A Distributional Approach for Terminological Semantic Search on the Linked Data Web

André Freitas
Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway
andre.freitas@deri.org

Edward Curry
Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway
ed.curry@deri.org

Seán O'Riain
Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway
sean.oriain@deri.org

ABSTRACT

The process of searching and understanding existing vocabularies (terminological artifacts) on the Linked Data Web is an intrinsic activity to the consumption and production of Linked Data. Data consumers trying to find and understand the vocabularies behind datasets in order to query them, or data producers searching for existing resources to describe their data, face the challenge of semantically searching existing concepts in vocabularies. Traditional search mechanisms do not address the level of semantic matching necessary to match users' information needs to vocabulary elements, bringing an additional barrier to the consumption and production of Linked Data on the Web. This work describes a terminological search mechanism which uses a distributional semantic model to provide a best-effort semantic matching solution. The distributional semantic model leverages the semantic information present in large volumes of unstructured text to improve the semantic matching capabilities of the search process. A quantitative evaluation of the quality of the search results shows that the approach provides an effective semantic matching mechanism for terminological search.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Indexing methods; H.3.3 [**Information Search and Retrieval**]: Retrieval models.

General Terms

Semantic Search.

Keywords

Terminological Search, Vocabulary Search, Distributional Semantics, Explicit Semantic Analysis, Linked Data.

SAC'12 March 25-29, 2012, Riva del Garda, Italy.

1. INTRODUCTION

The last few years have witnessed Linked Data [1] emerge as a de-facto standard for publishing data on the Web, bringing the potential of a paradigmatic change in the scale which users and applications reuse, consume and repurpose data. However, together with its opportunities, Linked Data brings inherent challenges in the way users and applications consume and generate Linked Data. Linked datasets are not based on a rigid schema. Instead, datasets on the Linked Data Web are dependent on the definition of data models based on *vocabularies* (a.k.a. *terminological artifacts or lightweight ontologies*) which define a lightweight data model for Linked datasets.

Vocabularies are in the center of the semantic model of the Linked Data Web and the process of understanding and reusing vocabulary concepts is an activity intrinsic to the consumption or production of Linked Data. From the Linked Data consumption perspective, users building structured SPARQL queries over existing datasets frequently need to engage in the time-consuming process of understanding the vocabularies behind a dataset in order to query existing data. From the perspective of Linked Data production, users trying to maximize the reuse of existing vocabularies to create new datasets, or to semantically enrich existing data, also need to go through the process of search and analysis of existing vocabularies.

Existing approaches to support users in the search process do not address the level of semantic matching necessary for searching concepts on the Linked Data Web. Most of the existing terminological search engines for the Linked Data Web (section 6) use variations of traditional vector space model approaches (TF/IDF extended with PageRank) over labels, descriptions and relations associated with both terminology-level and instance-level entities to index existing concepts on the Linked Data Web. These approaches lack the level of semantic approximation that is necessary to match the user information needs expressed in a keyword query to the entities present on the Linked Data Web. While the process of searching instance-level entities is less subject to ambiguity and polisemy (e.g. dbpedia:Abraham_Lincoln, dbpedia:Berlin), the lexical and semantic variability intrinsic to the terminological-level entities (classes and properties) (e.g. dbpedia-owl:weapon) demands more sophisticated semantic search approaches.

This paper describes a *terminological search approach* focusing on the provision of an effective semantic matching for searching vocabulary elements on the Linked Data Web.

The proposed approach uses a *distributional semantic model* based on the semantic information present on the Wikipedia corpus to define a meaning interpretation approach to be used in the search mechanism. The quality of the search results is evaluated and a model based on the concept of *semantic differential* is introduced. This work extends the discussion of a distributional query model introduced in [5], focusing on the construction and evaluation of a terminological search approach.

The paper is organized as follows: section 2 introduces motivation and requirements; section 3 describes the concept of distributional semantics, semantic relatedness and Explicit Semantic Analysis (ESA), which are in the center of the proposed approach; section 4 covers the proposed approach for indexing and searching terminological artifacts; section 5 provides an evaluation of the approach focused on the quality of results, which is followed by section 6, covering related works in the area; section 7 provides conclusions and future work.

2. SEMANTIC TERMINOLOGICAL SEARCH

2.1 Motivation & Applications

This section introduces the motivations and potential applications which can be enabled by the deployment of terminological semantic search mechanisms. All the motivational scenarios discussed below focus on problems highly dependent on effective semantic matching mechanisms. This discussion serves to build a set of requirements (section 2.2) from a user-centered perspective.

