

In the era of Web 2.0, the knowledge is the de-facto social currency in the global network environment. Knowledge is not an accumulation of data, but a relation-based representation of the information content, which needs to be distilled and arranged in a semantic infrastructure to guarantee interoperability and sharable understanding.
In the light of this scenario, the paper introduces a semantically enhanced document retrieval system that describes each retrieved document with an ontological multi-grained network of the extracted conceptualization. The system is based on two well-known latent models: Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA): LSA provides a spatial distribution of the input documents, facilitating their retrieval, thanks to an ontological representation of their relationship network. LDA works instead at deeper level: it drives the ontological structuring of the knowledge inside the individual retrieved documents in terms of words, concepts and topics. The novelty of this approach is a multi-level granulation of the knowledge: from a document matching the query (coarse granularity), to the topics that join documents, until to the words describing a concept into a topic (fine granularity). The final result is a SKOS-based ontology, ad-hoc created for a document corpus; graphically supported for the navigation, it enables the exploration of the concepts at different granularity levels.

Keywords:

Knowledge Structuring, Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), Concept Extraction, Ontology, SKOS

1. Introduction

The Web is potentially the most huge existing source of information that still needs to be distilled and transformed into accessible knowledge. In the era of big data, knowledge harvesting and modeling have become a key issue for the research avenue.
Although structured knowledge bases [1], [2] and publicly available resources [3], [4], [5] are recently catching on, there is still a huge amount of electronic information in the form of unstructured natural-language documents. Large knowledge bases are built by mining information from data-structured sources like Wikipedia; others cover specific domains and the activities to keep updated the domain changes are cost-intensive.

There are still many challenges in the knowledge discovering and modeling, especially in natural language texts [6] that should be addressed:

- discovering new entities beyond those provided by Wikipedia;
- capturing the temporal scope of facts;
- contributing to the Web of Linked Open Data (LOD), by adding *sameAs* linkage across many knowledge and data sources;
- enrich the knowledge with common sense relations;
- word disambiguation and concept identification to guarantee the right understanding of the text in natural language;
- capturing context-based sentiments and emotions enclosed into textual data.

Knowledge harvesting [6] is virtually a large-scale process that transforms raw data into structured information to feed a global source, accessible by everywhere on the Web.
In small-scale (and more realistic) approaches [7], [8], [9], [10], [11] the knowledge harvesting often regards the knowledge modeling within a limited domain (for example, a collection of documents, a community, an area of interest) [12]. The domain knowledge modeling is described as a selection of topics

(usually characterized by term sets) within a given collection (such collection can be a single document, an entire document collection, or a collection of other textual data underlying a domain) and relations between these topics. Domain knowledge modeling is of strong interest for companies and enterprise business; the performance of many organizations is determined more by their knowledge base than their real assets: an integrated, holistic view of the organization knowledge improves the knowledge sharing in the working flow and people interaction.

In the light of the outlined scenario, the paper introduces a framework for supporting the knowledge harvesting and modeling in small scale processes. Particularly, the framework accomplishes a semantically enhanced knowledge modeling from textual resources. The extracted knowledge is automatically coded in form of a SKOS ontology [13], that depicts a semantic multi-granular graph (or network) whose nodes are:

- documents, connected to each other, according to to the similar content,
- topics that describe the documents at a high level abstraction,
- concepts that are connected to the topics and detail them,
- individual words or key-phrases that compound and define a concept or a topic.

The methodologies and the techniques behind this framework are based on two latent models: Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) that synergically contribute to provide a query-based tool to retrieve documents and provide an explorable structure to navigate across all granular semantics layers. In brief, our main contributions are:

- a systematic approach to encode topic-driven knowledge into conceptual structure, by exploiting LDA modeling. The conceptual, graph-based structures encode two kinds of relations: semantic relations and grammatical relations. Particularly, semantic relations are added by using WordNet[2] and WordNet Domain,[14] which introduce new terms and domain labels respectively. Grammatical relations come from the textual parsing of the documents.

- a query-based model for documents retrieval. The method behind this modeling is LSA, that outlines a latent structure, i.e., a semantic vector space where documents are projected.

- multi-granular seamless representation of the extracted information, by an automatically generated ontology coded in SKOS language. The ontology collects all the informative granules. Facilities for the ontology visualization and navigation are provided.

The remainder of this paper is organized as follows. Section 2 sketches a brief overview of the related literature. Section 3 introduces a high-level view of the proposed framework, by describing the theoretical aspects on which our approach is based on. Main contribution of the paper is presented in Section 4, where the framework modeling is described. Experiments on our proposed framework validate the performance of our modeling: the implementation of the two latent models is individually assessed and then, a further analysis evidences relations between the system performance. Finally, the framework is evaluated as a semantic annotator. Conclusion closes the paper.

2. Related Work

Knowledge models have been developed in many research areas and disciplines, with various different applications: documents categorization, relational database, information structuring arranged for example, as a hierarchical or partially ordered graph [12]. Our review focuses on the knowledge modelling, i.e., the process to create machine-coded knowledge, providing a structure informative which connects the elements (often linguistic terms that are relevant within a given collection) through reciprocal relations.
There are mainly three different even though partially overlapping research streams that focus on this topic: Artificial Intelligence (AI), Natural Language Processing (NLP) and Information Retrieval (IR) [12].
In the AI research, the natural imprint of knowledge is to support automated reasoning: the role of Semantic Web (SW) as an extension of the traditional Web, through the sharing of machine-processable information and metadata is crucial to address this aspect. Coding (meta-) data and relations between them into an ontology, starting from unstructured text is a necessary step towards the knowledge modeling [15], [16]. Ontologies [17] and ontology-based applications (such as OntoLearn [11]) achieve natural language processing to extract domain-specific keywords from textual documents. Projects as

KnowItAll [18], DBpedia [3], Freebase [4] provide publicly available knowledge resources; some others such as ConceptNet [5], Yago [19] capture common sense knowledge; then, to preserve the quality, Cyc, OpenCyc [1] and WordNet are often built on manually compiled knowledge collection.
The knowledge modeling from natural language texts needs a clear understanding of the contextual meaning of the words. To address this issue, NLP tasks produce semantic networks (or conceptual graphs) that find correlations between terms [20], support the word sense disambiguation [21] and the text summarization [22] . Also the named-entity recognition (NER) is crucial in NLP tasks: FOX[1] [23] is an open-source framework that implements RESTful web services for providing users with disambiguated and linked named entities in several RDF serialization formats. Gate[2] is a text processing framework which offers a way of bridging NLP and SW by combining data-driven (words that describe concepts) and knowledge-driven (relations that link concepts) approaches. SHELDON [24] also represents a clear example of a NLP and SW hybridization tool: implements several machine reading task to extract RDF data from the text. It is based on FRED [25], a well-equipped tool for automatically producing RDF/OWL ontologies and linked data from text, adopting Ontology Design Patterns and Linked Data principles, relation extraction, frame detection, automatic typing of entities and the automatic labeling functions.
The IR area instead, focuses on discovering relevant documents by analyzing terms frequency rather than NLP-based relations between terms. In recent years, IR approaches are becoming more complex, trying to meet the users' preferences and behaviour during the navigation [26]; moreover, with the widespread of the paradigm Internet of Things (IoT), retrieval systems provide context-aware results for the user according to the users' physical state of the surrounding environment [27].
Nevertheless, the efficacy in retrieving documents that match a given query needs NLP techniques. IR techniques also capture correlated terms as co-occurrences within the same context (both topical and semantic), by accessing the networks of semantic relations: each further form of relatedness is crucial for the document indexing as well as the query expansion.
The research in this domain focuses on data-driven approaches: document

[1]http://aksw.org/Projects/FOX.html
[2]http://gate.ac.uk/.

clustering for the class identification; concepts clustering for the word discrimination (instead of using lexical support to disambiguation). At the same time, there are approaches [28], where formal ontologies encode the domain knowledge for query expansion and entity categorization [29]. Recently, with the widespread of Social Web and the availability of online collaborative knowledge resources, tagging and annotation have gained tremendous popularity among web users. Particularly, many applications have focused on semantic annotations, through data-driven analysis [30]. Research in the area of semantic annotation is very active e and constant progress is being made. This progress is driven by the increasing exigency to have "intelligent" documents [31], i.e., documents which "know about" their own content in order that automated processes can "know what to do" with them. The annotation of a (web) document with well-defined, machine interpretable mark-ups make the documents itself accessible from different sources and agents, by ensuring actually an unambiguous, sharable meaning. Some examples are (semi) automatic platforms like KIM [32] that provides an infrastructure for automatic semantic annotation, indexing, and retrieval of unstructured and semi-structured content. It extracts information based on the ontology and a massive knowledge base. In [33] a bootstrapping system for large scale semantic annotation by automatic mark-up is presented. Armadillo [34], and others [35], [29] are systems for knowledge modeling as well as document annotation.
Table 1 summarizes the principal frameworks and tools presented, evidencing their salient features in the Knowledge Harvesting and Modeling domain. Each row of the table describes a tool by means of the following six features/aspects:

- Research subfields involved in Knowledge Harvesting and Modeling (KHM): this feature identifies the research areas where the tools are located, according to the methodologies, functionalities and techniques employed. This feature evidences the synergy of overlapping areas aimed at addressing all the knowledge management needs, by providing a bridge of common syntax, methods, semantic structure.

- Standard Format & Ontology: standard formats are strongly required, because they provide a bridging mechanism that allows textual resources to be accessed [31] and shared easily (especially when a well-define reusable, ontological structure is provided).

- Knowledge Learning/representation: shows the method used in extracting knowledge from row data. Unsupervised, (semi-) supervised and automated methods represent a wide spectrum of approaches for the knowledge harvesting and modeling. Herein the type of knowledge representation used by the tools (i.e., entity, concept, relation extraction), as well as the quality of automation (automated or not) are sketchily presented.

- User Interface: tools that provide a simple point of entry interface, facilitate the interaction and the collaboration between users, which are two key facets for reusing semantically-enriched documents

- Ontology-based Support: in addition to supporting standard ontology formats, tools in KHM need to be able to support further ontologies, often ad-hoc designed for modeling knowledge in specific domains. At the same time, exploiting existing ontologies to describe own entities in the sake of the re-use and the universal concept understanding is a very desired requirement of the KHM.

- Semantic Annotation Support: annotating document content using semantic information from domain ontologies represents a way to ensure that annotator and annotation consumer actually share meaning [31]. Semantic annotation is not a simple textual annotation, but describes the content as a part (individual) of a concept enriched and better defined by relations in the application domain. Tools that achieve IR task with Semantic Web technologies often carries out annotation tasks as well.

