International Journal of Semantic Computing
Vol. 5, No. 4 (2011) 433–462

DOI: 10.1142/S1793351X1100133X

World Scientific
www.worldscientific.com

A DISTRIBUTIONAL STRUCTURED SEMANTIC SPACE FOR QUERYING RDF GRAPH DATA

ANDRÉ FREITAS*, EDWARD CURRY†, JOÃO GABRIEL OLIVEIRA‡ and SEÁN O'RIAIN§

Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway
IDA Business Park, Lower Dangan, Galway, Ireland
**andre.freitas@deri.org*
†*ed.curry@deri.org*
‡*joao.deoliveira@deri.org*
§*sean.oriain@deri.org*

The vision of creating a Linked Data Web brings together the challenge of allowing queries across highly heterogeneous and distributed datasets. In order to query Linked Data on the Web today, end users need to be aware of which datasets potentially contain the data and also which data model describes these datasets. The process of allowing users to expressively query relationships in RDF while abstracting them from the underlying data model represents a fundamental problem for Web-scale Linked Data consumption. This article introduces a *distributional structured semantic space* which enables data model independent natural language queries over RDF data. The center of the approach relies on the use of a distributional semantic model to address the level of semantic interpretation demanded to build the data model independent approach. The article analyzes the geometric aspects of the proposed space, providing its description as a distributional structured vector space, which is built upon the Generalized Vector Space Model (GVSM). The final semantic space proved to be flexible and precise under real-world query conditions achieving *mean reciprocal rank* = 0.516, *avg. precision* = 0.482 and *avg. recall* = 0.491.

Keywords: Linked data queries; semantic search; distributional semantics; semantic web; linked data.

1. Introduction

The vision behind the construction of a Linked Data Web [1] where it is possible to consume, publish, and reuse data at Web scale steps into a fundamental problem in the databases space. In order to query highly heterogeneous and distributed data at Web-scale, it is necessary to reformulate the current paradigm on which users interact with datasets. Current query mechanisms are highly dependent on an *a priori* understanding of the data model behind the datasets. Users querying Linked Datasets today need to articulate their information needs in a query containing explicit representations of the relationships in the data model (i.e. the dataset 'vocabulary'). This query paradigm is deeply attached to the traditional perspective of structured queries

over databases. This query model does not suit the heterogeneity, distributiveness, and scale of the Web, where it is impractical for data consumers to have a previous understanding of the structure and location of available datasets.

Behind this problem resides a fundamental limitation of current information systems to provide a semantic interpretation approach that could bridge the semantic gap between users' information needs and the 'vocabulary' used to describe systems' objects and actions. This *semantic gap*, defined by Furnas *et al.* [6] as the *vocabulary problem in human-system communication*, is associated to the dependency on human language (and its intrinsic variability) in the construction of systems and information artifacts. At Web-scale, the vocabulary problem for querying existing Linked Data represents a fundamental barrier, which ultimately limits the utility of Linked Data for data consumers.

For many years the level of semantic interpretation needed to address the vocabulary problem was associated with deep problems in the Artificial Intelligence space, such as knowledge representation and commonsense reasoning. However, the solution to these problems also depends upon some prior level of semantic interpretation, creating a self-referential dependency. More recently, promising results related to research on *distributional semantics* [7, 9] are showing a possible direction to solve this conundrum by bootstrapping on the knowledge present in large volumes of Web corpora.

This work proposes a distributional structured semantic space focused on providing a data model independent query approach over RDF data. The semantic space introduced in this paper builds upon the *Treo* query mechanism, introduced in [8]. The center of the approach relies on the use of distributional semantics together with a hybrid search strategy (entity-centric search and spreading activation search) to build the semantic space. The proposed approach refines the previous Treo query mechanism, introducing a new entity search strategy and structured vector space model based on distributional semantics. The construction of an index from the elements present on the original Treo query mechanism also targets the improvement of the scalability of the approach. The final semantic space, named *T-Space* (tau space), proved to be flexible and precise under real-world query conditions. This article extends the original discussion of the T-Space presented in [28], providing a more comprehensive description and analysis of the T-Space.

The construction of a semantic space based on the principles behind Treo (discussed in Sec. 3) defines a search/index generalization which can be applied to different problem spaces, where data is represented as labelled data graphs, including graph databases and semantic-level representations of unstructured text.

The paper is organized as follows: Sec. 2 introduces the central concepts of distributional semantics and semantic relatedness measures describing one specific distributional approach, Explicit Semantic Analysis (ESA); Sec. 3 covers the basic principles behind the query processing approach; Sec. 4 describes the construction of the distributional structured semantic space; Sec. 5 formalizes and analyzes the geometric aspects of the proposed approach; Sec. 6 covers the

evaluation of the approach; Section 7 describes related work and Sec. 8 provides conclusion and future work.

2. Distributional Semantics

2.1. *Motivation*

Distributional semantics is built upon the assumption that the context surrounding a given word in a text provides important information about its meaning [9]. A rephrasing of the *distributional hypothesis* states that words that occur in similar contexts tend to have similar meaning [9]. Distributional semantics focuses on the construction of a semantic representation of a word based on the statistical distribution of word co-occurrence in texts. The availability of high volume and comprehensive Web corpora brought distributional semantic models as a promising approach to build and represent meaning. Distributional semantic models are naturally represented by Vector Space Models, where the meaning of a word is represented by a weighted concept vector.

However, the proper use of the simplified model of meaning provided by distributional semantics implies understanding its characteristics and limitations. As Sahlgren [7] notes, the distributional view on meaning is non-referential (does not refer to extra-linguistic representations of the object related to the word), being inherently differential: the differences of meaning are mediated by differences of distribution. As a consequence, distributional semantic models allow the quantification of the amount of difference in meaning between linguistic entities. This differential analysis can be used to determine the semantic relatedness between words [7]. Therefore, the applications of the meaning defined by distributional semantics should focus on a problem space where its differential nature is suitable. The computation of semantic relatedness and similarity measures between pairs of words is one instance in which the strength of distributional models and methods is empirically supported [5]. This work focuses on the use of distributional semantics in the computation of semantic relatedness measures as a key element to address the level of semantic flexibility necessary for the provision of data model independent queries over RDF data. In addition, the differential nature of distributional semantics also fits into a *semantic best-effort/approximate ranked results* query strategy which is the focus of this work.

2.2. *Semantic relatedness*

The concept of *semantic relatedness* is described [10] as a generalization of *semantic similarity*, where semantic similarity is associated with taxonomic relations between concepts (e.g. *car* and *airplane* share *vehicle* as a common taxonomic ancestor) and semantic relatedness covers a broader range of semantic relations (e.g. *car* and *driver*). Since the problem of matching natural language terms to concepts present in datasets can easily cross taxonomic boundaries, the generic concept of semantic relatedness is more suitable to the task of semantic matching for queries over the RDF data.

Until recently WordNet, an interlinked lexical database, was the main resource used in the computation of similarity and relatedness measures. The limitations of the representation present in WordNet include the lack of a rich representation of non-taxonomic relations (fundamental for the computation of relatedness measures) and a limitated number of modeled concepts. These limitations motivated the construction of approaches based on distributional semantics. The availability of large amounts of unstructured text on the Web and, in particular, the availability of Wikipedia, a comprehensive and high-quality knowledge base, motivated the creation of relatedness measures based on Web resources. These measures focus on addressing the limitations of WordNet-based approaches by trading structure for volume of commonsense knowledge [5]. Comparative evaluations between WordNet-based and distributional approaches for the computation of relatedness measures have shown the strength of the distributional model, reaching a high correlation level with human assessments [5].

2.3. *Explicit semantic analysis*

The distributional approach used in this work is given by the Explicit Semantic Analysis (ESA) semantic space [5], which is built using Wikipedia as a corpus. The ESA space provides a distributional model which can be used to compute an explicit semantic interpretation of a term as a set of weighted concepts. In the case of ESA, the set of returned weighted concept vectors associated with the term is represented by the titles of Wikipedia articles. A *universal ESA space* is created by building a vector space containing Wikipedia articles' document representations using the traditional TF/IDF weighting scheme. In this space, each article is represented as a vector where each component is a weighted term present in the article. Once the space is built, a keyword query over the ESA space returns a list of ranked articles titles, which define a concept vector associated with the query terms (where each vector component receives a relevance weight). The approach also allows the interpretation of text fragments, where the final concept is the centroid of the vectors representing the set of individual terms. This procedure allows the approach to partially perform word sense disambiguation [5]. The ESA semantic relatedness measure between two terms or text fragments is calculated by comparing the concept vectors representing the interpretation of the two terms or text fragments. The use of the ESA distributional approach in the construction of the proposed semantic space is covered in the next three sections.

3. Query Approach

3.1. *Motivation*

The distributional structured semantic space introduced in this paper generalizes and improves the approach used in the *Treo* query mechanism [8]. The construction

of a semantic space, based on the principles behind Treo, defines a structured vector space generalization which can be applied into different problem spaces, where data is represented as a labelled graph, such as RDF/Linked Data, graph databases and semantic-level representation of unstructured text. This section first introduces the strategies and principles behind the Treo query approach, followed by an instantiation of the search model for an exemplar natural language query.