1. *Dataset discovery and understanding*: In order to formulate a SPARQL query over Linked Datasets users should be able to discover which Linked Datasets potentially contain data of interest and understand the vocabulary behind these datasets. A terminology-level search mechanism can support data consumers in the process of dataset discovery by allowing keyword queries over terminology-level data in distributed Linked Datasets. The returned terminological resources can be used to determine datasets of interest or can be used to formulate SPARQL queries.

2. *Reuse of existing vocabularies:* The reuse of existing vocabularies is a fundamental element in the process of creating new Linked Data, in order to maximize the shared semantics across datasets. Users trying to maximize the reuse of concepts in existing vocabularies need to engage in the time consuming process of searching for vocabularies of potential interest and getting familiarized with their concepts, in order to use them.

2.2 Requirements

This section enumerates the core requirements for a terminological search mechanism based on the motivations and applications introduced in the previous section.

1. *Semantic matching:* In order to cope with the lexical variability and semantic differences in vocabularies, a terminological semantic search mechanism should be able to semantically match the user information needs expressed as a keyword query to the closest terminological concepts present in the Linked Data Web.

2. *Semantic conjunction for multiple keywords:* Users should be able to express the intended semantics using multiple keywords in the terminological search process. The meaning of each term present in the keyword query should be semantically composed, where each term defines a semantic refinement operation (in contrast with each keyword working as a semantically disjoint query element).

3. *List of semantically related terms as a result set:* The motivations and uses of terminological search include an exploratory search process where, instead of searching for a specific concept, users are focused in understanding the resources available on the Linked Data Web. In this scenario, a *best-effort* approach returning a ranked list of semantically related terminological resources allows users to explore a semantic neighborhood that best matches the query, instead of returning the top-most result.

4. *Ability to filter unrelated results/concise result set:* Differently from the filtering criteria of traditional search engines, which return every resource containing the keywords in the query, a semantic search engine ranking resources by semantic relatedness can include resources which are very indirectly related to the search query. With usability in mind, a terminological search engine should be able to return to users a concise result set.

5. *Capacity to handle vocabularies with minimum description:* The degree of description associated with terminological resources varies from vocabulary concepts annotated with rich natural language labels and descriptions, passing through rich taxonomic or vocabulary structures, to concepts which just rely on the information present on the associated URIs. While a terminological search engine can use the additional information to improve its results, it should be able to operate with a minimum description level.

6. *Common requirements for search mechanisms:* The list above emphasizes the set of requirements specific to a terminological search mechanism. In addition, terminological search mechanisms should attend the set of common requirements for search engines which include high precision and recall, low query execution time, low index construction/update time and scalability.

3. DISTRIBUTIONAL SEMANTIC MODEL

The problem of providing the level of semantic interpretation needed to enable effective semantic search capabilities has been associated with challenges in Artificial Intelligence such as commonsense knowledge representation and reasoning. The rationale behind this perception is consistent: an ideal semantic search mechanism, where users could be completely abstracted from the actual representation of the information, needs to be supported by a semantic model which is able to provide a rich semantic interpretation of both the query and the information collection.

More recently the availability of large quantities of unstructured text on the Web motivated the creation of semantic models that are leveraged using the semantic infor-

mation embedded in these corpora [4]. Instead of approaching the construction of a semantic model from a knowledge representation perspective, these approaches have defined a simplified semantic model which is based on the statistical distribution of words in corpora. These *distributional semantic models* rely on the assumption that words that co-occur in similar contexts tend to have similar meanings (distributional hypothesis) [4].

Distributional semantic models are naturally represented using vector space models, where the meaning of a word is usually defined by a weighted vector of co-occurring words (e.g. gun, weapon, pistol). The definition of meaning provided by distributional semantics defines a semantic model with an intrinsic differential nature, suitable for computing differences in meaning between words [4]. More explicitly, it is possible to rephrase the distributional hypothesis to differences in meaning are mediated by differences in distribution [4]. The concept of *semantic relatedness* follows as the dual counterpart of the computation of semantic differences. The differential nature of the meaning implied by distributional semantics defines the scope of its applicability, making it suitable for the computation of *semantic relatedness measures* between text elements.