Our framework covers all the features listed in Table 1; it was mainly designed for the concept learning: it provides an ontological support for the knowledge modeling as well as a user facility to graphically navigate the ontology and explore the concepts placed in this semantic structure. The main novel aspect is the multi-granule representation of the extracted knowledge, which is composed of different types of informative granules (i.e., documents, topics, words). This review is just a non-exhaustive overview of the main areas involved in Knowledge Harvesting and Modeling. It aims at highlighting how, especially in the big data era, the knowledge structuring is a mandatory task to join ad-hoc methodologies and technologies, to improve the machine-oriented knowledge understanding and guarantee a global knowledge enrichment, by sharing and re-use.

Table 1: Feature Comparison with other main tools/approaches in the Knowledge Harvesting and Modeling domain

Frameworks/Tools	Research subfields involved in KHM	Standard Format & Ontology	Knowledge Representation	User Interface	Ontology-based Support	Semantic Annotation Support
Gate	NLP, IE, IR Multi-paradigm Search [36]	RDF, OWL plugins	Automated concept learning	Yes	OWLIM, OntoText LOD repositories, SESAME RDF repository	Yes
De Maio et al. [30]	NLP, Conceptual Analysis	proprietary RDF(S), OWL	Automated concept learning	Yes	ad-hoc ontology, WordNet, DBpedia	Yes
Armadillo [34]	IE, NER pattern-based approaches web service search	RDF(S)	'Subject - Verb - Object' fact learning	Unknown	WordNet FrameNet, VerbNet	RDF(S)
KIM [32]	NER, Querying Gate-based.	proprietary RDF(S), OWL	NE-based hierarchy	Yes, with plugins	KIMO SESAME RDF repository	RDF(S), OWL
OntoLearn [11]	NLP statistical approaches	proprietary RDF, OWL	Taxonomy learning	Unknown	WordNet FrameNet, VerbNet	No
KnowItAll [18]	IE, statistical approaches	HTML	rule-based sentence extraction	Unknown	No	No
SHELDON [24]	NLP, IE	proprietary	FRED[25]	Yes, with interactive Data Visualization (infoVis^3)	DOLCE, WordNet, VerbNet, DBpedia	No
FOX [23]	NLP	proprietary	NE-based conceptualization	Yes	DBpedia	No
Cerno [35]	NLP, context-free grammar	No	user-designed conceptual model	No	WordNet, thesauri	Yes
Our Framework	NLP, IE, IR, Multi-type Querying	proprietary RDF(S), OWL, SKOS	Automated multi-granule concept learning	Yes, with interactive graphical ontology navigation (LODLive)	Local ontologies, LOD, WordNet, DBpedia WordNetDomain	RDF(S), OWL, SKOS

3. The theoretical foundation of the framework

3.1. Framework Overview

Figure 1 shows the whole framework as a black-box, evidencing the input and output. Precisely, it is composed of two main components: the *Knowldege Modeling* and *Augmented Information Retrieval*. The *Knowldege Modeling* gets as input a document collection and generates a topic-driven knowledge structuring of the input. This component uses the LDA model, that produces the word-topic-document relationship. The *Augmented Information Retrieval* achieves an Information Retrieval system, that exploits LSA model to arrange the documents in the latent space and returns relevant documents, with respect to a given query. The overall framework generates a semantic net that represents the content of the retrieved documents: words and topics from each document are linked and extended by WordNet-based semantic relations; documents with similar topics and words are linked to each other and added to this knowledge net. At each query, the knowledge is enriched with new semantic relationship that extends the net. A SKOS-based ontology encodes this knowledge. The ontology provides a multi-grained representation of the involved entities, which are listed from a coarse to a fine granularity, as follows:

- document: the biggest entity returned by the framework, it is the result to the query submission;
- topic: a document may contain one or more topics. Topics are described as word collections. In general, each document is composed of one or more topics and each topic may belong to many documents;
- word: the atomic entity of information; each word may have one or more "sense" (i.e., meaning) according to the role that it plays in the context of the word usage. In WordNet, each word sense is associated to a synset, i. e., a set of synonyms and it is correlated to other words by some semantic relations, such as synonymy, antonymy, homony, etc.

Next sections briefly introduce LDA and LSA, the two formal models used in the framework design.

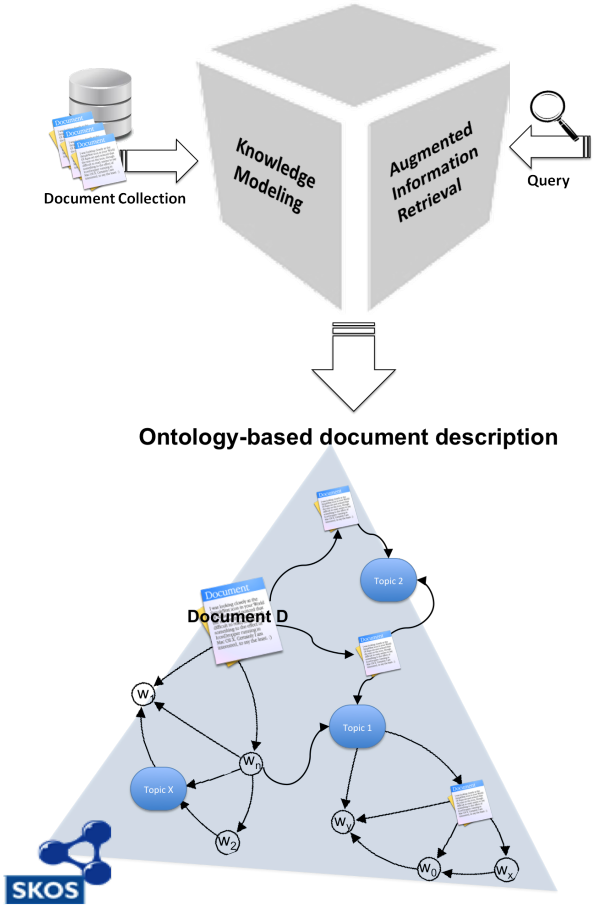

Figure 1: A high level view of the whole framework as a black-box: only input and output are evidenced.

3.2. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [37] attracted a considerable interest from the statistical machine learning and natural language processing communities. It is a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is represented as a finite mixture over an underlying set of topics, whereas each topic is modeled as an infinite mixture over an underlying set of topic probabilities. In the context of the text modeling, the topic probabilities provide an explicit representation of a document. Latent Dirichlet Allocation (LDA) models documents using a Dirichlet prior

distribution. Using this topic model, we are able to obtain a suitable Dirichlet parameter that provides the maximum likelihood for the document set. Formally, according to [37]:

- A *word* is the basic unit of discrete data, defined to be an item from a vocabulary indexed by $\{1, ..., V\}$. We represent words using unit-basis vectors that have a single component equal to one and all other components equal to zero. Thus, using superscripts to denote components, the vth word in the vocabulary is represented by a V-vector w such that $w^v = 1$ and $w^u = 0$ for $u \neq v$.
- A document is a sequence of N words denoted by $\mathbf{w} = (w_1, w_2, ..., w_N)$, where w_n is the nth word in the sequence.
- A corpus is a collection of M documents denoted by $D = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_M\}$.

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA assumes the following generative process for each document $\mathbf{w}$ in a corpus D:

1. Choose $N \sim Poisson(\xi)$.
2. Choose $\theta \sim Dir(\alpha)$.
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim$ Multinomial(θ).
   (b) Choose a word w_n from $p(w_n \mid z_n,\beta)$, a multinomial probability conditioned on the topic z_n.

A k-dimensional Dirichlet random variable θ can take values in the (k - 1)-simplex (a k-vector θ lies in the (k - 1)-simplex if $\theta_i \geq 0$, $\sum_{i=1}^{k} \theta_i = 0$), and has the following probability density on this simplex:

$$p(\theta \mid \alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} ... \theta_k^{\alpha_k - 1} \tag{1}$$

where the parameter α is a k-vector with components $\alpha_i > 0$, and where $\Gamma(x)$ is the Gamma function.

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics $\mathbf{z}$, and a set of N words $\mathbf{w}$ is given by

$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) = \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \tag{2}$$

Integrating over θ and summing over z, we obtain the marginal distribution of a document:

$$p(\mathbf{w} \mid \alpha, \beta) = \int p(\theta \mid \alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \right) d\theta \tag{3}$$

taking the product of the marginal probabilities of single documents, we obtain the probability of a corpus:

$$p(D \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \beta) \right) d\theta_d \tag{4}$$

The key inferential problem to use LDA is computing the posterior distribution of the hidden variables given a document:

$$p\left(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta\right) = \frac{p\left(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta\right)}{p\left(\mathbf{w} \mid \alpha, \beta\right)} \tag{5}$$

this distribution is intractable to compute in general. Marginalizing over the hidden variables to normalize the distribution, Eq. 3 can be written in terms of the model parameters:

$$p(w \mid \alpha, \beta) = \frac{\Gamma(\Sigma_i \alpha_i)}{\Pi_i \Gamma(\alpha_i)} \int \left(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N} \sum_{i=1}^{k} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j} \right) d\theta \tag{6}$$

a function which is intractable due to the coupling between θ and β in the summation over latent topics. The posterior distribution is intractable for exact inference. One of the approximate inference algorithms for LDA is a simple convexity-based variational algorithm, characterized by the following variational distribution:

$$q(\theta, \mathbf{z} \mid \gamma, \phi) = q(\theta \mid \gamma) \prod_{n=1}^{N} q(z_n \mid \phi_n) \tag{7}$$

where the Dirichlet parameter γ and the multinomial parameters $(\phi_1, ..., \phi_N)$ are the free variational parameters. As detailed in [37], the values of the

variational parameters γ and ϕ are found by minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior $p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta)$:

$$(\gamma^*, \phi^*) = arg \min_{(\gamma,\phi)} D(q(\theta, \mathbf{z} \mid \gamma, \phi) \parallel p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta)) \tag{8}$$

Computing the derivatives of the KL divergence and setting them equal to zero, we obtain the following pair of update equations:

$$\phi_{ni} \propto \beta_{iw_n} \exp\{E_q[\log(\theta_i) \mid \gamma]\} \gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni} \tag{9}$$

where the expectation in the multinomial update can be computed as follows:

$$E_q[\log(\theta_i) \mid \gamma] = \Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{k} \gamma_i\right) \tag{10}$$

where Ψ is the first derivative of the $\log \Gamma$ function which is computable via Taylor approximations.