The characteristics of the query approach merges elements from both the Information Retrieval (IR) and from the Database perspectives. In the proposed query model, users are allowed to input queries referring to structures and relations present in the data (database perspective) while a ranked list of results is expected (IR perspective). Additionally, since the proposed approach is formulated using elements from IR (such as a Vector Space Model), many operations involved in the query processing are mapped to search operations. These two perspectives are reflected in the discourse of this work.

3.2. *Principles behind the query approach*

In order to build the data model independent query mechanism, five main guiding principles are employed:

(1) *Approximate query model*: The proposed approach targets an approximate solution for queries over Linked datasets. Instead of expecting the query mechanism to return exact results as in structured SPARQL queries, it returns a semantically approximate and ranked answer set which can be later cognitively assessed by human users. An explicit requirement in the construction of an approximate approach for queries over structured data is the conciseness of the answer set, where a more selective cut-off function is defined, instead of an exhaustive ranked list of results (as in most document search engines).
(2) *Use of semantic relatedness measures to match query terms to dataset terms*: Semantic relatedness and similarity measures allow the computation of a measure of semantic proximity between two natural language terms. The measure allows query terms to be semantically matched to dataset terms by their level of semantic relatedness. While semantic similarity measures are constrained to the detection of a reduced class of semantic relations, and are mostly restricted to compute the similarity between terms which are nouns, semantic relatedness measures are generalized to any kind of semantic relation. This makes them more robust to the heterogeneity of the vocabulary problem at Web-scale.
(3) *Use of a distributional semantic relatedness measure built from Wikipedia*: Distributional relatedness measures are built using comprehensive knowledge bases on the Web, by taking into account the distributional statistics of a term, i.e. the co-occurrence of terms in its surrounding context. The use of comprehensive

knowledge sources allows the creation of a high coverage distributional semantic model.

(4) *Compositionality given by query dependency structure and data* (*s, p, o*) *structure*: The approach builds upon the concept of using *Partial Ordered Dependency Structures* (PODS) as the query input. PODS are an intermediate form between a natural language query and a structured graph pattern that is built upon the concept of dependency grammars [11]. A dependency grammar is a syntactic formalism that has the property of abstracting over the surface word order, mirroring semantic relationships and creating an intermediate layer between syntax and semantics [11]. The idea behind the PODS query representation is to maximize the matching probability between the natural language query and triple-like (subject, predicate and object) structure present in the dataset. Additional details are covered in [8].

(5) *Two phase search process combining entity search with spreading activation search*: The search process over the graph data is split into two phases. The first phase consists of searching in the datasets for instances or classes (*entity search*) which are expressed as terms in the query, defining *pivot entities* as entry points in the datasets for the semantic matching approach. The process is followed by a semantic matching phase using a *spreading activation search based on semantic relatedness*, which matches the remaining query terms. This separation allows the search space to be pruned in the first search step by the part of the query which has higher specificity (the key entity in the query), followed by a search process over the properties of the pivot entities (attributes and relations).

The next section details how the strategies described above are implemented in a query approach over RDF data.

3.3. *Query processing steps*

The query processing approach starts with the pre-processing of the user's natural language query into a partial ordered dependency structure (PODS), a format which is closer to the triple-like (subject, predicate, and object) structure of RDF. The construction of the PODS demands an *entity recognition step*, where key entities in the query are determined by the application of named entity recognition algorithms, complemented by a search over the lexicon defined by dataset instances and classes labels. This is followed by a *query parsing step*, where the partial ordered dependency structure is built by taking into account the dependency structure of the query, the position of the key entity and a set of transformation rules. An example of PODS for the example query '*From which university did the wife of Barack Obama graduate?*' is shown as gray nodes in Fig. 1. For additional details on the query preprocessing, including entity recognition and the query parsing steps, the reader is directed to [8].

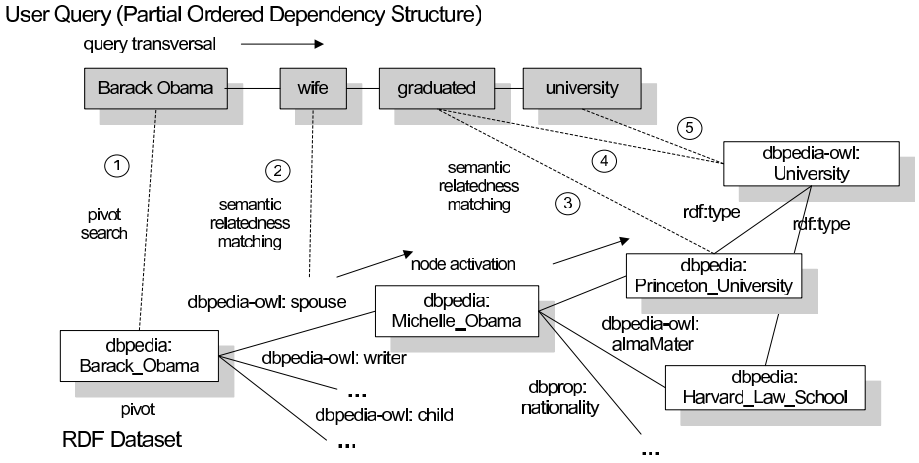

Fig. 1. The semantic relatedness based spreading activation search model for the example query.

The semantic search process takes as input the PODS representation of the query and consists of two steps:

(1) *Entity Search and Pivot Entity Determination*: The key entities in the PODS (which were detected in the entity recognition step) are sent to an entity-centric search engine, which maps the natural language terms for the key entities into dataset entities (represented by URIs). In the entity-centric search engine, instances are indexed using TF/IDF over labels extracted from URIs, while classes are indexed using the ESA semantic space for its associated terms (see Sec. 4). The URIs define the pivot entities in the datasets, which are the entry points for the semantic search process. In the example query, the term *Barack Obama* is mapped to the URI *http://dbpedia.org/resource/Barack_Obama* in the dataset.

(2) *Semantic Matching* (*Spreading Activation using Semantic Relatedness*): Taking as inputs the pivot entities URIs and the PODS query representation, the semantic matching process starts by fetching all the relations associated with the top ranked pivot entities. In the context of this work, the semantics of a relation associated with an entity is defined by taking into account the aggregation of the predicate, associated range types and object labels. Starting from the pivot node, the labels of each relation associated with the pivot node have their semantic relatedness measured against the next term in the PODS representation of the query. For the example entity *Barack Obama*, the next query term *wife* is compared against all predicates/range types/objects associated with each predicate (e.g. *spouse*, *child*, *religion*, etc.). The relations with the highest relatedness measures define the neighboring nodes which are explored in the search process. The search algorithm then navigates to the nodes with high

relatedness values (in the example, *Michelle Obama*), where the same process happens for the next query term (*graduate*). The search process continues until the end of the query is reached, working as a spreading activation search over the RDF graph, where the activation function (i.e. the threshold which determines the further node exploration process) is defined by a semantic relatedness measure.

The spreading activation algorithm returns a set of *triple paths*, which are a connected set of triples defined by the spreading activation search path, starting from the pivot entities over the RDF graph. The triple paths are merged into a final graph and a visualization is generated for the end user (see Fig. 5). The next section uses the elements of the described approach to build a distributional structured semantic space.

4. Distributional Structured Semantic Space

4.1. *Introduction*

The main elements of the approach described in the previous section are used in the construction of a distributional structured semantic space, named here a *T-Space* (tau-space). The final semantic space is targeted towards providing a vocabulary/data model independent semantic representation of RDF datasets. This work separates the discussion between the definition of the semantic space model and the actual implementation of its corresponding index. Despite the implementation of an experimental index for evaluation purposes, this article concentrates on the definition and description of the semantic space model.

The distributional semantic space is composed by an entity-centric space where *instances* define vectors over this space using the TF/IDF weighting scheme and where *classes* are defined over an ESA entity space (the construction of the ESA space is detailed later). The construction strategy for the *instance entity space* benefits a more rigid and less semantically flexible entity search for instances, where the expected search behavior is closer to a *string similarity matching* scenario. The rationale behind this indexing approach is that instances in RDF datasets usually represent named entities (e.g. names for people and places) and are less constrained by lexico-semantic variability issues in their dataset representation.

Classes demand a different entity indexing strategy and since they represent categories (e.g. yago:UnitedStatesSenators) they are more bound to a variability level in their representation (e.g. the class yago:UnitedStatesSenators could have been expressed as yago:AmericanSenators). In order to cope with this variability, the *entity space for classes* should have the property of semantically matching terms in the user queries to dataset terms. In the case of the class name *United States Senators* it is necessary to provide a semantic match with equivalent or related terms such as *American Senators* or *American Politicians*. The desired search behavior for a query in this space is to return a ranked list of semantically related

class terms, where the matching is done by providing a semantic space structure which allows search based on a semantic interpretation of query and dataset terms. The key element in the construction of the semantic interpretation model is the use of distributional semantics to represent query and dataset terms. Since the desired behavior for the semantic interpretation is of a semantic relatedness ranking approach, the use of distributional semantics is aligned with the differential meaning assumption (Sec. 2.2). The same distributional approach can be used for indexing *entity relations* which, in the scope of this work, consists of both terminology-level (properties, ranges, and associated types) and instance-level object data present in the set of relations associated with an entity.