This work uses Explicit Semantic Analysis (ESA) [3], a distributional semantic model which represents the meaning of a text in a vector space of concepts derived from the Wikipedia corpus. Traditional distributional approaches build the meaning vector space using text windows through the corpus to build the co-occurrence word vector. The ESA approach differs as it represents the interpretation of the meaning of a word using references to Wikipedia article titles, where the occurrence of the word is semantically significant. The ESA space is built by indexing Wikipedia articles as documents in an inverted list index. ESA uses the TF/IDF ranking scheme to determine the semantic significance of a term in the text collection and the article-based structure of Wikipedia as its semantic context. The ESA semantic interpreter uses the index to build concept vectors associated with keyword terms, returning a weighted vector of Wikipedia article titles associated with a term. Multiple keyword terms are handled by calculating the centroid of the multiple vectors generated from the ESA concept space. The experimental evaluation of ESA shows a high correlation with human assessments in the computation of semantic relatedness [3].

4. INDEXING AND SEARCHING TERMINOLOGICAL DATA

The core idea behind the construction of the proposed terminological search approach is to use the interpretation vectors provided by ESA to build a semantic vector space which can support the semantic matching between user information needs expressed as keyword queries and vocabulary concepts. The knowledge embedded in a third-party corpus (in this case, Wikipedia), defines a comprehensive semantic model which is used as the base for the semantic interpretation of the query and vocabulary elements.

The procedure for the terminological space construction starts by building the ESA interpreter (*ESA concept space*) (Figure 1)(1). The construction of the ESA interpreter was described in the previous section. Keyword queries sent to the interpreter return a weighted vector of article titles, which defines a concept vector associated with the set of keywords. The weighted concept vector encodes a distributional representation of the word semantics. Terms associated with vocabularies' URIs (e.g. labels or parsed URIs) are extracted from the vocabularies (2) and are sent as queries to the *ESA concept space* (3), which returns the associated concept vector. The concept vectors for the vocabulary terms are used to build the final terminological semantic space (4). The concepts associated with each vector component generate new dimensions in the terminological semantic space. The final space contains a set of weighted vectors representing the vocabulary concepts, where the dimensions are defined by the ESA concepts, and the weight of each vector component is given by the TF/IDF score associated with the incidence of the vocabulary term in relation to each article.

The terminological semantic space forms a vector space which has its dimensionality dependent on the number of indexed concepts and on the arbitrary decision on the dimensionality of the ESA concept vectors. In the worst-case scenario the dimensionality of the terminological space equals the number of Wikipedia articles which are indexed in the ESA concept space. In this work the dimensionality of the ESA concept vector is defined as d=50, a number which was determined empirically by observing the value of the ESA weight decay on the ordered list of concept vectors.

The vector space dimensionality impacts the search performance and the scalability of the proposed approach, depending on the application of scalability strategies such as index distribution for handling large volumes of data. The proposed index can be distributed by applying a vocabulary distribution criteria (e.g. alphabetical order of vocabulary concepts, sets of vocabularies, etc), indexing the concepts in independent indexes. The search process can be distributed across the indexes and the results can be merged afterwards, since the calculation of the semantic relatedness score only depends on the common ESA concept space.

A query over the terminological semantic space is functionally equivalent to the computation of the semantic relatedness between the query term and all the vocabulary concepts which are represented in the semantic space, returning a ranked list of semantically related vocabulary elements. The search process starts with the computation of the ESA concept vector (Figure 2) (2) for the keyword query (1). Multiple query terms are handled according to the default ESA procedure. The query vector is then used to compute the cosine similarity between the query and the indexed vocabulary concepts (3). The ranked list of similar vectors is then filtered by using the combination of a discrimination threshold (covered in section 5.3) and a fix top-k cut-off filter (4). Figure 2 depicts the search process, where for an example keyword query *gun*, the approach returns a list of related concepts (5) from DBPedia. In this case, the target vocabulary concept is the top-most result (*Weapon*). The approach also returns additional terms with some degree of semantic relatedness to *gun*.

5. EVALUATION

5.1 Evaluating Terminological Search

The evaluation of a terminological search mechanism should measure the suitability in relation to the set of requirements raised in section 2 and in particular, the quality of the proposed semantic matching. In order to evaluate the approach,

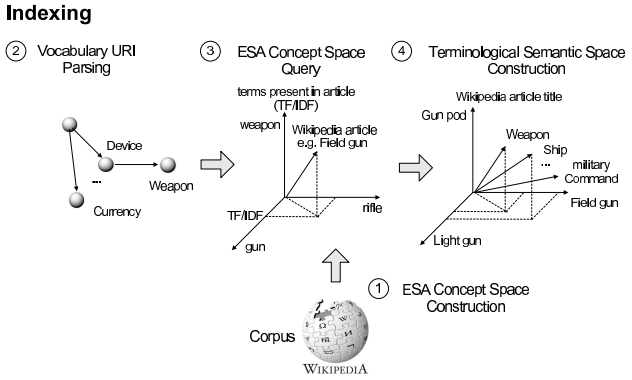

Figure 1: Terminological semantic space construction.