Then, we wish to find parameters α and β that maximize the (marginal) log-likelihood of the data. Given a corpus of documents $D = \{w_1, w_2, ..., w_M\}$, the log-likelihood is defined as follows:

$$\ell(\alpha, \beta) = \sum_{d=1}^{M} \log p(\mathbf{w}_d \mid \alpha, \beta) \tag{11}$$

Since, the quantity $p(\mathbf{w} \mid \alpha, \beta)$ cannot be computed tractably, it is replaced with the approximation calculated using by variational inference, in order to calculate the values of α and β that maximize the log likelihood. Iterating the steps of the variational inference algorithm until the convergence of the log likelihood approximation it is possible obtain the following variational Expectation Maximization (EM) procedure:

1. (E-step) For each document, find the optimizing values of the variational parameters $\{\gamma_d^*, \phi_d^* : d \in D\}$.
2. (M-step) Maximize the resulting lower bound on the log likelihood with respect to the model parameters α and β. This corresponds to finding maximum likelihood estimates with expected sufficient statistics for each document under the approximate posterior which is computed in the E-step.

3.3. Latent Semantic Analysis

Latent Semantic Analysis (LSA) was first introduced in 1988 [38] and then it succeed as a technique for improving automatic indexing and retrieval [39], [40] also called Latent Semantic Indexing (LSI). The latent semantic structure analysis uses a matrix of terms by documents. This matrix is analyzed by singular value decomposition (SVD) to derive a latent semantic structure model. The matrix is decomposed into three other matrices of a very special form by SVD. The resulting matrices contain singular vectors and singular values. These special matrices show a breakdown of the original relationships into linearly independent components or factors. The assumptions behind LSI is that there is some latent structure in word usage that is partially obscured by variability in word choice. A truncated singular value decomposition is used to estimate the structure in word usage across documents. Retrieval is then performed using the database of singular values and vectors obtained from the truncated SVD.

More formally [40], given an $m \times n$ matrix A, without loss of generality, $m \geq n$ and rank$(A) = r$, the SVD of A, denoted by SVD(A), is defined as

$$A = U\Sigma V^T \tag{12}$$

where $U^TU = V^TV = I_n$ and $\Sigma = diag(\sigma_1, ..., \sigma_n), \sigma_i > 0 \; for \; 1 \leq i \leq r, \sigma_j = 0 for j \geq r+1$. The first r columns of the orthogonal matrices U and V define the orthonormal eigenvectors associated with the r nonzero eigenvalues of AA^T and A^TA, respectively. The columns of U and V are referred to as the left and right singular vectors, respectively, and the singular values of A are defined as the diagonal elements of Σ, which are the nonnegative square roots of the n eigenvalues of AA^T.

In order to implement LSI, a matrix of terms by documents must be constructed. The elements of the term-document matrix are the occurrences of each word in a particular document; precisely

$$A = [a_{ij}] \tag{13}$$

where a_{ij} denotes the frequency in which term i occurs in document j. Since every word does not normally appear in each document, the matrix A is usually sparse.

For purposes of information retrieval, a user's query must be represented as a vector in k-dimensional space and compared to documents. A query (as

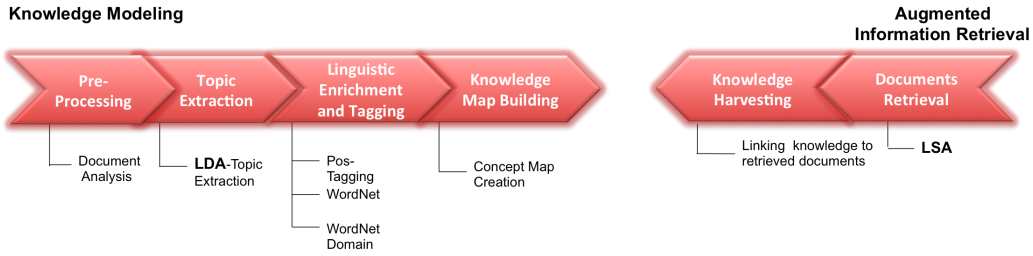

Figure 2: The whole process, across the identification of the main steps of the two components: the *Knowldege Modeling* and the *Augmented Information Retrieval*.

well as a document) is a set of words. For example, the user query can be represented:

$$\hat{q} = q^T U_k \Sigma_k^{-1} \tag{14}$$

where q is simply the vector of words in the user's query, multiplied by the appropriate term weight vectors. The query vector is compared to all existing document vectors, then the documents ranked by similarity (closeness) to the query are returned. One common measure of similarity is the cosine calculated between the query vector and document vector. Typically, the z closest documents or all documents exceeding some cosine threshold are returned to the user.

4. A closer look to the overall process

Figure 2 shows the comprehensive process overview. The *Knowldege Modeling* and the *Augmented Information Retrieval* components are described by a sequence of steps converging towards a merged output: a semantically enhanced knowledge modeling of the content extracted by the retrieved documents. Next sections will detail the process steps, sketched in Figure 2.

4.1. Knowldege Modeling

This component is in charge of the knowledge extraction and modeling from documents by NLP techniques. It uses the LDA model to collect the words describing a topic in a document. Figure 3 gives a closer look to the steps of the LDA-based component process, described as follows.

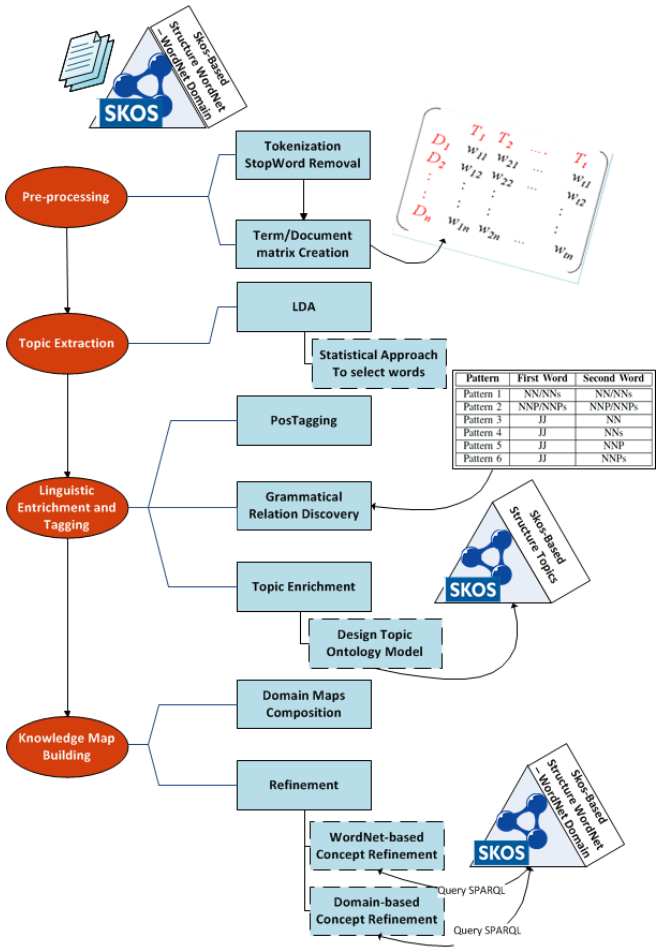

Figure 3: Data-flow of the *Knowldege Modeling* component

4.1.1. Pre-processing

The typical activities of lexical analysis, such as tokenization, stop-words removal (including numbers, punctuation and words whose length is less than three), and singularization of plural nouns are covered by this step.
The remaining document text is mapped into Vector Space Model (VSM); precisely, each document is represented as a vector of terms (or words) $d_j = (w_{1,j}, w_{2,j}, ..., w_{t,j})$ where each dimension w_{ij} corresponds to a weight associated with a term. In our LDA-based design, w_{ij} is the occurrences number of the term t_i in the document d_j. The whole document collection is represented by a term-document matrix, as required by the Latent Dirichlet

Allocation method.

4.1.2. Topic Extraction

This step is based on an implementation of variational EM for LDA[4] (see Section 3.2), which returns a topic set; each topic is composed of a list of words, ranked according to their probability to be relevant for the topic.

To select the most relevant words, the method for determining quartiles has been applied. In descriptive statistics, the quartiles of a ranked set of data values are the three points that divide the data set into four equal groups, each group comprising a quarter of the data. In our context, the most relevant words are found by considering the third quartile that splits off the highest 25% of data from the lowest 75%. Thanks to the quartile method, the topmost relevant words (whose the multinomial probability to be relevant for that topic is the highest) can be selected, depending on each topic. The general approach fixes an apriori number h of words for each topic; for example, topics are usually described by printing the top-10 terms (the 10 most probable terms) per topic [41]. Selecting the first h words for each topic could be not accurately discriminant in the topic identification, as stated in [42] that proposes a variant of LDA to improve the discrimination power of the words. The $h + 1$-th word in a topic could be more relevant (highest probability) than one among the first h words, in another topic. Prefixing the top-h words (or considering a prefixed threshold) that cuts off the words with lower value of "relevance weight" was not a good strategy to get a clear topic description. Empirical evidence has shown that in our approach, the quartile method provides words that are more relevant to their own topic.

The data extracted so far provide three types of relationship: topic-word, topic-document and document-word. These relations are stored in a knowledge base; they are the edges in a high-view graph, whose nodes are topics, words or documents. Then, the net is translated in SKOS[5], an ontological data model for sharing and linking knowledge organization systems (such as thesauri [43], classification schemes and taxonomies); it is part of the Semantic Web standards built upon RDF and RDFS. The final outcome of this step is a SKOS-based ontology populated by topic individuals, their relevant

[4]http://www.cs.princeton.edu/ blei/lda-c/

[5]http://www.w3.org/2004/02/skos/

words and the related documents.

4.1.3. Linguistic Enrichment and Tagging

The SKOS data model is extended by identifying new linguistic pointers from the ontological objects to grammar elements of a linguistic resource. This step accomplishes a kind of productive word derivational process, which adds new linguistic expressions (word compounds, key-phrases, etc.) to the initial topic words, in order to capture contextual, syntactic and language-specific features. The goal is to increase the linguistic expressiveness in describing each topic by providing additional, more structured and specific compound words that can refine context information for a better characterization of the topic meaning.

Raw text indeed was additionally processed by Stanford Parser[6]: part-of-speech (POS) tags and grammatical relations (typed dependencies), as well as named entities (NER) were returned.
The text and topic word (that is also in the text) are tagged with a grammatical category. The topic words are furthermore analyzed with respect to the grammatical relations with the surrounding text. These relations generate new links between words, for the SKOS ontology and enrich the lexical semantic net of the domain knowledge, built in the previous step. Specifically, the grammatical relations represent the following typed dependencies:

- amod: adjectival modifier. An adjectival modifier of an Noun Phrase[7] (NP) is any adjectival phrase that serves to modify the meaning of the NP [44]. As an example, the *amod* of the phrase "soft computing" is *soft*;

- nn: noun compound modifier. A noun compound modifier of an NP is any noun that serves to modify the head noun [44]. The compound word "knowledge modeling" has *knowledge* as an *nn*;

- prep_of: is a collapsed preposition that links a term to the preposition *of*;

[6]http://nlp.stanford.edu/software/lex-parser.shtml

[7]Noun phrases often function as verb subjects and objects, as predicative expressions, and as the complements of prepositions

Pattern	First Word	Second Word
Pattern 1	NN/NNs	NN/NNs
Pattern 2	NNP/NNPs	NNP/NNPs
Pattern 3	JJ	NN
Pattern 4	JJ	NNs
Pattern 5	JJ	NNP
Pattern 6	JJ	NNPs

Table 2: Extracted Phrases Patterns

- appos: appositional modifier. An appositional modifier of an NP is an NP that appears immediately to the right of the first NP that defines or modifies that NP. In the phrase "Natural Language Processing (NLP)", NLP is the *appos*.