4.2. *Building the T-Space*

The steps in the construction of the distributional structured semantic space (T-Space) are:

(1) *Construction of the Universal Explicit Semantic Analysis* (*ESA*) *Space*: The distributional structured semantic space construction starts by creating a *universal Explicit Semantic Analysis* (*ESA*) *space* (step 1, Fig. 3). A *universal ESA space* is created by indexing Wikipedia articles using the TF/IDF vector space approach. Once the space is built, a keyword query over the ESA space returns a set of ranked articles titles which defines a concept vector associated with query terms (where each component of this vector is a Wikipedia article title receiving a relevance score). Figure 2 depicts two ESA interpretation vectors. The concept vector is called the *semantic interpretation* of the term and can be used as its semantic representation.

(2) *Construction of the Entity Space* (*Instances and Classes*): As previously mentioned, instances in the graph are indexed by calculating the TF/IDF score over

ESA interpretation vector

United States Senators From Illinois

```
Governor of Illinois (0.1219)
Senate  (0.10695)
History of Illinois (0.09895)
Western Illinois University
(0.09180)
Springfield, Illinois (0.09122)
University of Illinois  (0.09106)
Normal, Illinois (0.09028)
Illinois Central Railroad (0.08792)
Illinois (0.08704)
Peter Fitzgerald (0.08685)
Cairo, Illinois (0.08670)
David Davis (senator) (0.08596)
Carbondale, Illinois (0.08548)
Illinois State University (0.084508)
...
```

spouse

```
Spouses of the Prime Ministers
of Canada (0.6558)
Adultery (0.4153)
Widow (0.4095)
Alimony (0.3751)
Spousal abuse (0.3467)
Domestic partnership (0.3292)
Rights and responsibilities of
marriages in the United States
(0.3258)
First Lady (0.3206)
Common-law marriage (0.2919)
Family  (0.27550)
Princess consort (0.2705)
Divorce (0.2383)
...
```

Fig. 2. Examples of ESA interpretation vectors for *United States Senators from Illinois* and *spouse*.

the labels of the instances (step 2, Fig. 3). The ESA universal space is used to generate the *class space*. The construction of the ESA semantic vector space is done by taking the interpretation vectors for each graph element label and by creating a vector space where each dimension of the coordinate basis of the space is defined by a concept component present in the interpretation vectors. The dimensions of the class space correspond to the set of distinct concepts returned by the interpretation vectors associated with the terms which describe the classes. Each class can then be mapped to a vector in this vector space (the associated score for each component is given by the TF/IDF scores associated with each interpretation component). This space has the desired property of returning a list of semantically related terms for a query (ordered from the most to the less semantically related). This procedure is described in the step 3 of Fig. 3 for the construction of the class entity space. The *final entity space* can be visualized as space with a double coordinate basis where instances are defined using a *TF/IDF term basis* and classes with an *ESA concept basis* (Fig. 3).

(3) *Construction of the Relation Spaces*: Once the entity space is built, it is possible to assign for each point defined in the entity vector space, a linear vector space which represents the relations associated with each entity. For the example instance *Barack Obama*, a relation is defined by the set of properties, types and objects which are associated with this entity in its RDF representation. The procedure for building the relation spaces is similar to the construction of the class space, where the terms present in the relations (properties, range, types and objects) are used to create a linear vector space associated with the entity. One property of *entity relation spaces* is the fact that each space has an independent number of dimensions, being scoped to the number of relations specific for each entity (step 4, Fig. 3).

4.3. *T-Space structure*

The use of an orthogonal coordinate basis to depict the instance, class and relation spaces in Fig. 3 has the purpose of simplifying the understanding of the figure. The coordinate basis for these spaces follows a Generalized Vector Space Model (GVSM), where there is no orthogonality assumption.

At this point the *T-Space* has the topological structure of two linear vector spaces ($E_I^{TF/IDF}$ and E_C^{ESA}) defined for the *instances* and *classes* respectively. Each point over these spaces defined by an entity vector has an associated *vector bundle* $R^{ESA}(E)$ which is the space of relations. The relations' spaces, however, have a variable number of dimensions and a different coordinate basis. Despite the fact that this topological model of the T-Space can be easily mapped to an inverted index structure, it can introduce unnecessary complexity to its mathematical model. Section 5 provides a simplification of this model, translating and formalizing the T-Space to a Generalized Vector Space Model (GVSM).

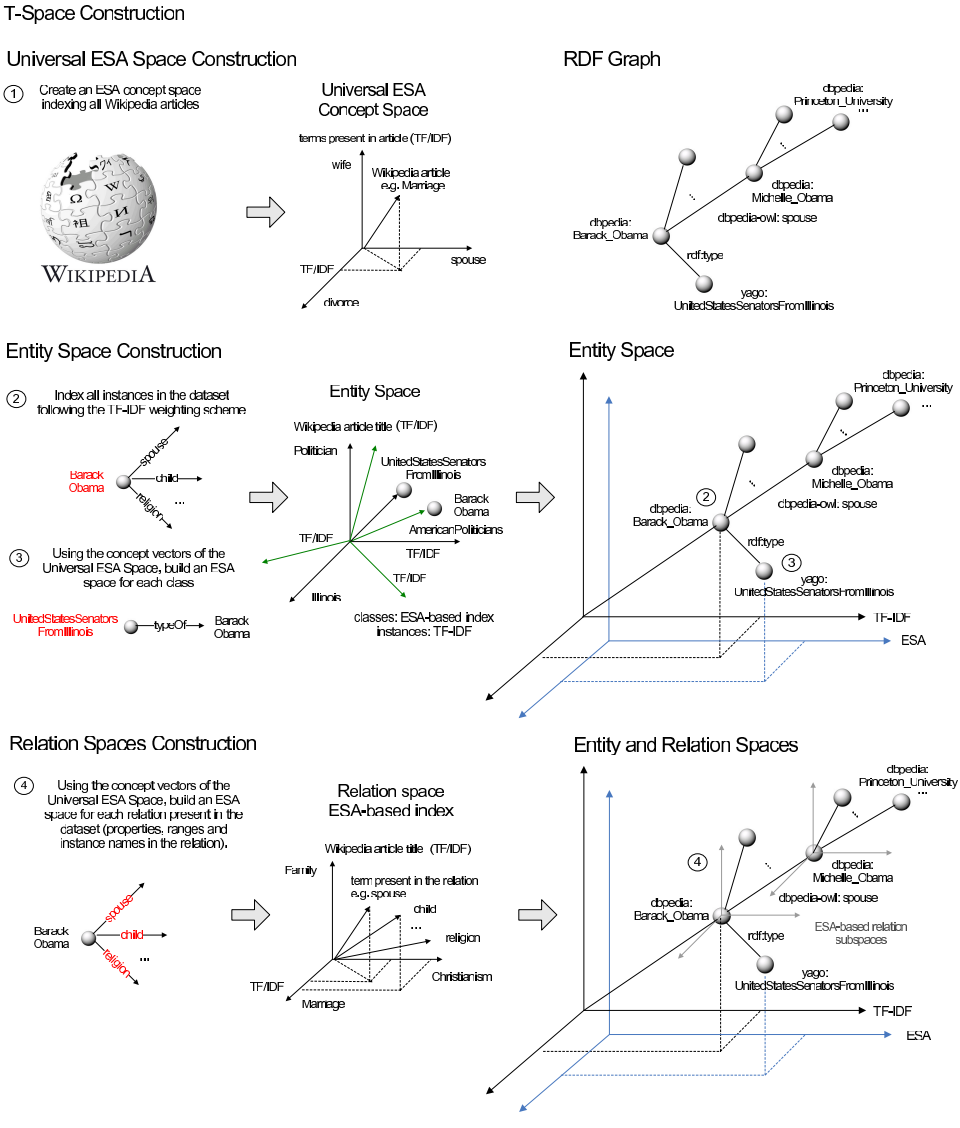

Fig. 3. Construction of the base spaces and of the final distributional structured semantic space (T-Space).

4.4. *Querying the T-Space*

With the final T-Space built, it is necessary to define the search procedure over the space. The query input is a partial ordered dependency structure (PODS) with the key query entity defined (Fig. 4). The key query entity is the first term to be searched on the entity space (it is searched in the instances entity space in case it is a named entity; otherwise it is searched over the class space). The entity search