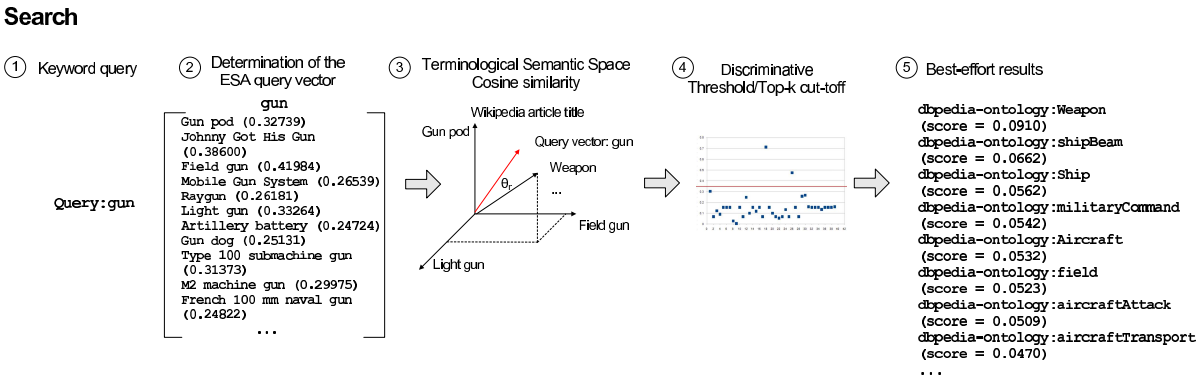

Figure 2: Terminological semantic space search process.

three evaluation dimensions are proposed: *quality of search results*, *semantic differential analysis* and *performance indicators*.

The approach was evaluated indexing 1,610 concepts (275 classes and 1,335 properties) present in the 3.6 version of the DBPedia vocabulary. The DBPedia vocabulary was chosen due to the size and comprehensive nature of the vocabulary. A prototype, named *Brí* (after the Irish word for *meaningful*) was implemented. The prototype was built focusing on measuring the quality of the proposed approach, consisting of an in memory inverted terminological index and an ESA concept space [3]. A 2006 version of Wikipedia (approximately 1.5 million articles) was used in the creation of concept space and a size of 50 concepts was defined for each concept vector. The performance indicators were collected on a Intel Core 2 Duo machine with 1GB of RAM allocated for the prototype. The procedure for generating the set of keyword queries was based on the process of asking two users to tag 60 commonsense images and their constituent elements with keywords. The set of tags which could be mapped to related concepts in the DBPedia ontology were used to define the set of 143 keyword queries (query size of 1-2 terms). This procedure was used to generate the *search for highly related concepts* behavior expected in terminological search. In order to comply with the minimum description assumption (req. 5), the information present in properties' domains and ranges axioms were not used in the indexing process: just the specific vocabulary element name embedded in each URI was used. The data associated with the experiments can be found in [2].

5.2 Quality of Search Results

In order to make the discussion on the semantic matching properties of the terminological space more concrete, examples of keyword queries and best-effort results are listed in Figure 3. The example queries lists the top-8 most semantic related terms to natural language queries over the DBPedia ontology. The example queries illustrate the semantic matching problem for terminological search, where the closest related concept can be expressed by different semantic relationships, varying from string variations (e.g. books - Book), synonyms and taxonomic ancestors to broader classes of semantic relations (e.g. justice - SupremeCourtOfTheUnitedStatesCase, Judge). The fine grained semantic nature of the search approach is exemplified in the queries *bass* and *bassist*, where the closest related concept *Instrument* is highly ranked in the *bass* query. For the query *bassist* the closest related concept *musician* is highly ranked. These examples give a taste of the reasoning-like behavior which is supported by distributional semantics. Some of the queries allow the verification of the semantic conjunction behavior (req. 2) where multiple keywords should match the closest related concept in the conjunction of keyword concepts, instead of returning disjoint matches for each keyword query. Figure 3 exemplifies this behavior using the queries *engine* and *car engine* and the list of associated rankings.