In addition, the typed dependencies along with the tag categories are also used to identify linguistic patterns in the text. Table 2 shows the extracted patterns. They include singular and plural nouns (NN an NNs, respectively), singular and plural proper nouns (NNP and NNPs) and adjective (JJ). As an example, Pattern 3 is composed of two words: the first word is an adjective, the second is a noun. These patterns capture the whole meaning of sequence of words such as compound nouns, multiword expressions, keyphrases: they describe atomic, well-defined concepts, by a clear identification of the surrounding word context. Disambiguation and word sense identification are even the immediate results of this task.

The remaining tagged text (i.e., verbs and individual adverbs or adjectives) are discarded, because they do not generate nouns or compound nouns in our linguistic pattern analysis. Words from these grammatical categories need a deeper natural language process to discover implicit word dependencies, or grammatical relations, that are not directly derivable by the word sequences in a sentence structure. They could enrich the topics with further nouns or other grammar categories for a more complete linguistic enrichment step.

At the end of this step, each topic will be described by the original words and the extracted compound nouns that will ensure a more fine-grained description of the topic itself. Consequently, each document will be represented in turn, by the (original words of) topics, but also by the context-specific word compounds. To give an example of the linguistic enrichment, Figure

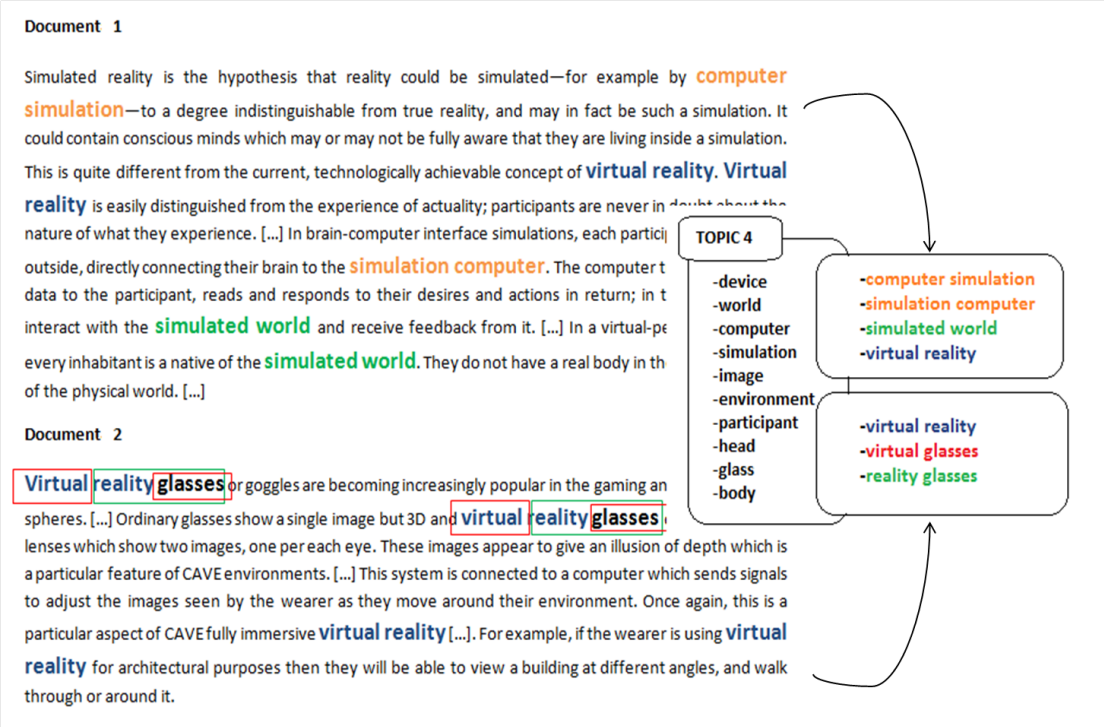

Figure 4: An example of topic enrichment: compound nouns are added to the topic, which is split in two different specialized sub-topics

4 shows two document fragments. The documents represent similar content, described by the same topic (identified as *Topic 4*). The words of the *Topic 4*, are individual words, returned by the LDA-based topic extraction. By the grammatical analysis, a topic enrichment is produced for each document: *document 1* is enriched with the compound nouns *computer simulation, simulation computer, simulated world, virtual reality* also highlighted in the document text, whereas *document 2* has different compound nouns: *virtual reality, virtual glasses, reality glasses*. Although the two documents share the same topic, after the additional grammatical analysis, they are clearly described by very context-specific word compounds.
The proposed grammatical process mainly focuses on nouns; extending the analysis to other grammatical categories could improve its robustness, in term of accuracy in the compound nouns extraction (by considering further grammatical dependencies) and specialization of the topic contexts (especially when the topics share a common subset of words).

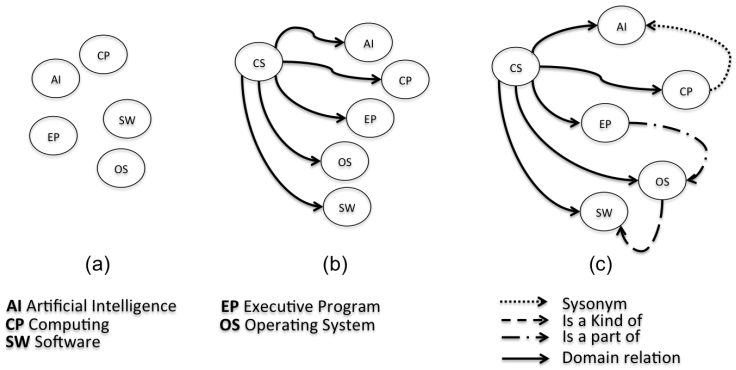

Figure 5: Knowledge Map building: Further word relations are added by WordNet and WordNet Domain.

4.1.4. Knowledge Map Building

This step achieves a complex knowledge structuring, by extending the ontology with new relations from two external semantic resources: WordNet and WordNet Domains.

WordNet Domains is a lexical resource created in a semi-automatic way by augmenting WordNet synsets with domain labels. The WordNet synsets have been annotated with at least one semantic domain label, selected from a set of about two hundred labels structured according to a well-defined WordNet Domain Hierarchy[8].

The *Domain Map Composition* subtask (shown in Figure 3) uses WordNet Domains, to assign a domain label to each word. Thus, nouns with the same domain label are joint in a conceptual "map" with that domain as a root. At the end of this subtask, all the nouns extracted in the previous step, are associated with the involved domain labels and encoded in the SKOS ontology. At the same time, WordNet suggests new relations: each noun is linked to its own WordNet synset and if a WordNet relation exists between two nouns, it is added to the ontology. Particularly, the following WordNet semantic relations have been taken into account:

1. synonymy;
2. hypernymy: Y is a hypernym of X if every X is a (kind of) Y (*canine* is a hypernym of *dog*);

[8]http://wndomains.fbk.eu/index.html

3. hyponymy: Y is a hyponym of X if every Y is a (kind of) X (*dog* is a hyponym of *canine*);
4. coordinate terms: Y is a coordinate term of X if X and Y share a hypernym (*wolf* is a coordinate term of *dog*, and *dog* is a coordinate term of *wolf*);
5. meronymy: Y is a meronym of X if Y is a part of X (*window* is a meronym of *building*);
6. holonymy: Y is a holonym of X if X is a part of Y (*building* is a holonym of *window*).

Just to give an example, let us suppose having the nouns w_1=Artificial Intelligence (AI), w_2=Software (SW), w_3=Operating System (OS), w_4=Executive Program (EP), w_5=Computing (CP), shown in Figure 5(a)). After the *Domain Map Composition* subtask, these words are all associated with the same domain label d = Computer Science (CS). In Figure 5(b) indeed, a hierarchal representation puts "Computer Science" as a root of all the selected nouns. Then, new WordNet-based relations are added; for instance a link between SW and OS is created since, OS is a WordNet hypernym of SW. Another link connects OS and EP because EP is a part of (viz., a meronym) OS. Eventually, the last link is a synonymy relation between CP and AI (see Figure 5(c)). Similarly, other relations are added enriching the remaining nouns when a semantic relation exists.
In nutshell, each noun is semantically enriched by additional related terms that characterize a conceptualization and, at the same time, it is connected to a domain label that enriches the concept. All the extracted relations feed the SKOS ontology population.

```
<rdf:Description rdf:about="Synset#6164496">
    <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
    <Synset:hasType rdf:resource="Synset#Noun"/>
    <Synset:offset>6164496</Synset:offset>
    <Synset:label>operating_system</Synset:label>
    <Synset:label>os</Synset:label>
    <Synset:meronyms>supervisory_program, executive_program  </Synset:meronyms>
    <Synset:hypernyms>software, software_system, software_package  </Synset:hypernyms>
    <Synset:domain>computer_science</Synset:domain>
 </rdf:Description>
```

Figure 6: SKOS-based statements for the noun "operating system".

Figure 6 shows an example of a SKOS-based code for the compound noun "operating system". It is defined as a SKOS concept (by the *rdf:type* prop-

erty), with the identifier *6164496*, which corresponds to the WordNet synset id (usually called offset). One or more labels can appear: in this case, a label is the term itself, whereas the other is the acronym *os*. Then, the grammar category *Noun* is also given. The discovered WordNet relations are encoded: *Synset:meronyms* and *Synset:hypernyms* provide the list of meronyms and hypernyms, respectively. One of meronyms of this noun is for instance, *executive program* that is *a part of* an *operating system*. Analogously, *software system* is *a kind of* software and it is an *hypernym* of *operating system*. A property *Synset:domain* relates the noun with the domain label of WordNet Domain *computer science*.

This noun enrichment produces a semantic map of connected words; it is achieved in the *WordNet-based concept refinement* which is the first action in the *Refinement* subtask (see Figure 3). The other task is the *Domain-based Concept Refinement*: it aims at merging maps having as a root the same domain. The process is repeated for each map built in the previous subtask, in order to connect the greatest possible number of maps that share domain labels and then getting larger conceptual maps.

The final output of this step (and also of the whole *Knowldege Modeling* component) is a comprehensive knowledge map that provides a local conceptualization (nouns and compound nouns associated with document topics), refined with external knowledge to improve the contextual knowledge representation of the input document collection. Thanks to the SKOS-based modeling, relationships between these involved concepts can be easily retrieved by means of SPARQL[9] queries.