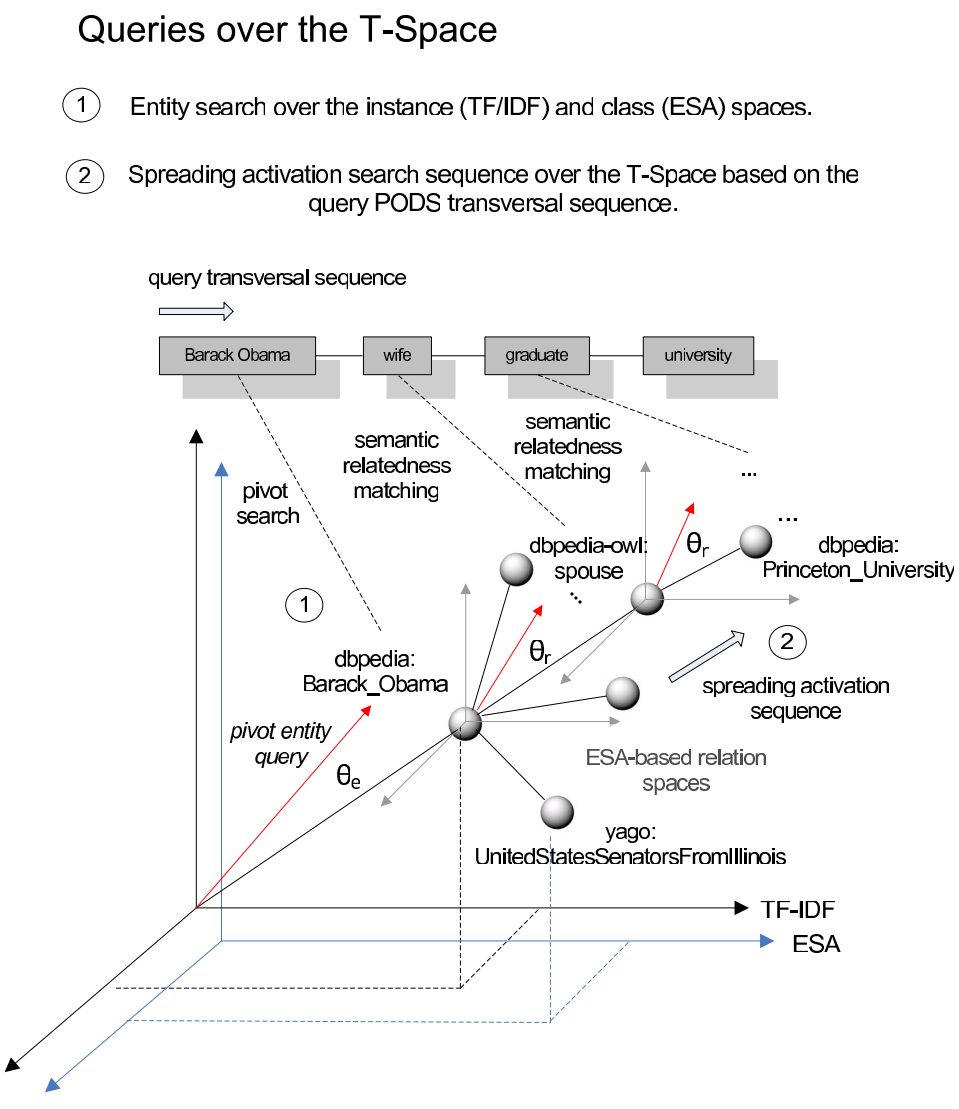

Fig. 4. Querying the T-Space using the example query.

operation is defined by the cosine similarity between the query vector and the entities vectors. For queries over the ESA entity space, the ESA interpretation vector for the query is defined using the *Universal ESA space*. The return of the query is a set of URIs mapping to entities in the space (e.g. dbpedia:Barack_Obama in the example). After, the next term of the PODS structure sequence is taken ('wife') and it is used to query each relation space associated with the set of entities (cosine similarity of the interpretation vector of the query term and the relation vectors in the space). The set of relations with high relatedness scores is used to

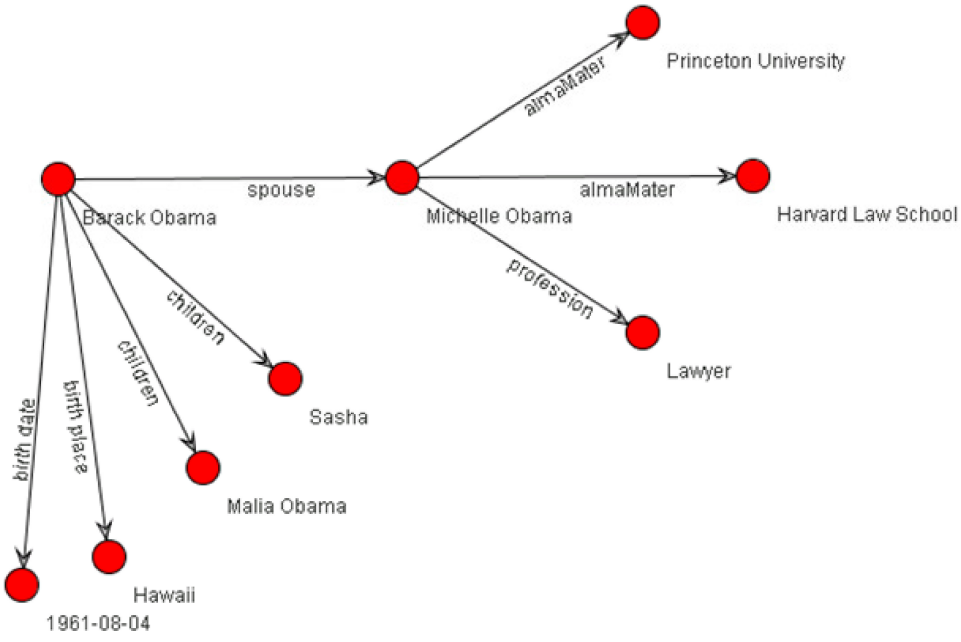

Fig. 5. Screenshot of the returned graph for the implemented prototype for the example query.

activate other entities in the space (e.g. dbpedia:Michelle_Obama). The same process follows for the activated entities until the end of the query is reached. The search process returns a set of ranked triple paths where the rank score of each triple path is defined by the average of the relatedness measures. Figure 5 contains a set of merged triple paths for the example query.

In the node selection process, nodes above a relatedness score threshold determine the entities which will be activated. The activation function is given by an adaptive discriminative relatedness threshold which is defined based on the set of returned relatedness scores. The adaptive threshold has the objective of selecting the relatedness scores with higher discrimination. Additional details on the threshold function are available in [8]. A more recent investigation on the use of ESA semantic relatedness as a ranking function and a better semantic threshold function for ESA can be found in [22].

4.5. *Analysis*

The approximative nature of the approach allows the improvement of *semantic tractability* [17] by returning an answer set which users can quickly assess to determine the final answer to their information needs. The concept of semantic tractability in natural language queries over databases can be described as the mapping between the terms and syntactic structure of a query to the lexicon and data model structure of a database. Typically, semantically tractable queries are

queries which can be directly mapped to database structures, and the improvement of semantic tractability of queries have been associated with difficult problems such as commonsense reasoning (the concept of semantic tractability is a rephrasing of the vocabulary problem for natural language interfaces to databases). As an example consider the query '*Is Albert Einstein a PhD?*'. In the current version of DBPedia there is no explicit statement containing this information. However, the proposed approach returns an answer set containing the relation '*Albert Einstein doctoral-Advisor Alfred Kleiner*' from which users can quickly derive the final answer. Differently from Question Answering systems which aims towards a precise answer to the user information needs (in this case 'Yes/No'), the proposed approach uses the semantic knowledge embedded on the distributional model to expose the supporting information, delegating part of the answer determination process to the end user. The approach, however, improves the semantic tractability of the queries by finding answers which support the query.

The final distributional structured semantic space unifies into a single approach important features which are emerging as trends in the construction of new semantic and vector space models. The first feature is related to the adoption of a *distributional model of meaning* in the process of building the semantic representation of the information. The second feature is the use of *third-party available Web corpora* in the construction of the distributional model, instead of just relying on the indexed information to build the distributional semantic base. The third important feature is the inclusion of a *compositional element in the definition of the data semantics*, where the structure given by the RDF graph and by the PODS are used to define the semantic interpretation of the query, together with the individual distributional meaning of each word.

5. Distributional Semantics and the Geometric Structure of the T-Space

5.1. *Motivation*

This section provides a formal description of the structure defined by the T-Space. A formal model of the T-Space is created based on the Generalized Vector Space Model (GVSM) for Explicit Semantic Analysis (ESA) [18– 20]. The analysis focuses on the description of a principled connection between the semantics of the T-Space and its geometric properties. The geometric properties which arise in the model can provide a principled way to model the semantics of RDF or, more generally, labelled data graphs, adding to the vector space model *structures* and *operations* which support an *approximate semantic matching*.

While the previous section covered the basic principles of the T-Space which can be used to build an inverted index, this section focuses on the description of the T-Space as a vector space model. The description of the T-Space in the previous section generates a complex topological model, due to the differences between the

nature of the coordinate systems and the dimensionality of the instance, entity and relation spaces. The T-Space, however, can be unified into a single coordinate system. The objective of this unified description is twofold: (i) the reduction of the T-Space to a mathematical model which can support the understanding of its properties and (ii) casting the T-Space into existing information retrieval models.

The strategy for unifying the T-Space into a single coordinate system consists in using the connection between ESA and TF/IDF, where the distributional reference frame (defined by the ESA concept vectors) can be defined from the TF/IDF term space. This allows the unification of the instance, class and relation spaces into a base TF/IDF coordinate system. In the unified space, relations between entities are defined by the introduction of a vector field over each point defined by an entity. The vector field, defined over the ESA distributional reference frame, preserves the RDF graph structure, while the distributional reference frame allows a semantic matching over this structure.

This section is organized as follows: Sec. 5.2 introduces a formalization for the ESA model based on a Generalized Vector Space Model which serves as the basis for the construction of the space; Sec. 5.3 builds the geometric model behind the T-Space; Sec. 5.4 defines operations over the T-Space and Sec. 5.5 discusses the implications of the geometric model of meaning supported by the T-Space.