The quantitative part of the evaluation measures the quality of the approach under the scope of the motivations and

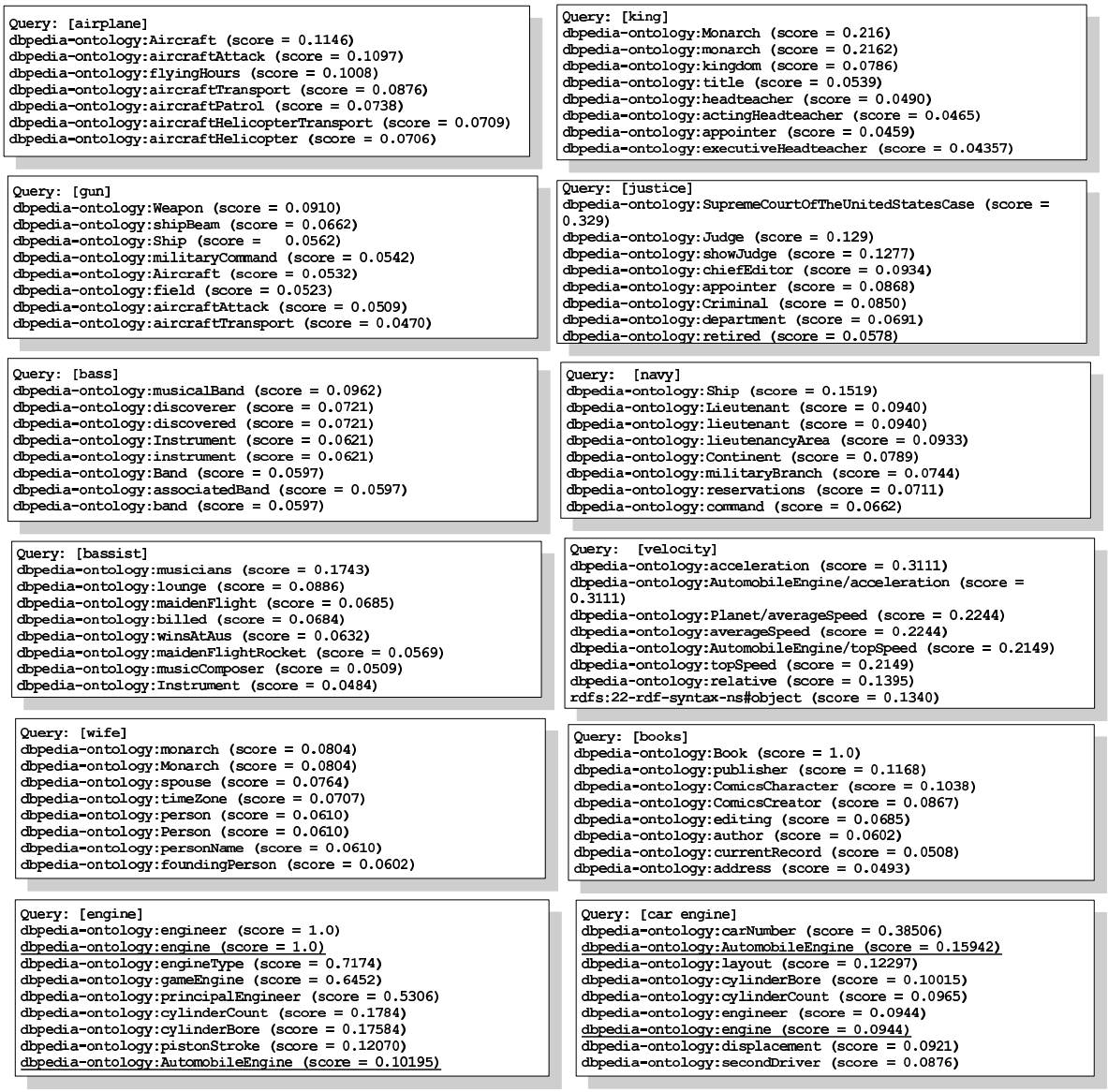

Figure 3: Set of example queries over the DBPedia vocabulary and top-8 results.