4.2. Augmented Information Retrieval

Figure 2 shows also the LSA-based component process. It is a traditional Information Retrieval (IR) process [45], where a query is submitted as an input and a ranked list of relevant documents is returned. As stated, LSA model builds a multidimensional space where the query and the documents (through their vector-based representation) are projected. The documents whose distance from query is below a given threshold are considered relevant

[9]SPARQL Protocol and RDF Query Language. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF. See http://www.w3.org/TR/rdf-sparql-query/

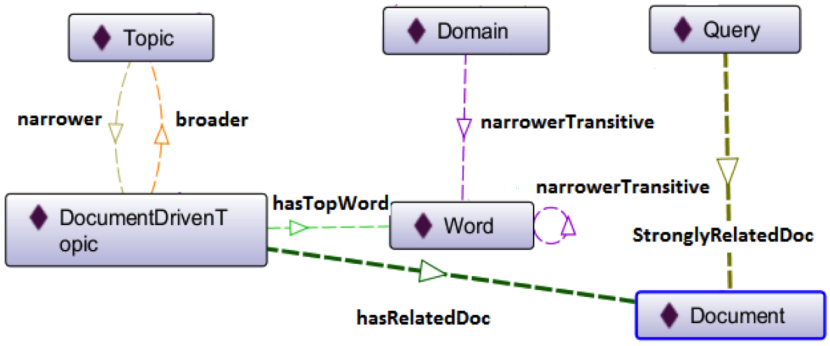

Figure 7: Ontological schema describing the concepts and relations in the global knowledge model. The concepts are all *skos:Concept* and some primitive SKOS relations have been reused: *skos:broader, skos:narrower, skos:narrowerTransitive* and *skos:broaderTransitive*. Finally, the relations named *hasTopWord*, *hasRelatedDoc* and *StronglyRelatedDoc* are *owl:ObjectProperty*.

and returned as an output. The *Documents Retrieval* (see Figure 2) accomplishes this activity.
The second step, *Knowledge Harvesting* creates an ontological bridge between the retrieved documents and the knowledge map extracted in the *Knowldege Modeling* component. In other words, each document is linked to its own topics; that in turn, are connected to the knowledge map built on the topic words.
This fusion semantically reinforces the traditional IR activity. A document matching a submitted query will return the following additional features:

- topics describing the document content;
- individual or compound words specializing the topics associated with the document;
- semantic relations among words: individual relations regarding a word (i.e., its own synonyms, meronyms, etc.) or relations between words occurring in the same topic;
- one or more domain categories.

The final result is a SKOS ontology which enables a full exploration of the involved participating entities.

Figure 7 sketches the main *skos:Concept*s and the relations of the ontological schema: the generic *Topic* concept is specialized in a more specific topic (*DocumentDrivenTopic*). In turn, it is composed of some relevant *Word*s (by means of *hasTopWord*), such as simple or compound words. The document-specific topic is associated with (*hasRelatedDoc*) a *Document*, which is retrieved from a *Query* (by means of *StronglyRelatedDoc*). Finally each *Word* can have more generic and more specific words (by the SKOS primitive *narrowerTransitive* and *broaderTransitive*) and a *Domain* category, which is related by means of the SKOS primitive *narrowerTransitive*.
Figure 8 shows a fragment of the generated ontology. The ontology visualization is provided by LodLive[10] that is a navigator of RDF resources only based on SPARQL endpoint. It provides a simple graphical browsing of the ontology structure: each node is an individual of a skos:Concept and an edge is a relation between concepts. LodLive allows exploring the ontological structure by clicking on the small balls located around each node: hidden relations can be displayed and explored by following every node. The submitted query is "operating system" (the bottom right node in Figure 8). The figure shows the semantic structure for a retrieved document, called *comp10.txt*: the document is related (by *hasTopic* relation) to the topic identified as *topic5*; in turn, the topic is characterized by the words *program, computer system, computer program, operating system, computer hardware*. A word can be semantically related to other words: for instance, the word *software* is a narrower term than *program*, but it is broader than *computer program* and so on. Finally, the figure shows that all the terms are connected to the domain *Computer Science* (on the far right in the whole screen snapshot).

5. Experimental results

To provide a straightforward evaluation of the system performance, it was necessary to analyze the performance of each system component, depending on its own specific functionality. For this reason, the system performance evaluation consists of the following quality and effectiveness measurements:

- topic quality: the LDA-based component supports in the identification of a consistent and accurate number of topics. This aspect is crucial for

[10] http://blog.lodlive.it/

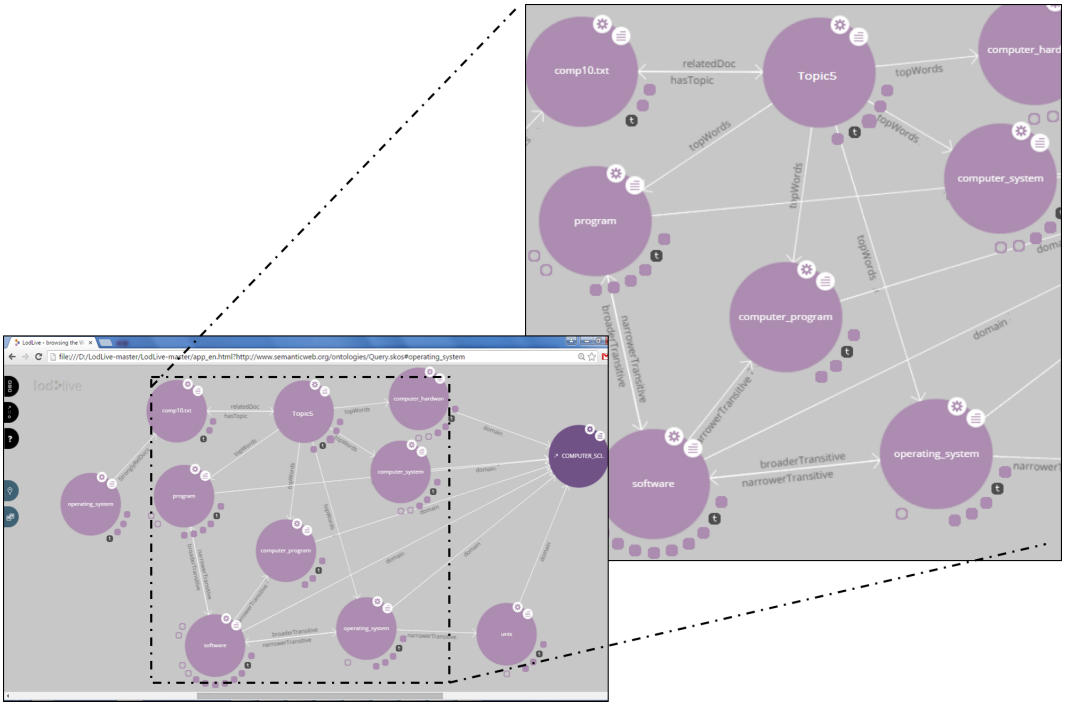

Figure 8: LodLive visualization of the knowledge structuring associated with a document, retrieved by the query "operating system".

the knowledge structuring: a smaller or bigger number of topics could upset the final ontology representation of the knowledge.

- effectiveness of the retrieval: the LSA-based component provides a simple way to retrieve relevant documents; traditional IR measures, such as the recall and precision are used to evaluate the retrieval performance.

- effectiveness of the retrieval with respect to the number of involved topics: this analysis aims at providing a comprehensive performance evaluation, based on the two main components, strongly dependent on one another. For this reason, recall and precision are evaluated with respect to the number of pertinent topics involved.

- effectiveness of the semantic annotation: although the work present an IR system, its goal is (also) the semantic structuring of knowledge. Measuring the quality of the ontology, depending on the level of concept refinement is not easy, but evaluate the named entities discovered

Table 3: Datasets used in the experimentation

Dataset Name	**#document**	**#words**	**#queries**
CACM	3204	8340	64
MED	1033	12064	30

during the text analysis is a typical task of the semantic annotation. Our system can work as a primary semantic annotation system, thus the annotation accuracy is evaluated by a comparison with known annotation tools.

Next section will describe the datasets used in this experimentation, and then the performance will be introduced according to the listed evaluation measures.

5.1. Dataset Description

We validated our framework by considering the following two datasets (Table 3).

- *CACM*: it is a common dataset used in early information retrieval evaluation, composed of 3204 abstracts from the Communications of the ACM Journal and 64 queries.
- *MED*: it is composed of medical abstracts from the National Library of Medicine and comprises 1033 documents and 30 queries. It has been widely used to evaluate information retrieval systems.

As stated in Section 4.1, the texts extracted from each corpus was preprocessed, deleting stopwords, numbers, punctuation and terms with a length less than three. The final result is in form of a term-document matrix.

5.2. LDA-based system topic quality

According to the traditional topic modeling experimentation, a mandatory step is to build fine-grained, high-quality topic models from domain-specific corpora. The purpose is therefore to explore the extent to which information collected by documents can be used to assess the topic quality. In order to determine the adequate number of topics, we used the coherence

and entropy measures. According to [46], the coherence is calculated as the sum of pairwise distributional similarity. It is defined as follows:

$$coherence(V) = \sum_{v_i, v_j \in V} score(v_i, v_j, \varepsilon) \tag{15}$$

where V is a set of words describing the topic and v_i, v_j are two words in V and ε indicates a smoothing factor which guarantees that *score* returns real numbers. The score is calculated with the UMass metric [47]:

$$score(v_i, v_j, \varepsilon) = \log \frac{D(v_i, v_j) + \varepsilon}{D(v_j)} \tag{16}$$

where $D(v_i, v_j)$ counts the number of documents containing words v_i and v_j and $D(v_j)$ counts the number of documents containing v_j.
Just to give an example of the coherence and entropy trends, Figure 9 shows the tendency for the CACM dataset. Specifically, in Figure 9(a), the average coherence value is shown by varying the number of topics. The model reaches a stable average after 300 topics, remaining in the range $[-110, -120]$. As described in [46] the coherence entropy is also calculated by dropping the log and ϵ factor from each scoring function. As shown in Figure 9(b), the entropy tendency also is quite stable after 300 topics.

A similar study has been applied on the MED dataset, discovering a trend stabilization with 100 topics.

5.3. LSI-based system analysis

Our framework has been analyzed as an IR system, by considering the LSA (or more appropriately, in this specific case, LSI) component. Traditional measures to assess the information retrieval effectiveness are the precision (P) and the recall (R), that usually are applied for binary classification. To evaluate performance average across categories, micro-average recall and precision as well as macro average recall and precision represent the extension of basic definition [48], [49].