5.2. *Generalized vector space model for ESA*

This work uses the formalization of ESA introduced in [20] and [19]. Anderka and Stein [20] describe the ESA model using the Generalized Vector Space Model (GVSM). In the GVSM model, Wong *et al.* [18] propose an interpretation of the term vectors present on the index as linearly independent but not pairwise orthogonal. Anderka and Stein also analyzes the properties of ESA which affects its retrieval performance and introduce a formalization of the approach. Gottron *et al.* [19] proposes a probabilistic model for Explicit Semantic Analysis (ESA), using this model to provide deeper insights into ESA. The following set of definitions adapted from [18–20] and [23] are used to build the structure of the T-Space.

Definition 1. Let $K = k_1, \ldots, k_T$ be the set of all terms available in a document collection (index terms). Let $w_{i,j} > 0$ be a weight associated with each term k_i contained in a document d_j (pair $[k_i, d_j]$), where $j = 1, \ldots, N$. For a k_i term not contained in a document d_j, $w_{i,j} = 0$. A document d and a query q are represented as weighted vectors $\mathbf{d_j} = w_{1,j}, w_{2,j}, \ldots, w_{T,N}$ and $\mathbf{q} = q_1, q_2, \ldots, q_M$ in a t-dimensional space.

The set of k_i terms defines a unitary coordinate basis for the vector space. Representing the document in relation to the set of basis term vectors:

$$\mathbf{d}_j = \sum_{i=1}^{T} w_{i,j}\mathbf{k_i}, \quad (j = 1, \ldots, N) \tag{1}$$

and the query:

$$\mathbf{q} = \sum_{i=1}^{T} q_i \mathbf{k_i}. \tag{2}$$

Definition 2. Let $freq_{i,j}$ be the frequency of term k_i in the document $\mathbf{d_j}$. Let $count(\mathbf{d_j})$ be the number of terms inside the document $\mathbf{d_j}$. The normalized term frequency $tf_{i,j}$ is given by:

$$tf_{i,j} = \frac{freq_{i,j}}{count(\mathbf{d_j})}. \tag{3}$$

Definition 3. Let n_{k_i} be the number of documents containing the term k_i and N the total number of documents. The *inverse document frequency* for the term k_i is given by:

$$idf_i = \log \frac{N}{n_{k_i}}. \tag{4}$$

Definition 4. The final TF/IDF weight value based on the values of *tf* and *idf* is defined as:

$$w_{i,j} = tf_{i,j} \times \log \frac{N}{n_{k_i}} \tag{5}$$

where the weight given by TF/IDF provides a measure on how a term is discriminative in relation to the relative distribution of other terms in the document collection.

The process of searching a document for a query $\mathbf{q}$ consists in computing the similarity between $\mathbf{q}$ and $\mathbf{d_j}$ which is given by the inner product between the two vectors:

$$sim_{\mathrm{VSM}}(\mathbf{q}, \mathbf{d_j}) = \langle \mathbf{q}, \mathbf{d_j} \rangle = \sum_{i=1}^{T} \sum_{l=1}^{T} w_{i,j} q_l \mathbf{k_i} \cdot \mathbf{k_l}, \quad (j = 1, \ldots, N). \tag{6}$$

In the traditional VSM the term vectors have unit length and are orthogonal. Embedded in these conditions is the assumption that there is no interdependency between terms (non-correlated terms) in the corpus defined by the document collection [18]. The generalized vector space model (GVSM) takes into account term interdependency, generalizing the identity matrix which represents $\mathbf{k_i} \cdot \mathbf{k_l}$ into a matrix G with elements $g_{i,l}$. The similarity between two vectors $\mathbf{q}$ and $\mathbf{d}$ in the GVSM using the matrix notation (W is defined as the matrix $w_{i,j}$) is:

$$sim_{\mathrm{GVSM}}(\mathbf{q}, \mathbf{d}) = \mathbf{q} G W^T. \tag{7}$$

Definition 5. Let D' be a collection representing the set of documents where each document d'_i is a Wikipedia article with a vector representation defined by a TF/IDF weighting scheme in a GVSM space. Let d be an arbitrary document.

The representation of the document d in the ESA model is a concept vector $\mathbf{c}$ which is given by:

$$\mathbf{c} = \sum_{i=1}^{N} \langle \mathbf{d}'_{\mathbf{i}}, \mathbf{d} \rangle \tag{8}$$

where $\langle \mathbf{d}'_{\mathbf{i}}, \mathbf{d} \rangle$ defines the computation of similarity between $\mathbf{d}'_{\mathbf{i}}$ and $\mathbf{d}$.

In the ESA model the similarity between two documents d_a and d_b is given by the inner product between their associated concept vectors c_a and c_b:

$$sim_{\text{ESA}}(\mathbf{d_a}, \mathbf{d_b}) = \cos(\mathbf{c_a}, \mathbf{c_b}) = \frac{\langle \mathbf{c_a}, \mathbf{c_b} \rangle}{|\mathbf{c_a}|, |\mathbf{c_b}|}. \tag{9}$$

$$sim_{\text{ESA}}(\mathbf{d_a}, \mathbf{d_b}) = \frac{1}{|\mathbf{c_a}|, |\mathbf{c_b}|} \sum_{i=1}^{m} \sum_{j=1}^{m} w_{a,j} w_{b,j} g_{i,j}. \tag{10}$$

For a set of documents D ($d_i \in D$) it is possible to build a vector space spanned by the set of ESA concept vectors associated with each document, where the concept vectors define the coordinate basis for the vector space.

$$\mathbf{d}_j = \sum_{i=1}^{T} v_{j,i} \mathbf{c_i}, \quad (j = 1, \ldots, N). \tag{11}$$

A query in this vector space also needs to be formulated in relation to its associated concept vectors. Alternatively it is possible to reformulate the coordinate basis to the original term coordinate basis. Using the Einstein summation convention, a document have its associated concept vector:

$$\mathbf{d} = V^i \mathbf{c_i} \tag{12}$$

where the document vector can be transformed to the TF/IDF basis:

$$\mathbf{d} = W'^i \mathbf{k_i} \tag{13}$$

$$W'^i = \alpha_i^{i'} V^i \tag{14}$$

where $\alpha_i^{i'}$ is a second-order transformation tensor which is defined by the set of TF/IDF vectors of ESA concepts. Figure 6(a) depicts the relation between the document vector $\mathbf{d}$ in relation to its concept basis $\mathbf{c}$ and term basis $\mathbf{k}$.

The distributional formulation of the vector space model supports the application of different distributional models (different corpora or metrics) to support the semantic interpretation of the document. A second-order tensor can be used to define the transportability between different distributional vector spaces (Figs. 6(a) and 6(b)).

5.3. *The structure of the T-Space*

The construction of the T-Space is targeted towards labelled data graphs. This work focuses on a model of graph defined by RDF. The Resource Description Framework

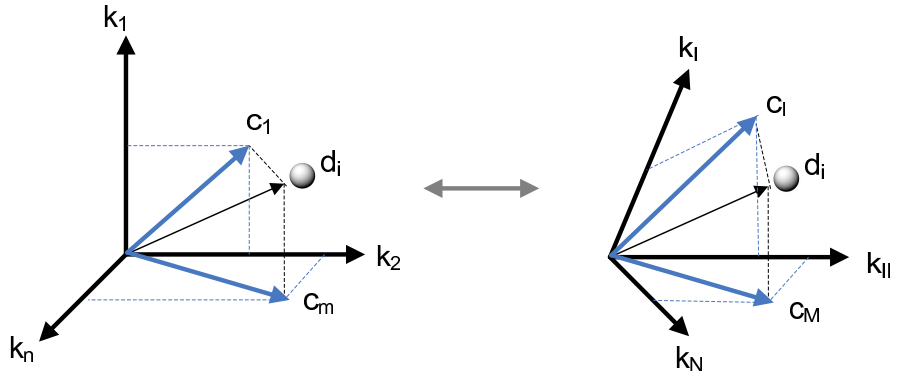

Fig. 6. Depiction of the relation between ESA and TF/IDF coordinate systems and transformation from different distributional models. The two coordinate systems represent different sets of terms, concepts and weights for the same resource in two different distributional models.

(RDF) provides a structured way of publishing information describing entities and its relations through the use of RDF terms and triples. RDF allows the definition of names for entities using URIs. RDF triples supports the grouping of entities into named classes, the definition of named relations between entities, and the definition of named attributes of entities using literal values. This section starts by providing a simple formal description of RDF. This description is used in the construction of the T-Space structure.

5.3.1. *RDF elements*

Definition 6 (RDF Triple). Let U be a finite set of URI resources, B a set of blank nodes and a L a finite set of literals. A triple $t = (s, p, o) \in (U \cup B) \times U \times (U \cup B \cup L)$ is an RDF triple where s is called the *subject*, p is called the *predicate* and o the *object*.

Definition 7 (RDF Graph). An RDF graph G is a subset of G, where $\mathrm{G} = (U \cup B) \times U \times (U \cup B \cup L)$.

RDF Schema (RDFS) is a semantic extension of RDF. By giving special meaning to the properties *rdfs:subClassOf*, *rdfs:subPropertyOf*, *rdfs:domain*, *rdfs:range*, *rdfs:Class*, *rdfs:Resource*, *rdfs:Literal*, *rdfs:Datatype*, etc., RDFS allows to express simple taxonomies and hierarchies among properties and resources, as well as domain and range restrictions for properties. The following definitions based on the notation of Eiter *et al.* [21] cover an incomplete description of specific RDFS aspects that are necessary to the description of the T-Space. A more complete formalization of the RDFS Semantics can be found in [21].