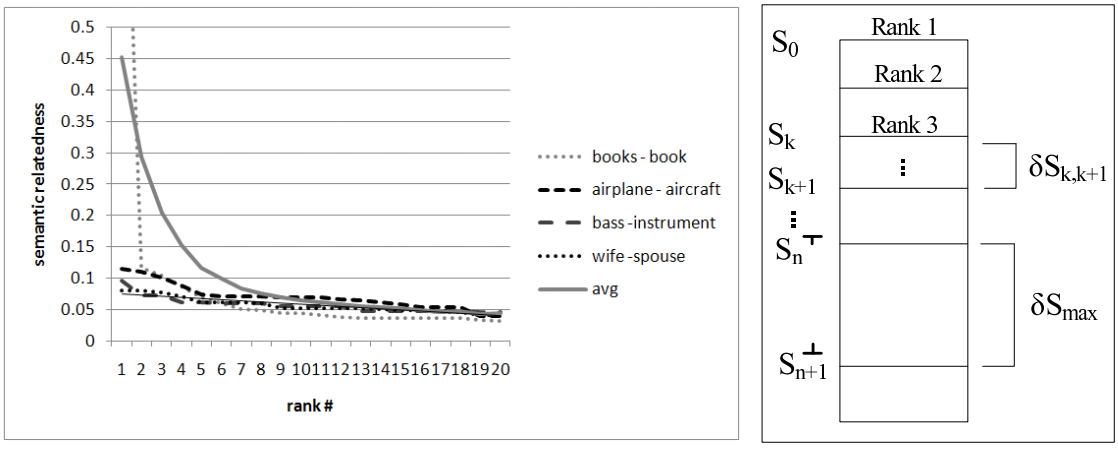

Figure 4: On the left, semantic relatedness scores for sample query-vocabulary matches. On the right a representation of the semantic differential model.

requirements for terminological search. The first measure, *% of queries correctly answered with semantically related terms*, evaluates the percentage of queries which are answered with resources which are closely semantically related. The results show that the **semantic approach answers 92.25%** of the 143 queries with semantically related terms. *Average precision@n* is defined as the number of closely related terms in the top-n semantically related results over the number of returned results. The approach presents high average precision, which is kept along the top-5 and top-10 results (**avg. p@5=0.732, avg. p@10=0.691**). *Mean reciprocal rank* (MRR) measures the ranking quality by calculating the inverse of the rank of the best result (e.g. for the best answer ranked as the second result, the reciprocal rank is 1/2). In the case of the list of semantically related results the best-result is defined as the closest related concept and not as the top related ranked result. The final MRR value shows that the quality of the ranking is high (**MRR=0.646**) where, on the average, most of best results are located on the first or second positions.

In order to provide a comparative baseline for the approach, the same vocabulary was indexed using a TF/IDF index which, under the minimum description assumption of the experiments, worked essentially as a simple string matching approach (with stemming). A second baseline was generated using a WordNet-based query expansion. The results show that the *distributional approach largely outperforms the string matching and simple WordNet-based approaches*, where the **first (string matching) baseline answers 45.77% of the queries** and the **second (string matching + WordNet query expansion) baseline answers 52.48% of the queries**, compared to **92.25% for the distributional ESA approach**. Additionally, one characteristic which is not fully expressed in the comparative evaluation measures is the fact that the proposed approach provides a much more comprehensive exploratory search results, allowing users to have a better understanding of the conceptual coverage of the elements on the vocabularies.

5.3 Semantic Differential Analysis

The purpose of this part of the evaluation is to analyze the distribution and the semantic differential behavior of the ranked list of semantic relatedness scores. This analysis can support the detection of a semantic gap (or semantic discrimination) between highly semantically related resources and the top average non-related terms. The semantic distribution of the top-20 (non-filtered) relatedness scores for 4 queries + the average of the scores for all 143 queries is depicted on the left side of Figure 4. The right side of the figure shows the symbols that are used to describe the main concepts of the *semantic differential model*, depicting a ranked list of results, where S_k represents the relatedness values associated with the k+1 ranked concept, S_0 is the maximum relatedness value, $\delta S_{k,k+1}$ the semantic differential between two adjacent ranked concepts, δS_{max} is the maximum semantic differential in the unfiltered ranked list and $S_n^{\top}$, $S_{n+1}^{\perp}$ are respectively the top and bottom relatedness values of δS_{max}.