Specifically, given $q_1, q_2, ..., q_k$ a set of queries on a documents benchmark, let $RetrievedDocs_{q_1}, RetrievedDocs_{q_2}, ..., RetrievedDocs_{q_k}$ be the corresponding resulting retrieved documents associated with each query q_i ($i = 1, \ldots, k$) and $RelevantDocs_{q_1}, RelevantDocs_{q_2}, \ldots RelevantDocs_{q_k}$ are the expected relevant documents (i.e., the class of documents) for each query. The micro average of the precision (P_{micro}) and recall (R_{micro}), the macro

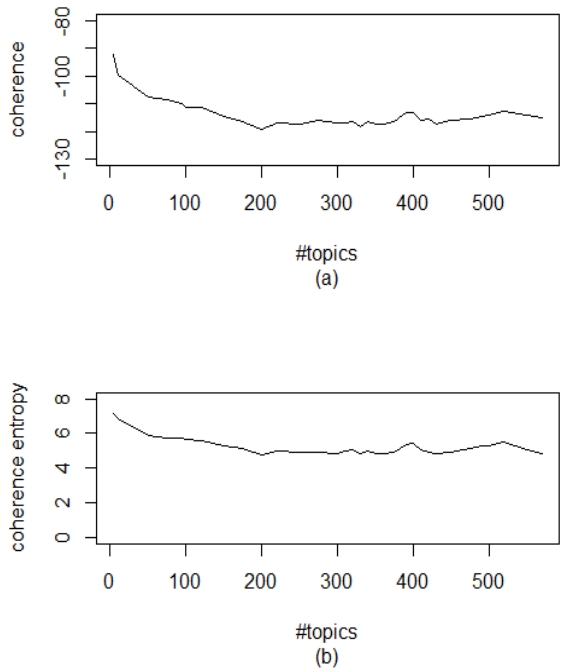

Figure 9: Coherence and coherence entropy on CACM, in the topic range $[10, 500]$

average of precision (P_{macro}) and recall (R_{macro}) are defined respectively as follows:

$$P_{micro} = \frac{\sum_{i=1}^{k} \mid RelevantDocs_{q_i} \cap RetrievedDocs_{q_i} \mid}{\sum_{i=1}^{k} \mid RetrievedDocs_{q_i} \mid} \tag{17}$$

$$R_{micro} = \frac{\sum_{i=1}^{k} \mid RelevantDocs_{q_i} \cap RetrievedDocs_{q_i} \mid}{\sum_{i=1}^{k} \mid RelevantDocs_{q_i} \mid} \tag{18}$$

$$P_{macro} = \sum_{i=1}^{k} \frac{\mid RelevantDocs_{q_i} \cap RetrievedDocs_{q_i} \mid}{\mid RetrievedDocs_{q_i} \mid} \tag{19}$$

$$R_{macro} = \sum_{i=1}^{k} \frac{\mid RelevantDocs_{q_i} \cap RetrievedDocs_{q_i} \mid}{\mid RelevantDocs_{q_i} \mid} \tag{20}$$

Macroaveraging gives equal weight to each class, whereas microaveraging gives equal weight to each per-document classification decision. According

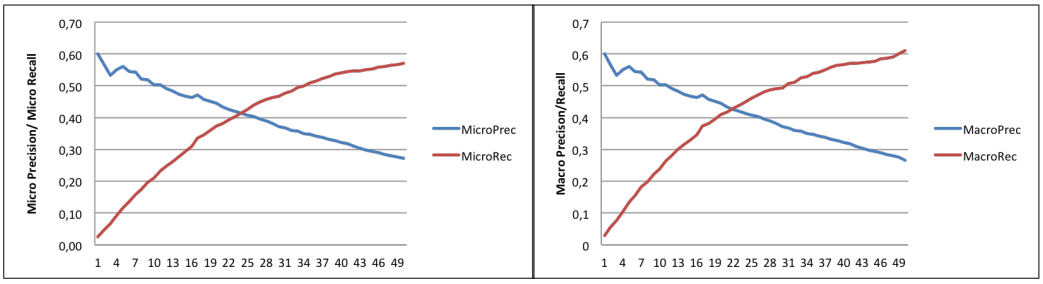

Figure 10: Micro-averaged Recall and Precision, Macro-averaged Recall and Precision on MED collection, with $r = 100$

to (17), (18), large classes dominate small classes in microaveraging. Microaveraged results provide a measure of effectiveness on the large classes in the document collection (it may be preferred in multilabel settings). The macroaveraged results instead, supports the effectiveness on small classes, because it gives equal weight to every class, instead of to give equal weight to each document as in the microaveraging. Moreover, we evaluate the most relevant documents in the set of topmost retrieved ones, by calculating $P_{q_i}@n$ and $R_{q_i}@n$ for each query q_i, where n is the number of the topmost retrieved documents. Figure 10 shows the micro and macro averaged precision ($P_{micro}@n$ and $P_{macro}@n$) and micro and macro averaged recall ($R_{micro}@n$ and $R_{macro}@n$) calculated on MED dataset, given 30 queries. The x-axis represents the topmost retrieved documents by varying n from 1 up 50.
Let us notice that the tendency of micro and macro averaging of two measures are very similar; particularly the micro precision and the macro precision show an almost identical trend (variation between the two measure are in the order of 10^{-2}). This is due to the fact that the documents are almost equally distributed among the categories (i.e., the groups of documents which are relevant for each query). The best averaged values of precision are obtained by considering only the first few documents in the returned ranked list of documents. Conversely the averaged value of recall improves by enlarging the "window" of topmost retrieved documents. These results are obtained by considering the number of dimensions (rank), $r = 100$ in the reduced LSI representation (see Section 3.3).

Similarly, Figure 11 shows the tendency of micro-average and macro-average precision and recall on the CACM dataset. The dimension r of reduced LSI is 800, chosen according to [50]. As described in [50], the CACM dataset is particularly challenging for LSA: queries are formulated in natu-

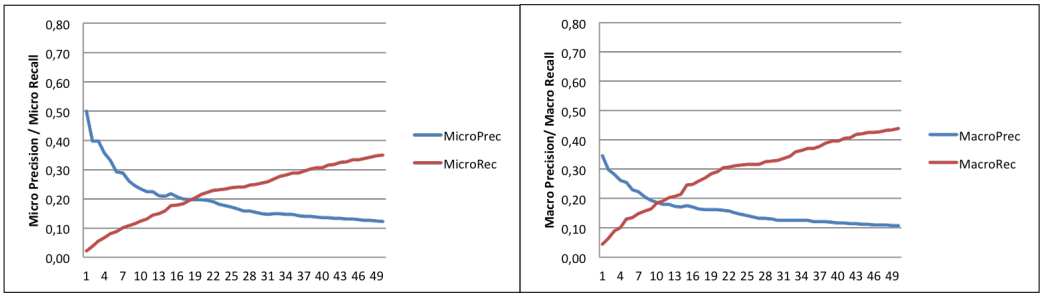

Figure 11: Micro-averaged Recall and Precision, Macro-averaged Recall and Precision on CACM dataset (original queries), with $r = 800$

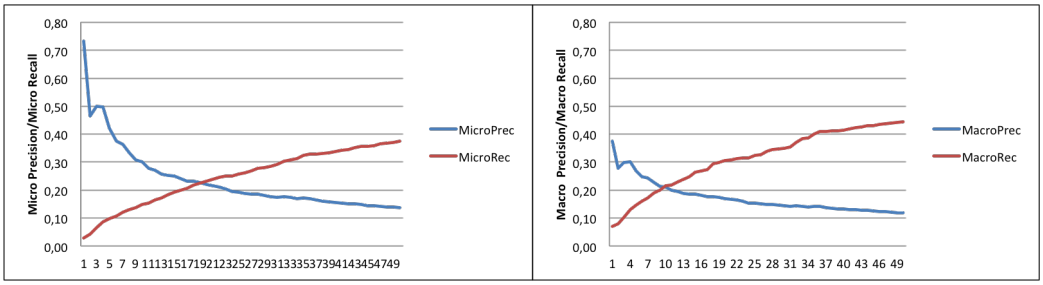

Figure 12: Micro-averaged Recall and Precision, Macro-averaged Recall and Precision on CACM dataset (revised queries), with $r = 800$

ral language, close to human interpretation, rather than machine use. For instance, queries as : "*Interested in articles on robotics, motion planning particularly the geometric and combinatorial aspects. We are not interested in the dynamics of arm motion*" are not easily processable by a search system. In this case, it is probably that also documents which deal with "*dynamics of arm motion*" are retrieved.

Figure 12 shows indeed, the micro and macro averaging of recall and precision, computed after revising the original set of queries, by discarding some simple natural language sentences that could add noise (i.e., linguistic expressions that are not directly related to the meaning of the query), in order to make them directly machine processable. For example, expressions like "I am interested in" are eliminated: they help the user to define the query, but need further language processing from the search engine viewpoint. The tendency curves of micro-average precision tends to improve, especially for the precision that reaches values above 70%. The micro recall improvement

instead is minimal with respect to the recall tendency with revised queries. The retrieval of relevant documents is tricky when the queries are expressed in natural language, especially when they are composed of complex sentences. The word matching achieved by LSA-component could be ineffective to retrieve documents: the intended meaning of the query could be not coded in the words of the sentences explicitly. In these case, the human interpretation is crucial to select relevant (or not) documents.
As expected, the precision is improved: a query formulation that is more oriented to the machine processing allows retrieving more documents that are coherent with the given query.

Comparing the micro and macro averaging of recall in Figures 11 and 12, it seems that the macro-recall tendency assumes higher values when the number of topmost documents increases; that means that increasing the retrieved documents, small group of documents (classes) relevant for a certain query get more strength; aspect that is not evident in the micro-average recall where all the documents have the same weight. Conversely, micro precision tendency is higher than macro-precision especially when the number of retrieved documents is small; this aspect is supported by the fact that retrieving a few individual relevant documents reinforces the micro-precision, targeted at evaluating the individual documents, whereas weakens the macro-precision that supports the documents classes.

5.4. Analysis of effectiveness of the retrieval with respect to the discovered topics

In the light of the performance evaluation of the individual components that compose the whole system, this section is devoted providing a further system evaluation in order to show conceivable relations between the IR performance (in term of recall and precision, from the LSA component), and the topics discovered (from the LDA component). This analysis shows the level of description (specialization) of a topic in accurately describing the retrieved and/or relevant documents.

As first analysis, Figure 13 shows the precision and recall, given a query from MED collection, with respect to the number of topics involved with respect to the number of the retrieved documents.