Definition 8 (Class). The set of classes C is a subset of the set of URIs U such that $\forall\, c \in C$:

$$\forall\, c(\mathit{triple}(c, \mathit{rdf} : \mathit{type}, \mathit{rdfs} : \mathit{Class})) \supset \mathit{triple}(c, \mathit{rdfs} : \mathit{subClassOf}, \mathit{rdfs} : \mathit{Resource}). \tag{15}$$

Definition 9 (Domain and Range). The *rdfs*:*domain* and *rdfs*:*range* of a property p in the triple t in relation to a class c are given by the following axioms:

$$\forall\, s, p, o, c(\mathit{triple}(s, p, o)) \wedge \mathit{triple}(p, \mathit{rdfs} : \mathit{domain}, c) \supset \mathit{triple}(s, \mathit{rdf} : \mathit{type}, c) \tag{16}$$

$$\forall\, s, p, o, c(\mathit{triple}(s, p, o)) \wedge \mathit{triple}(p, \mathit{rdfs} : \mathit{range}, c) \supset \mathit{triple}(o, \mathit{rdf} : \mathit{type}, c). \tag{17}$$

Definition 10 (Instances). The set of instances I is a subset of the set of URIs U such that $\forall\, i \in I$:

$$\forall\, i(\mathit{triple}(i, \mathit{rdf} : \mathit{type}, \mathit{rdfs} : \mathit{Class})) \supset \mathit{triple}(i, \mathit{rdf} : \mathit{type}, \mathit{rdfs} : \mathit{Resource}). \tag{18}$$

Definition 11 (Effective Range). An effective range $e \in E$ for a predicate p in a triple t is defined as the set of classes C associated as ranges of the corresponding predicate p and an instance i corresponding to the object of p.

Definition 12 (Relation). A relation r is given by a property p and its effective range e.

Every p, c, i and e has an *associated literal identifier* which is built by removing the namespace of the URI string, spliting the remaining string into separated terms.

The T-Space is built by embedding the set of *associated literal identifier* of instances, classes and relations into a ESA distributional vector space. Instances are resolved into the T-Space using the TF/IDF coordinate term basis, while classes and relations are resolved using the ESA coordinate concept basis, which can be transformed into the TF/IDF basis.

5.3.2. *Instances resolution*

Let I' be the set of literal identifiers associated with instances in an RDF graph G. The vector space $E^{TF/IDF}$ containing the embedding of the instances $\mathbf{i}'_a$ is built by the determination of the associated term vector $\mathbf{k_i}\, \forall\, i' \in I'$.

$$\mathbf{i}'_j = W^i_j \mathbf{k_i}, \quad (j = 1, \ldots, M) \tag{19}$$

where W^i is defined by the TF/IDF weighting scheme and M is the number of instances.

5.3.3. *Classes resolution*

Let C' be the set of literal identifiers associated with classes in an RDF graph G. The vector space E^{ESA} containing the embedding of the classes $\mathbf{c}'_a$ is built by determining the associated concept vector $\mathbf{c_j}$ from the terms $\mathbf{t}_u$ associated

with each $c' \in C'$.

$$\mathbf{c}'_j = V^i_j \mathbf{c_i}, \quad (j = 1, \ldots, N). \tag{20}$$

Alternatively the vectors in E^{ESA} can be mapped to the TF/IDF coordinate basis by the application of the following transformation:

$$\mathbf{c}'_j = \alpha^{i'}_i V^i_j \mathbf{k_i}, \quad (j = 1, \ldots, N) \tag{21}$$

where $\alpha^{i'}_i$ is a second-order transformation tensor which is defined by the set of TF/IDF term vectors of ESA concepts.

5.3.4. *Relations resolution*

Let R' be a set of literal identifiers r'_i for the relations associated with instances I' or classes C' in a RDF graph G. For all vectors $\mathbf{i}'^a$ and $\mathbf{c}'^b$ in $E^{TF/IDF}$, exists a vector field $r'^{(n)}(P)$, $\forall P \in \mathbb{R}^N$ and defined by $\mathbf{i}'^a$ and $\mathbf{c}'^b$, such that:

$$\mathbf{r}'^{(m)}(\mathbf{i}'^a) = \mathbf{i}'^a + U^{i(m)} \mathbf{k_i} \tag{22}$$

$$\mathbf{r}'^{(n)}(\mathbf{c}'^b) = \mathbf{c}'^b + V^{j(n)} \mathbf{c_j} \tag{23}$$

where U^i and V^j are the weights in relation to the term and concept components and m, n are indexes for the relation vectors. The set of vectors $r'^{(n)}(P)$ represent the distance to the neighboring graph nodes and can be grouped as a second-order tensor in relation to the concept coordinate basis. Figure 7 depicts the construction of the representation of relations from the elements in the data graph and the associated concept representation, while Fig. 8 shows the vector field structure of the T-Space defined by the relations.

5.4. *Operations over the T-Space*

5.4.1. *Input query*

An input query is given by three sets Q, C'^Q, I'^Q where Q is an ordered set representing the q_b query terms in a partial ordered dependency structure (PODS), I'^Q is

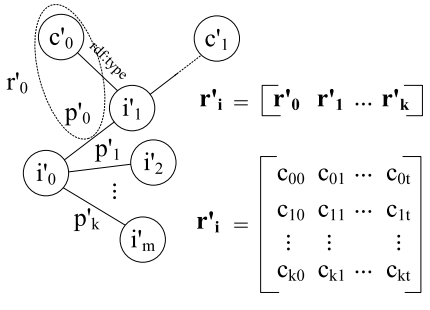

Fig. 7. Construction of the relation vectors associated with each instance or class.

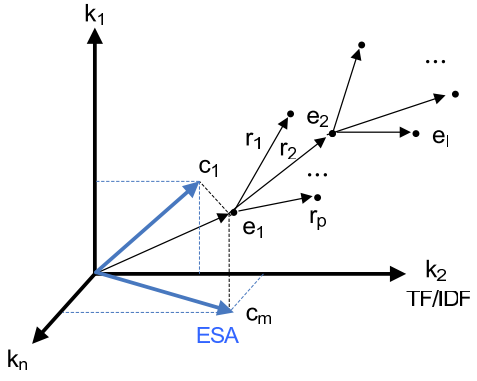

Fig. 8. Vector field representation for entities and relations.

a set of candidate instances' terms $i_b'^{Q}$ and C'^{Q} is a set of candidate classes' terms $c_a'^{Q}$ in the query, where $\forall\, c_a'^{Q}$ and $\forall\, i_b'^{Q}$ exists a corresponding $q_b \in Q$.

5.4.2. *Instance search*

Let $E^{TF/IDF}$ be the space containing instance vectors $\mathbf{i}'$. An instance query $q^{I'}$ is given by the $q_i^{I'}$ query terms in $(Q \cap I'^{Q})$. The instance search operation is defined by the computation of the cosine similarity $sim_{GVSM}(\mathbf{q}^{I'}, \mathbf{i}'_a)$ for each instance $q_i^{I'}$ and the instance vectors $\mathbf{i}'$ in $E^{TF/IDF}$.

5.4.3. *Class search*

Let E^{ESA} be the space containing class vectors $\mathbf{c}'$. A class query $q^{C'}$ is given by the $q_i^{C'}$ query terms in $(Q \cap C'^{Q})$. The class search operation is defined by the computation of the cosine similarity $sim_{ESA}(\mathbf{q}^{\mathbf{C}'}, \mathbf{c}'_a)$ between the ESA interpretation vector of each class query $q_i^{C'}$ and the class vectors $\mathbf{c}'$ in E^{ESA}, where references to the E^{ESA} can be transported to the $E^{TF/IDF}$ coordinate basis.

5.4.4. *Relation search*

Let E^{ESA} be a vector space containing the relation vector field $\mathbf{r}'^{(m)}(\mathbf{e}'^{a})$. A relation query $q^{R'}$ is given by the elements in the ordered set $(Q \backslash I'^{Q}) \cap (Q \backslash C'^{Q})$. The relation search operation is composed by the following operations:

(1) Determination of the concept vector $\mathbf{c}^{q}$ for the query $q^{R'}$:

$$\mathbf{c}^{q} = T^{i}\mathbf{c}_i \tag{24}$$

(2) Translation of $\mathbf{r}'^{(m)}(\mathbf{e}'_a)$, to the origin of the coordinate system:

$$\mathbf{r}''^{(m)}(\mathbf{e}'_a) = \mathbf{r}'^{(m)}(\mathbf{e}'_a) - \mathbf{e}'_a. \tag{25}$$

(3) Computation of the similarity $sim_{ESA}(\mathbf{c}^q, \mathbf{r}''^{(m)}(\mathbf{e}'_a))$ between $\mathbf{c}^q$ and each relation vector $\mathbf{r}''^{(m)}(\mathbf{e}'_a)$.
(4) Selection of a set of relation vectors $\mathbf{r}''_c$ through a threshold function $\mathbf{r}''_c = thr(sim_{ESA}(\mathbf{q}, \mathbf{r}''_b))$.

where references to the E^{ESA} can be transported to the $E^{TF/IDF}$ coordinate basis. Examples of threshold functions can be found in [8, 22].