Table 1 shows the values and the distribution of the elements of the semantic differential model for the full (unfiltered) query set. Queries with literal string matching approach semantic relatedness scores close to 1 (the maximum value). On the average, high conceptually related matching happens on the range between 0.5 and 0.1. The average size of the maximum semantic differential is significantly larger than the average semantic differential, showing a clear discriminative nature for the relatedness score. Most of δS_{max} values are located above 0.1. This is confirmed by the distribution of $S_n^{\top}$, $S_{n+1}^{\perp}$ which also shows that very few δS_{max} fall below 0.07. The range 0.1 to 0.07 still represents a significant range for semantically related concepts.

The semantic differential analysis defines a threshold criteria for the relatedness scores. The values which define the threshold are specific to ESA and to the corpora used and it is likely that these values will vary for other corpora and distributional models. The main contribution of the differential analysis proposed here is the definition of a principled differential semantic model and threshold determination methodology which can be reused in different distributional models. The threshold t(S) is defined as:

$$t(S) = \begin{cases} \mathrm{S}_{n+1}^{\perp} & \text{if } S_n^{\top} > 0.1 \text{ and } S_{n+1}^{\perp} > 0.07 \\ 0.07 & \text{if } S_{n+1}^{\perp} < 0.07 \end{cases}$$

5.4 Additional Analysis

The last evaluation dimension consists in the measurement of the execution time performance and scalability of the approach under the current implementation setting. These indicators however, should be taken in the context of the fact that no optimization mechanisms were considered in the reference implementation. The final dimensionality of the terminological space is approximately 30,000 for indexing 1,610 concepts (for the dimension of each ESA vector d=50). A constant (in relation to the dimensionality) *average index update time* was measured as 11,184.71 ms and an *average query execution time* for the terminological space was measured as 5,232 ms. In relation to the listed requirements (section 2.2), the experimental approach was able to provide a best-effort semantic matching approach (req.1) with a *semantic relatedness behavior* (req.3), working in a *minimum description scenario* (req.5) The approach also addresses the *semantic conjunction requirement* (req.2) and provides a differential semantic model which allows the *quantification of a discriminative threshold* (req.4). The approach provided high-quality results, however, further investigation is needed in relation to *execution performance and scalability mechanisms* (req.6).

6. RELATED WORK

The related work section concentrates on the analysis of search mechanisms which had focused on providing a terminology level search functionality.

Swoogle [6] is a document-centric search engine for indexing Semantic Web vocabularies and documents (documents in supported Semantic Web standards which contain triples). The ranking approach used in Swoogle focuses on a rational random surfing model which applies a variation of the PageRank algorithm, weighting differently links with different semantics across different vocabularies. Swoogle classifies the links between documents into four categories taking into account their document-level relationship: *imports*, *uses-term*, *extends*, and *asserts*. The Swoogle index is built by extracting keywords from URIs present in the documents, building a keyword-based vector space where documents are represented. Swoogle allows users to search

Measure	Value
Avg. Semdiff (δS)	0.006523
Avg. Maximum Semdiff (δS_{max})	0.281752
Avg. Maximum Relatedness Value (S_{max})	0.452145
Avg. Relatedness Value: Top Semdiff Extreme ($S_n^{\top}$)	0.417370
Avg. Relatedness Value: Bottom Semdiff Extreme ($S_{n+1}^{\perp}$)	0.135618
% of Top Semdiff Extreme ($S_n^{\top}$) ≥ 0.1	81%
$0.09 \leq$ % of Top Semdiff Extreme ($S_n^{\top}$) < 0.1	4%
$0.07 \leq$ % of Top Semdiff Extreme ($S_n^{\top}$) < 0.09	8%
% of Top Semdiff Extreme ($S_n^{\top}$) < 0.07	7%
% of Bottom Semdiff Extreme ($S_{n+1}^{\perp}$) ≥ 0.1	44%
$0.09 \leq$ % of Bottom Semdiff Extreme ($S_{n+1}^{\perp}$) < 0.1	9%
$0.07 \leq$ % of Bottom Semdiff Extreme ($S_{n+1}^{\perp}$) < 0.09	18%
% of Bottom Semdiff Extreme ($S_{n+1}^{\perp}$) < 0.07	29%

Table 1: Measures and distribution for the semantic differential analysis.

vocabularies, documents, and URIs present in vocabularies and documents.

Falcons Concept Search [7] is a keyword-based ontology search engine, which retrieves ontology concepts based on a combination of term-based relevance and popularity scores. The term-based similarity defines a ranking score between virtual documents associated with ontology concepts (a virtual document is defined by a subgraph containing neighboring ontology elements) and the user keyword query.