Let us notice that initially, the retrieved documents are effectively all the relevant (high precision) and there is almost a 1:1 correspondence with the topics. It seems that the number of topics increases quite linearly by increasing the recall: that means a relevant document is described by a specific

topic. Steps in the *#topics* curve evidences that there are some documents with the same topic. For this query, the first 50 topmost retrieved documents are associated with 36 topics. The best recall is computed with more than 38 documents and at least 28 topics. The recall tendency intersects the precision at a middle point, which represents about the 57% of retrieved documents that are relevant (precision) and, at the same time, the 57% of relevant document that are retrieved (recall), with 20 topics, on 27 documents. This evidences once again that the topics are very specialized and specific for each retrieved document.

Figure 14 instead shows the tendency of recall, precision and number of

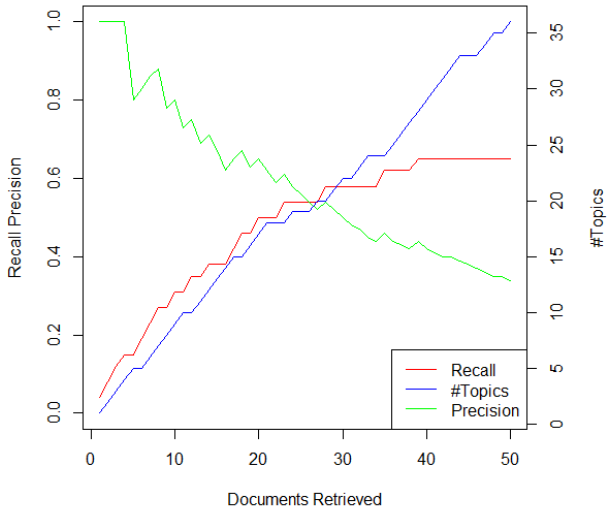

Figure 13: Recall, Precision and Topics evaluated for the query n.5: *"the crossing of fatty acids through the placental barrier. normal fatty acid levels in placenta and fetus."* from MED collection

topics on the whole set of queries from MED dataset. Precisely, the figure plots the average value of recall, precision and topics on all the queries. The curves behaviour is similar: at the intersection point of recall and precision (whose value is about 50%), the topics involved are 17, on about 35 documents retrieved. On average, there is a new topic that characterizes each two of documents. Similar observations hold for the CACM dataset, shown in Figure 15: the recall intersects the precision when about 20 documents have

been retrieved and the number of topics is 14.

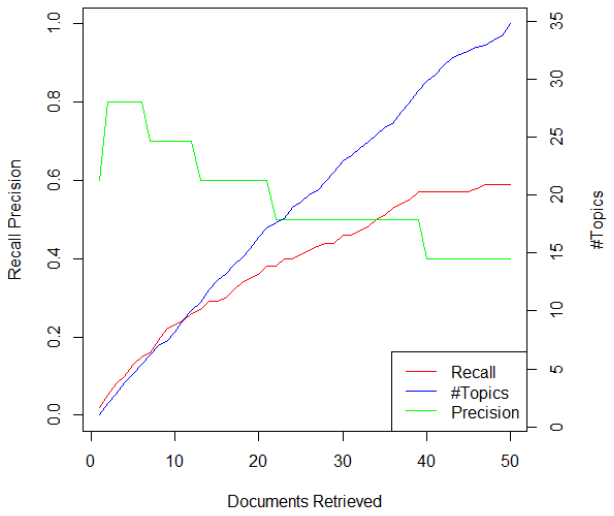

Figure 14: Recall, Precision and Number of Topics evaluated on all the queries from MED collection

5.5. A simple annotation system

Although the primary goal of our framework is an IR system, equipped with the ontological support, it can work as a basic annotator: as described in Section 4.1.3, the Stanford parser identifies also named entities which appear among the words of a topic.

Although this framework was not developed specifically for the semantic annotation, it has been tested as an annotator and compared with other typical semantic annotation platforms. The precision, recall, and F-measure are measures often used to evaluate the effectiveness of annotation systems. A slightly modified version of the recall and precision has been defined, to work with words playing the role of annotation [51]. Annotation can be individual terms or compound terms. Let *human_ann* be the set of all annotations provided by a human, whereas where *acc_ann* and *inacc_ann* refer to annotations generated semi-automatically by a semantic annotation platform. The recall and precision for semantic annotation could be expressed as follows:

$$AnnotationRecall = \frac{acc\_ann}{human\_ann} \tag{21}$$

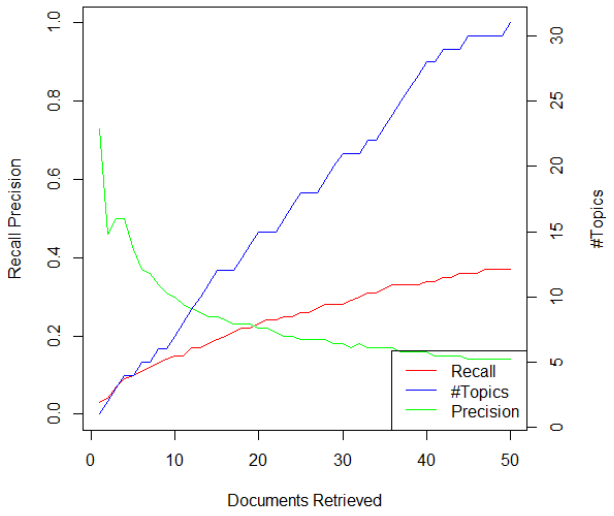

Figure 15: Recall, Precision and Number of Topics evaluated on all the queries from CACM collection

$$AnnotationPrecision = \frac{acc\_ann}{acc\_ann + inacc\_ann} \tag{22}$$

As stated in [35], it is not really feasible to directly compare the annotation quality of different tools because of unavailability of implementations, semantic models and comparable data sets. For this purpose, an approximate comparison was applied, by considering the author-reported performance previously summarized in Table 4 [51].

Our framework provides interesting performance, especially considering the fact that it was not designed for the semantic annotation. In the current version, it overcomes the performance of the third tool in the table, with discrete values in terms of precision, recall and F-measure. In order to improve the performance, a first improvement should be an extension of the named entity recognition: in our framework, it is limited to find the named entities that are in the term set computed for the topic modeling. The LDA component indeed, provides only individual words, so compound named entities are lost. As a future work, an enhancement of the text processing has been taken into account, including the improvement of the named entities recognition activity.

Table 4: Quality performance rates for different tools.

Framework	Precision	Recall	F-measure
Armadillo	91.0	74.0	87.0
KIM	86.0	82.0	84.0
Ont-O-Mat:PANKOW	65.0	28.2	24.9
SemTag	82.0	n/a	n/a
Cerno	90.6	90.8	90.7
Our Framework	66.7	42.8	52.1

6. Conclusions

As the number of online web resources continues to increase, the need of knowledge structuring is becoming a crucial theme in many approaches concerning the Knowledge Modeling and Harvesting domain. To address this issue, our approach provides an enhanced document retrieval approach that helps the user to seek documents that are relevant for the content quality.

The framework modeling exploits two well-known latent models, LDA and LSA that are often individually used in that domain; the merging of these models represents the novelty of our contribution: a semantic net of knowledge in form of an ontology that gathers all the document content. An additional contribution is the run-time building of semantic net associated to the documents retrieved by the submitted query. Our framework generates indeed, for each retrieved document, an ontological structure that describes the document content, at different semantic granulation: topics, concepts and terms.
The ontological representation suits to be explored and visualized across all the granularity levels, starting from a document matching the query (coarse granularity), to the topic that is associated with the documents, until to the words describing a concept into a topic (fine granularity).

In order to valide the approach, the experimentation concentrates on showing the effectiveness of the principal components of the framework. Although the framework achieves an enhanced document retrieval, the basic IR approach has been accomplished by the LSA component. Traditional IR measures has been used to assess the system performance, by showing the trend of micro and macro precision and recall on the query collection. Then, the quality of topics extracted by LDA component from the documents has been validated by the coherence measure in the topic words. Since the two

components feed the knowledge modeling, a study discovers the relationship between the topics extracted and the relevant documents.
The actual contribution is the knowledge modeling supported by an ad-hoc defined ontology, that join together all the extracted data in a comprehensive net (also enriched by external lexical resources). Since it is easy to measure the quality of an ontology generated, the system performance has been evaluated as a semantic annotator, by a comparison with other existing annotation systems. In nutshell, our system achieves the following tasks:

- typical search engine: returns relevant documents, matching the keyword-based user query;
- graphical ontology navigator: retrieved documents can be explored inside the ontology by following semantic relations.
- knowledge model: each document is semantically connected to its own content: the documents is split in topics, words, keyphrases, but, at the same time, it is connected to other documents, (with) other topics, words, etc.
- linked data: this semantic network is compliant to the LOD (Linked Open Data)[11] principles, i.e., it is build on the standard Web technologies (HTTP, RDF, URIs) can query the data on the net, draw inferences using external vocabularies. Let us remark that our ontology is modeled to connect the extracted information with external resources such as WordNet and WordNet Domain. This aspect holds the underpinning foundations of the linked data.
- semantic annotator: topic, words, key-phrases generated by the analysis of extracted terms as well as the WordNet-supported semantic relations describes individual documents by semantic tagging.

A future extension is to use additional external resources such as ConceptNet or BabelNet [52], in order to better individuate the topic context for each document. To this end, also a deeper textual analysis should be taken into account.

[11] https://www.w3.org/standards/semanticweb/data

7. References

[1] C. Matuszek, J. Cabral, M. Witbrock, J. Deoliveira, An introduction to the syntax and content of cyc, in: Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, 2006, pp. 44–49.

[2] G. A. Miller, Wordnet: A lexical database for english, Communications of the ACM 38 (1995) 39–41.

[3] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 722–735.
URL http://dl.acm.org/citation.cfm?id=1785162.1785216

[4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: A collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08, ACM, New York, NY, USA, 2008, pp. 1247–1250. doi:10.1145/1376616.1376746.
URL http://doi.acm.org/10.1145/1376616.1376746

[5] R. Speer, C. Havasi, Representing general relational knowledge in conceptnet 5, in: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, May 23-25, 2012, 2012, pp. 3679–3686.

[6] F. Suchanek, G. Weikum, Knowledge harvesting in the big-data era, in: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, SIGMOD '13, ACM, New York, NY, USA, 2013, pp. 933–938. doi:10.1145/2463676.2463724.
URL http://doi.acm.org/10.1145/2463676.2463724

[7] U. Kruschwitz, M.-D. Albakour, J. Niu, J. Leveling, N. Nanas, Y. Kim, D. Song, M. Fasli, A. De Roeck, Moving towards Adaptive Search in Digital Libraries, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 41–60. doi:10.1007/978-3-642-23160-5_4.

[8] X. Liu, Y. Song, S. Liu, H. Wang, Automatic taxonomy construction from keywords, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, ACM, New York, NY, USA, 2012, pp. 1433–1441. doi:10.1145/2339530.2339754.
URL http://doi.acm.org/10.1145/2339530.2339754

[9] J. Diederich, W.-T. Balke, et al., Automatically created concept graphs using descriptive keywords in the medical domain, Methods of information in medicine 47 (3) (2008) 241–250.