5.4.5. *Spreading activation*

The spreading activation is defined by a sequence of translations in the $E^{TF/IDF}$ space which is determined the computation of the $i+1$ transversal iteration vector $\mathbf{r}''_c = thr(sim_{ESA}(\mathbf{q}_{i+1}, \mathbf{r}''_b))$.

5.5. *Analysis*

The proposed approach introduced in this work embeds an RDF graph into a vector space, adding geometry to the graph structure. The vector space is built from a distributional model, where the coordinate reference frame is defined by interpretation vectors mapping the statistical distribution of terms in the reference corpora. This distributional coordinate system supports a representation of the RDF graph elements which allows a flexible semantic search of these elements (differential aspect of distributional semantics). The distributional model enriches the original semantics of the topological relations and labels of the graph. The distributional model, collected from ustructured data, provides a supporting commonsense semantic reference frame which can be easily built from available text. The use of an external distributional data source which provides this semantic reference frame is a key difference between the T-Space and more traditional VSM approaches.

The additional level of structure is introduced as a vector field which is applied over points in the vector space, defined by vectors of instances and classes. Each vector in the vector field points to other instances and classes. The process of query answering through entity search and spreading activation, maps to a set of cosine similarity computations and translations over the vector field. The set of vectors associated with each point which is defined by an entity vector can also be modeled as a tensor field attached to each point (the set of vectors can be grouped into a second order tensor). The vector field nature of the objects in the T-Space is another difference in relation to traditional VSMs, allowing the preservation of the graph structure. Comparatively, traditional VSMs represent documents as (free) vectors at the origin of the vector space. The vector field connecting the entities in the graph, combined with the distributional reference frame and with the cosine similarity and translation operations, supports the trade-off between structure mapping (compositionality) and semantic flexibility.

Vector and tensor quantities can be represented in relation to a reference frame (coordinate system). Under this representation, however, changes of reference frame imply a change in the representation of the object. However, the transformation rules associated with the changes of reference frame are well defined objects and the representation of the object in a different reference frame can be recalculated. Tensors can be seen as geometric objects represented by numeric arrays that transform according to certain rules under a change of coordinates. This definition, which allows a flexible description in relation to coordinate systems supports a generalized description of geometric objects and spaces.

This capacity to transform objects in the T-Space across different coordinate systems can support the transportability across different distributional models. Data graphs from different domains can be supported by different distributional models, instead of a *one size fits all* solution. While an open domain data graph like DBPedia can be supported by a distributional model derived from Wikipedia, a domain specific data graph covering financial data can use a domain specific financial distributional model. Spaces with different distributional models can form patches in a more complex distributional manifold. Additionally, different distributional models can be used in parallel to support multiple interpretation of the elements embedded in the space. In case the concept vectors of multiple distributional models can be described in a common coordinate system, the parallel interpretation can be done by the transformation among the concept vectors, without the need to index the graph elements in both distributional models. Tensor calculus allows a unified scheme to model and formalize the transformation of vectors and tensors in the distributional space. In the basis of the tensor calculus lies the ability to transport the structure across different reference frames, allowing the application of different distributional models.

The proposed model for the T-Space coordinate system allows addressing the following challenges for Vector Space Models: (i) embedding the RDF graph structure, (ii) adding meaning from distributional models and (iii) supporting different models of meaning.

6. Evaluation

6.1. *Setup and analysis*

An experimental evaluation of the proposed semantic space was implemented to evaluate the answer quality of the approach using 50 natural language queries over DBPedia [2], defined in the QALD evaluation query set [3]. Since the final approach returns answers as triple-paths and considering that some queries require the application of post-processing operations (e.g. such as aggregation), a definition of a correct answer for the triple path format had to be generated. In the experimental set-up a correct answer is given by a triple path containing the URI supporting the final answer. For the example query '*How many films did Leonardo DiCaprio star in?*'

the triple paths containing the URIs for the films were considered as the correct answer instead of the number of movies.

For evaluation purposes the entity indexes corresponding to the class and instance entity spaces were generated for all DBPedia instances and classes. In order to simplify the experimental set-up, only relation vector spaces associated with entities which were effectively explored by the algorithm were generated, without any impact on the results reported on the evaluation of the approach. The distributional model was built from a 2006 version of Wikipedia. The final approach was able to answer 58% of the queries. The results were collected with a minimum level of post-processing. The final *mean reciprocal rank*, *avg. precision* and *avg. recall* are given in Table 1. The measurements for each query and the output data generated from the experiment can be found online [4].

In order to evaluate the role of each element in the query approach, the errors for the set of unanswered queries were classified into 5 categories:

(1) *PODS Error*: Queries where the final PODS query form did not match the dataset structure.
(2) *Literal Pivot Error*: Queries in which the main entity was a literal instead of an object resource.
(3) *Overloaded Pivot Error*: Queries in which the main entity is a class with more than 3 terms e.g. yago:HostCitiesOfTheSummerOlympicGames.
(4) *Relatedness Error*: Queries where the relatedness measure leads to a wrong answer.
(5) *Combined Pre/Post-Processing Error*: Queries which demanded more sophisticated query interpretation and post-processing.

Table 2 contains the distribution of error types. The complementary error analysis for each query can be found online [4].

The error analysis shows that the distributional approach was able to cope with the semantic variation of the dataset (low level of *Relatedness Error*). The low level of *PODS Error* also shows that PODSs provide a primary query compositional representation suitable for the proposed query approach and for the dataset representation. Queries referring to literal objects as key query entities are currently not addressed by the approach (*Literal Pivot Error*) since only URI resources are mapped into pivot entities in the entity space. This limitation can be addressed by

Table 1. Quality of results for the semantic space measured using 50 natural language queries over DBPedia. The first row represents the results for the full QALD query set while the second row contains a reduced query set where some classes of queries were removed.

Query Set Type	MRR	Avg. Precision	Avg. Recall
Full DBPedia Query Set	0.516	0.482	0.491
Partial DBPedia Query Set	0.680	0.634	0.645

Table 2. Error types and distribution.

Error Type	% of Queries
PODS Error	8%
Literal Pivot Error	4%
Overloaded Pivot Error	8%
Relatedness Error	2%
Combined Pre/Post-Processing Error	20%

mapping literals to the entity space. Most of the errors in the evaluation are in the *Combined Pre/Post-Processing Error* category, which concentrates errors relative to the lack of a pre/post-processing analysis necessary to cope with a natural language query scenario, such as answer type detection, more comprehensive linguistic analysis of the query, post-processing filters, etc. Despite the relevance of evaluating the suitability of the proposed semantic space as a natural language query scenario, this error category does not reflect directly the effectiveness of the semantic representation and query approach as a supporting structure for the natural language query process.

The second line in Table 1 provides a comparative basis of quality measures removing the category containing errors which are considered addressable in the short term (*Literal Pivot Error*) and the category which does not reflect the core of the evaluation for this work (*Combined Pre/Post-Processing Error*). Compared to the results using the approach described in [8] but using the full QALD DBPedia training dataset, there is an improvement of 5.2% over mrr, 18% over avg. precision, and 8.2% over avg. recall. The individual analysis of the entity and spreading activation queries shows that the introduction of the proposed refinements for the semantic space construction led to a quantitative improvement which might be overshadowed by errors present in the *Combined Pre/Post-Processing Error* category.

6.2. *Discussion*

The evaluation focused on the determination of the quality of the approach. No rigorous index construction performance evaluation was considered since, to be comparatively meaningful with existing approaches, a minimum level of optimization in the index construction process was necessary. One clear strength of the approach from the index construction perspective is the fact that the intrinsic nature of the distributional semantic space makes the index construction process straightforward to parallelize, where different regions of the graph can be indexed independently and distributed across different machines. From the query/search perspective, the index structures corresponding to the individuals, classes and entity relations can be distributed across different machines. Query results can be merged once the ranking behavior of the distributional relatedness measure is well defined [22].

7. Related Work

The related work section concentrates on the analysis of works proposing new vector space based models with emphasis on distributional semantics [13, 24–26] and search/indexing strategies for structured graph data (structure indexes) [14, 15]. The motivation for this section is to provide to the reader a perspective over existing trends in the space of distributional semantics, new vector space models and structured data search, also providing a comparative basis with existing work.

7.1. *Distributional compositional semantics*

Clark and Pulman [13] provide a formal description of a compositional model of meaning, where distributional models are unified with a compositional theory of grammatical types (using Labek's pregroup semantics [12]). The approach focuses on the unification of the quantitative strength of distributional approaches with the compositionality provided by symbolic approaches. The final mathematical structure uses vectors to represent word meanings, grammatical roles represent types in a pregroup, and the tensor product to allow the composition of meaning and types. Coecke *et al.* [26] addresses some of the shortcommings present in the model of Clark and Pulman [13] proposing a generalized mathematical framework for a compositional distributional model of meaning. Grefenstette [24] proposes a concrete method for implementing the approach described in [26]. The proposed structured vector space is the tensor product of two noun spaces, in which the basis vectors are pairs of words each augmented with a grammatical role. Meaning of sentences are compared by computing the inner product of their vectors.

Erk and Pado [25] introduce a structured vector space model which integrates syntax into the computation of word meaning in its syntactic context. The model is intended to address the limitations of vector composition models, which reduce the structure of complex sentences to single vectors. The model proposed by Erk and Pado [25] takes into account syntax, by introducing, in addition to the word's lexical meaning, vectors representing the semantic expectations/selectional preferences for relations that the word supports.