Sindice [10] is an entity-centric search and query service for the Linked Data Web which ranks entities according to the incidence of keywords associated with the entities present in the dataset, using a node-labeled tree model which represents the relationship between datasets, entities, attributes and values.

Compared to the approach proposed in this work, all the previous search mechanisms [6][7][10] lack evaluation of the quality of the search results. In addition, these approaches try to address the semantic matching problem by leveraging on the information associated with entities on the ontology (such as descriptions or neighboring nodes), avoiding a more principled semantic matching approach for terminology-level elements.

SQORE [9] is an ontology retrieval system which targets a query interface that allows users to formulate queries containing structural ontology constraints. SQORE uses XML Declarative Description (XDD) to facilitate ontology matching and reasoning [9]. In addition, it uses lexical databases such as WordNet to support semantic matching. The ontology ranking is done by combining semantic and structural similarity scores and the approach is evaluated in terms of the quality of the search results. Compared to SQORE, the proposed approach introduced in this work targets the use of distributional semantics based on Web corpus (Wikipedia) to address the semantic matching problem, providing a more comprehensive semantic matching solution.

Alani & Brewster proposes an ontology ranking approach [8] for a keyword-based search mechanism based on the composition of graph-analysis measures. The approach combines basic string matching, structural features (e.g. taxonomic centrality), structural density (connectivity of the ontology elements) and semantic cohesiveness of the ontology (using a semantic similarity measure). The ranking approach [8] was evaluated by verifying the correlation with a human-based gold-standard rank. The reported results were inconclusive in relation to the quality of the proposed composite measure as a ranking function. The approach proposed by Alani & Brewster [8] focuses on the quantification of the structural quality and completeness of the ontologies, not focusing on the semantic matching problem.

Existing works on ontology search had not focused on the semantic matching problem for terminology-level search under the minimum description requirement scenario. An additional limitation in many of the existing approaches [6][7][10] is the lack of an evaluation for the quality of results. The approaches described in [6] [10] [8], which rely on authoritative ranking and structural quality ranking respectively are highly complementary to the approach proposed in this work.

7. CONCLUSION & FUTURE WORK

This paper introduces a terminological search mechanism having as a motivation the search for vocabularies on the Linked Data Web. The proposed approach uses a distributional semantic model instantiated by an Explicit Semantic Analysis (ESA) space to provide an efficient semantic matching approach. The suitability of the approach is confirmed by an experimental evaluation using 143 keyword queries over DBPedia. The final approach, using a minimum description assumption scenario outperformed the baseline approaches, answering 92.25% of the queries, achieving average precision@10=0.691 and MRR=0.646. In addition, the semantic relatedness behavior of the approach was modeled with a differential semantic model, showing a discriminative behavior between related and non-related concepts. The results show that the use of distributional semantics represents an effective semantic matching approach for terminological search. Future work includes the investigation of the quality of the search results using a larger set of vocabularies, the analysis of the proposed approach under a domain specific scenario and the investigation of performance optimization mechanisms.

Acknowledgments

The work presented in this paper has been funded by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

8. REFERENCES

[1] T. Berners-Lee. Linked Data Design Issues, http://www.w3.org/DesignIssues/LinkedData.html, 2009.

[2] Evaluation Dataset, http://treo.deri.ie/bri/swa2012.htm, 2011.

[3] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *In Proc. of the Inl. Joint Conference On Artificial Intelligence (IJCAI)*, 2007.

[4] 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, vol. 20, no. 1, 2008.

[5] 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 Intl. Conference on Semantic Computing (ICSC)*, 2011.

[6] L. Ding, T. Finin, A. Joshi, R. Pan, R. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. *In Proc. of the 13th ACM International Conf. on information and knowledge management(CIKM)*, 2004.

[7] Y. Qu and G. Cheng. Falcons Concept Search: A Practical Search Engine for Web Ontologies. *IEEE Transactions on Systems, Man and Cybernetics*, v.41, 810-816, 2011

[8] H. Alani and C. Brewster. Ontology ranking based on the analysis of concept structures. *In Proc. of the 3rd Intl. Conference on Knowledge Capture*, 2005.

[9] R. Ungrangsi, C. Anutariya and V. Wuwongse. SQORE-Based Ontology Retrieval System. *In Database and Expert Systems Applications*, Springer Berlin-Heidelberg, 4653, 720-729, 2007.

[10] R. Delbru, S. Campinas, G. Tummarello. Searching Web Data: an Entity Retrieval and High-Performance Indexing Model. *In Journal of Web Semantics*, 2011.