[10] L. Yuan, Y. Ge, F. Yin, Q. J. Wu, A Rapid Learning Approach for the Knowledge Modeling of Radiation Therapy Plan, Springer International Publishing, Cham, 2015, pp. 1492–1494. doi:10.1007/978-3-319-19387-8_362.

[11] R. Navigli, P. Velardi, A. Cucchiarelli, F. Neri, Quantitative and qualitative evaluation of the ontolearn ontology learning system, in: Proceedings of the 20th International Conference on Computational Linguistics, COLING '04, Association for Computational Linguistics, Stroudsburg, PA, USA, 2004. doi:10.3115/1220355.1220505.
URL http://dx.doi.org/10.3115/1220355.1220505

[12] M. Clark, Y. Kim, U. Kruschwitz, D. Song, D. Albakour, S. Dignum, U. C. Beresi, M. Fasli, A. De Roeck, Automatically structuring domain knowledge from text: An overview of current research, Inf. Process. Manage. 48 (3) (2012) 552–568. doi:10.1016/j.ipm.2011.07.002.
URL http://dx.doi.org/10.1016/j.ipm.2011.07.002

[13] A. Miles, B. Matthews, M. Wilson, D. Brickley, Skos core: Simple knowledge organisation for the web, in: Proceedings of the 2005 International Conference on Dublin Core and Metadata Applications: Vocabularies in Practice, DCMI '05, Dublin Core Metadata Initiative, 2005, pp. 1:1–1:9.
URL http://dl.acm.org/citation.cfm?id=1383465.1383467

[14] B. Magnini, G. Cavagli, Integrating subject field codes into wordnet, 2000, pp. 1413–1418.

[15] R. Navigli, P. Velardi, From glossaries to ontologies: Extracting semantic structure from textual definitions, in: Proceedings of the 2008

Conference on Ontology Learning and Population: Bridging the Gap Between Text and Knowledge, IOS Press, Amsterdam, The Netherlands, The Netherlands, 2008, pp. 71–87.
URL http://dl.acm.org/citation.cfm?id=1563823.1563830

[16] J. Yao, V. V. Raghavan, Z. Wu, Web information fusion: A review of the state of the art, Information Fusion 9 (4) (2008) 446 – 449, special Issue on Web Information Fusion. doi:http://dx.doi.org/10.1016/j.inffus.2008.05.002.

[17] S. Staab, R. Studer, Handbook on Ontologies, 2nd Edition, Springer Publishing Company, Incorporated, 2009.

[18] A. Fader, S. Soderland, O. Etzioni, Identifying relations for open information extraction, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 1535–1545.
URL http://dl.acm.org/citation.cfm?id=2145432.2145596

[19] F. M. Suchanek, G. Kasneci, G. Weikum, Yago: A core of semantic knowledge, in: Proceedings of the 16th International Conference on World Wide Web, WWW '07, ACM, New York, NY, USA, 2007, pp. 697–706.

[20] Z. Kozareva, E. Hovy, A semi-supervised method to learn and construct taxonomies using the web, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 1110–1118.
URL http://dl.acm.org/citation.cfm?id=1870658.1870766

[21] R. Navigli, Word sense disambiguation: A survey, ACM Comput. Surv. 41 (2) (2009) 10:1–10:69. doi:10.1145/1459352.1459355.
URL http://doi.acm.org/10.1145/1459352.1459355

[22] G. Erkan, D. R. Radev, Lexrank: Graph-based lexical centrality as salience in text summarization, J. Artif. Int. Res. 22 (1) (2004) 457–479.
URL http://dl.acm.org/citation.cfm?id=1622487.1622501

[23] R. Speck, A.-C. N. Ngomo, Named entity recognition using fox, in: Proceedings of the 2014 International Conference on Posters & Demonstrations Track - Volume 1272, ISWC-PD'14, CEUR-WS.org, Aachen, Germany, Germany, 2014, pp. 85–88.
URL http://dl.acm.org/citation.cfm?id=2878453.2878475

[24] D. Reforgiato Recupero, A. G. Nuzzolese, S. Consoli, V. Presutti, M. Mongiovì, S. Peroni, Extracting knowledge from text using sheldon, a semantic holistic framework for linked ontology data, in: Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion, ACM, New York, NY, USA, 2015, pp. 235–238. doi:10.1145/2740908.2742842.
URL http://doi.acm.org/10.1145/2740908.2742842

[25] V. Presutti, F. Draicchio, A. Gangemi, Knowledge Extraction Based on Discourse Representation Theory and Linguistic Frames, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 114–129. doi:10.1007/978-3-642-33876-2_12.

[26] V. Loia, W. Pedrycz, S. Senatore, M. I. Sessa, Web navigation support by means of proximity-driven assistant agents, Journal of the Association for Information Science and Technology 57 (4) (2006) 515–527. doi:10.1002/asi.20306.
URL http://dx.doi.org/10.1002/asi.20306

[27] F. Zhao, Z. Sun, H. Jin, Topic-centric and semantic-aware retrieval system for internet of things, Information Fusion 23 (2015) 33 – 42. doi:http://dx.doi.org/10.1016/j.inffus.2014.01.001.

[28] J. Bhogal, A. Macfarlane, P. Smith, A review of ontology based query expansion, Information Processing & Management 43 (4) (2007) 866 – 886. doi:http://dx.doi.org/10.1016/j.ipm.2006.09.003.

[29] P. Cimiano, S. Handschuh, S. Staab, Towards the self-annotating web, in: Proceedings of the 13th International Conference on World Wide Web, WWW '04, ACM, New York, NY, USA, 2004, pp. 462–471. doi:10.1145/988672.988735.
URL http://doi.acm.org/10.1145/988672.988735

[30] C. De Maio, G. Fenza, M. Gallo, V. Loia, S. Senatore, Formal and relational concept analysis for fuzzy-based automatic semantic annotation, Applied Intelligence 40 (1) (2014) 154–177. doi:10.1007/s10489-013-0451-7.
URL http://dx.doi.org/10.1007/s10489-013-0451-7

[31] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, F. Ciravegna, Semantic annotation for knowledge management: Requirements and a survey of the state of the art, Web Semant. 4 (1) (2006) 14–28. doi:10.1016/j.websem.2005.10.002.
URL http://dx.doi.org/10.1016/j.websem.2005.10.002

[32] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, M. Goranov, KIM – Semantic Annotation Platform, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 834–849. doi:10.1007/978-3-540-39718-2_53.

[33] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, K. S. McCurley, S. Rajagopalan, A. Tomkins, J. A. Tomlin, J. Y. Zien, A case for automated large-scale semantic annotation, Web Semantics: Science, Services and Agents on the World Wide Web 1 (1) (2003) 115 – 132. doi:http://dx.doi.org/10.1016/j.websem.2003.07.006.

[34] F. Ciravegna, S. Chapman, A. Dingli, Y. Wilks, Learning to harvest information for the semantic web, in: C. Bussler, J. Davies, D. Fensel, R. Studer (Eds.), The Semantic Web: Research and Applications, Vol. 3053 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 312–326. doi:10.1007/978 − 3 − 540 − 25956 − $5_2$2.

[35] N. Kiyavitskaya, N. Zeni, J. R. Cordy, L. Mich, J. Mylopoulos, Cerno: Light-weight tool support for semantic annotation of textual documents, Data & Knowledge Engineering 68 (12) (2009) 1470 – 1492, including Special Section: 21st {IEEE} International Symposium on Computer-Based Medical Systems (IEEE {CBMS} 2008) Seven selected and extended papers on Biomedical Data Mining. doi:http://dx.doi.org/10.1016/j.datak.2009.07.012.

[36] V. Tablan, K. Bontcheva, I. Roberts, H. Cunningham, Mmir: An open-source semantic search framework for interactive information seeking and discovery, Web Semantics: Science, Services and Agents

on the World Wide Web 30 (2015) 52 – 68, semantic Search. doi:http://dx.doi.org/10.1016/j.websem.2014.10.002.

[37] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, the Journal of machine Learning research 3 (2003) 993–1022.

[38] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, R. Harshman, Using latent semantic analysis to improve access to textual information, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '88, ACM, New York, NY, USA, 1988, pp. 281–285. doi:10.1145/57167.57214.
URL http://doi.acm.org/10.1145/57167.57214

[39] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science 41 (6) (1990) 391–407.

[40] M. W. Berry, S. T. Dumais, G. W. O'Brien, Using linear algebra for intelligent information retrieval, SIAM review 37 (4) (1995) 573–595.

[41] J. H. Lau, D. Newman, S. Karimi, T. Baldwin, Best topic word selection for topic labelling, in: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 605–613.
URL http://dl.acm.org/citation.cfm?id=1944566.1944635

[42] Y. Zhuang, H. Gao, F. Wu, S. Tang, Y. Zhang, Z. Zhang, Probabilistic word selection via topic modeling, IEEE Transactions on Knowledge and Data Engineering 27 (6) (2015) 1643–1655. doi:10.1109/TKDE.2014.2377727.

[43] T. Sabbah, A. Selamat, M. Ashraf, T. Herawan, Effect of thesaurus size on schema matching quality, Knowledge-Based Systems 71 (2014) 211 – 226. doi:http://dx.doi.org/10.1016/j.knosys.2014.08.002.

[44] M. D. Marneffe, C. D. Manning, Stanford typed dependencies manual (2008).

[45] R. A. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[46] K. Stevens, P. Kegelmeyer, D. Andrzejewski, D. Buttler, Exploring topic coherence over many models and many topics, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL '12, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012, pp. 952–961.
URL http://dl.acm.org/citation.cfm?id=2390948.2391052

[47] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, A. McCallum, Optimizing semantic coherence in topic models, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 262–272.
URL http://dl.acm.org/citation.cfm?id=2145432.2145462

[48] Y. Yang, An evaluation of statistical approaches to text categorization, Information retrieval 1 (1-2) (1999) 69–90.

[49] J. J. Rocchio, Evaluation viewpoint in document retrieval, Information Storage and Retrieval, Report ISR-9, to the National Science Foundation, Section XXI, Harvard Computation Laboratory.

[50] F. A. González, J. C. Caicedo, Quantum latent semantic analysis, in: Advances in Information Retrieval Theory, Springer, 2011, pp. 52–63.

[51] L. Reeve, H. Han, Survey of semantic annotation platforms, in: Proceedings of the 2005 ACM Symposium on Applied Computing, SAC '05, ACM, New York, NY, USA, 2005, pp. 1634–1638. doi:10.1145/1066677.1067049.
URL http://doi.acm.org/10.1145/1066677.1067049

[52] R. Navigli, S. P. Ponzetto, BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network, Artificial Intelligence 193 (2012) 217–250.