One common aspect between the T-Space and [13, 24–26] is the use of distributional semantics in conjunction with the compositional element provided by syntax. The T-Space approach, however, focuses on a Generalized Vector Space Model (GVSM) formalization, defining the process of semantic matching as search operations over the distributional structured space. Additionally, this work also concentrates on an experimental verification of the suitability of the proposed model as a semantic information retrieval model.

7.2. *Query and search mechanisms for RDF*

Different works have focused on searching and querying RDF data. This section concentrates on index structures which preserve the graph structure (structure

indexes). For a more comprehensive discussion on existing query/search approaches for RDF data, the reader is directed to [28]. Semplore [14] is a search engine for Linked Data which uses a hybrid query formalism, combining keyword search with structured queries. The Semplore approach consists in indexing entities of the Linked Data Web (instances classes and properties) using the associated tokens and sub/superclasses as indexing terms. In addition to entity indexing, Semplore uses a position-based index approach to index relations and join triples. In the approach, relation names are indexed as terms, subjects are stored as documents and the objects of a relation are stored in the position lists. Based on the proposed index, Semplore reuses the IR engine's merge-sort based Boolean query evaluation method and extends it to answer unary tree-shaped queries. Dong and Halevy [15] propose an approach for indexing triples allowing queries that combine keywords and structure. The index structure is designed to cope with two query types: predicate queries and neighborhood keyword queries. The first type of queries covers conjunctions of predicates and associated keywords. Dong and Halevy propose four structured index types which are based on the introduction of additional structure information as concatenated terms in the inverted lists. Taxonomy terms are introduced in the index using the same strategy. Schema-level synonyms are handled using synonyms tables. Both approaches [14, 15] provide limited semantic matching strategies and are built upon minor variations over existing inverted index structures. By avoiding major changes over existing search paradigms, these approaches can inherit the implementation of optimized structures used in the construction of traditional indexes.

This work generalizes the basic elements present in the query approach introduced in [8], building a distributional structured vector space model. This model is a fundamental step towards bringing scalability to the basic elements of the query approach described in [8]. The generalization also includes a change from the previous semantic relatedness approach, which was based on a link-based relatedness measure (Wikipedia Link Measure [16]), to a distributional approach based on Explicit Semantic Analysis (ESA). An additional refinement includes the entity indexing strategy which moved from a uniform entity indexing to an entity index which differentiates instances (TF/IDF) and classes (ESA). Differently from the previous approach [8], which performed a query execution time navigation over the graph and a pairwise computation of the semantic relatedness measure between query terms and each relation, this work proposes the introduction of a relation index associated with each entity, bringing a principled solution to reduce the original query execution time.

8. Conclusion and Future Work

This work proposes a distributional structured semantic space (T-Space) focused on addressing a fundamental challenge for RDF data queries, where the data model heterogeneity of the Web demands a query approach focused on abstracting users

from an *a priori* understanding of the data model behind datasets. Key elements in the construction of the approach are: (i) the application of *distributional semantics*, which in this work is defined by Explicit Semantic Analysis (ESA); (ii) a *compositional semantic model* based on the structure of RDF and on the partial ordered dependency structures for natural language queries and (iii) a *hybrid search model* where *entity-centric search* is complemented by *spreading activation search*. The final distributional structured semantic space allows data model independent natural language queries over the RDF data, achieving *mean reciprocal rank* = 0.516, *avg. precision* = 0.482 and *avg. recall* = 0.491, evaluated using 50 natural language queries over DBPedia. The elements of the proposed approach are formalized in a distributional vector space model based on the Generalized Vector Space Model (GVSM). The construction of the T-Space preserves the semantic information present in the graph structure, while keeping the scalability of the vector space model and the flexibility of the semantic matching provided by the distributional semantic model.

Future work will include the implementation of optimizations in the index construction process and evaluation of the index construction/query execution time, the elimination of limitations which are considered addressable in the short term and the implementation of a question answering (QA) system over RDF data using the proposed index. The implementation of QA features (e.g. answer tye detection) will allow the comparative evaluation against existing QA systems [3].

Acknowledgment

This work has been funded by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

References

[1] T. Berners-Lee, Linked Data Design Issues, http://www.w3.org/DesignIssues/LinkedData.html, 2006.

[2] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak and S. Hellmann, DBpedia — A crystallization point for the Web of Data, in *Web Semantics*: *Science, Services and Agents on the World Wide Web* 7, 2009.

[3] 1st Workshop on Question Answering over Linked Data (QALD-1), http://www.sc.cit-ec.uni-bielefeld.de/qald-1, 2011.

[4] Evaluation Dataset, http://treo.deri.ie/semanticspace/icsc2011.htm, 2011.

[5] E. Gabrilovich and S. Markovitch, Computing semantic relatedness using Wikipedia-based explicit semantic analysis, in *Proc. International Joint Conference on Artificial Intelligence*, 2007.

[6] G. Furnas, T. Landauer, L. Gomez and S. Dumais, The vocabulary problem in human-system communication, *Communications of the ACM* **30**(11) (1987) 964–971.

[7] M. Sahlgren, The distributional hypothesis: From context to meaning, Distributional models of the lexicon in linguistics and cognitive science (Special issue of the *Italian Journal of Linguistics*), *Rivista di Linguistica* **20**(1) (2008).

[8] A. Freitas, J. G. Oliveira, S. O'Riain, E. Curry and J. C. Pereira da Silva, Querying linked data using semantic relatedness: A vocabulary independent approach, in *Proc. of the 16th International Conference on Applications of Natural Language to Information Systems*, NLDB 2011, 2011.

[9] P. D. Turney and P. Pantel, From frequency to meaning: Vector space models of semantics, *Journal of Artificial Intelligence Research* **37** (2010) 141−188.

[10] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in *Proc. International Joint Conference on Artificial Intelligence*, 1995.

[11] S. Pado and M. Lapata, Dependency-based construction of semantic space models, *Computational Linguistics* **33**(2) (2007) 161−199.

[12] J. Lambek, The mathematics of sentence structure, *The American Mathematical Monthly* **65**(3) (1958) 154−170.

[13] S. Clark and S. Pulman, Combining symbolic and distributional models of meaning, in *Proc. of the AAAI Spring Symposium on Quantum Interaction*, 2007, pp. 52−55.

[14] H. Wang, Q. Liu, T. Penin, L. Fu, L. Zhang, T. Tran, Y. Yu and Y. Pan, Semplore: A scalable IR approach to search the Web of Data, *Web Semantics*: *Science, Services and Agents on the World Wide Web* **7**(3) (2009) 177−188.

[15] X. Dong and A. Halevy, Indexing dataspaces, in *Proc. of the 2007 ACM SIGMOD International Conference on Management of Data*, 2007.

[16] D. Milne and I. H. Witten, An effective, low-cost measure of semantic relatedness obtained from Wikipedia links, in *Proc. of the First AAAI Workshop on Wikipedia and Artificial Intelligence* (*WIKIAI'08*), Chicago, IL, 2008.

[17] A.-M. Popescu, O. Etzioni and H. A. Kautz, Towards a theory of natural language interfaces to databases, in *Proc. of the 8th International Conference on Intelligent User Interfaces*, 2003, pp. 149−157.

[18] S. K. M. Wong, W. Zarko and P. C. N. Wong, Generalized vector space model in information retrieval, in *Proc. of the 8th ACM SIGIR Conference on Research and Development in Information Retrieval*, 1985, pp. 18−25.

[19] T. Gottron, M. Anderka and B. Stein, Insights into explicit semantic analysis, in *Proc. of the 20th ACM International Conference on Information and Knowledge Management*, 2011.

[20] M. Anderka and B. Stein, The ESA retrieval model revisited, in *Proc. of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009, pp. 670−671.

[21] T. Eiter, G. Ianni, T. Krennwallner and A. Polleres, Rules and ontologies for the semantic web, *Reasoning Web*, 2008.

[22] A. Freitas, E. Curry and S. O'Riain, A distributional approach for terminological semantic search on the linked data web, in *Proc. of the 27th ACM Symposium on Applied Computing, Semantic Web and Applications*, 2012.

[23] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval* (Addison-Wesley, New York, 1999).

[24] E. Grefenstette, M. Sadrzadeh, S. Clark, B. Coecke and S. Pulman, Concrete sentence spaces for compositional distributional models of meaning, in *Proc. of the 9th International Conference on Computational Semantics*, 2011.

[25] K. Erk and S. Pado, A structured vector space model for word meaning in context, in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2008.

[26] B. Coecke, M. Sadrzadeh and S. Clark, Mathematical foundations for a compositional distributional model of meaning, Vol. 36, *Linguistic Analysis* (Lambek Festschrift), 2010.

[27] A. Freitas, J. G. Oliveira, E. Curry and S. O'Riain, A multidimensional semantic space for data model independent queries over RDF data, in *Proc. of the 5th International Conference on Semantic Computing*, 2011.

[28] A. Freitas, E. Curry, J. G. Oliveira and S. O'Riain, Querying heterogeneous datasets on the linked data web: Challenges, approaches and trends, *IEEE Internet Computing, Special Issue on Internet-Scale Data*, 2012.