Linked Data-Based Concept Recommendation: Comparison of Different Methods in Open Innovation Scenario

Danica Damljanovic[1], Milan Stankovic[2,3], and Philippe Laublet[3]

[1] University of Sheffield,
Department of Computer Science, United Kingdom
d.damljanovic@dcs.shef.ac.uk
[2] Hypios, 187 rue du Temple, 75003 Paris, France
milan.stankovic@hypios.com
[3] STIH, Université Paris-Sorbonne, 28 rue Serpente, 75006 Paris, France
philippe.laublet@paris-sorbonne.fr

Abstract. Concept recommendation is a widely used technique aimed to assist users to chose the right tags, improve their Web search experience and a multitude of other tasks. In finding potential problem solvers in Open Innovation (OI) scenarios, the concept recommendation is of a crucial importance as it can help to discover the right topics, directly or laterally related to an innovation problem. Such topics then could be used to identify relevant experts. We propose two Linked Data-based concept recommendation methods for topic discovery. The first one, hyProximity, exploits only the particularities of Linked Data structures, while the other one applies a well-known Information Retrieval method, Random Indexing, to the linked data. We compare the two methods against the baseline in the gold standard-based and user study-based evaluations, using the real problems and solutions from an OI company.

Keywords: concept recommendation, structure-based similarity, semantic similarity, information retrieval, statistical semantics, linked data, ontologies, recommender systems, concept discovery, open innovation.

1 Introduction

The ability to innovate is essential to the economic wellbeing, growth and survival of most companies, especially when the market competition becomes strong. With the global economic uncertainties in recent years, companies and innovation experts started to question the old innovation models and seek new, more efficient ones. The paradigm of Open Innovation (OI) [1] is proposed as a way to outsource the innovation and seek solutions of R&D problems outside the company and its usual network of collaborators. OI is intended to leverage the existing knowledge and ideas, that the company is unaware of, and somehow democratize the process of innovation. In recent years, one interesting realisation of OI is the one that encourages the innovation

E. Simperl et al. (Eds.): ESWC 2012, LNCS 7295, pp. 24–38, 2012.

to emerge over the Web. This realization is the core business of companies such as Hypios.com, Innocentive.com and NineSigma.com, which provide Web innovation platforms where companies with R&D needs can post problems and find innovative solutions. The companies looking to innovate, called seekers, would represent their R&D needs through an innovation problem statement describing the context of the problem to be solved. Such a statement is then published on a problem-solving platform. Experts, called solvers then submit their solutions. The seeker then selects the best contribution and acquires the rights to use it, often in exchange for a prize to the solver and any other due fees.

Identification of the potential solvers and broadcasting problems to their attention is already used by the Web innovation platforms to boost the problem-solving activity [2]. In our previous work [3] we developed a method for solver finding that leverages the user traces (e.g., blogs, publications, presentations) available in Linked Data. However, finding the users with expertise in the problem topics is often not good enough, as Web innovation platforms also seek a greater diversity in solutions in terms of domains of knowledge that they are coming from, as well as in terms of different perspectives on the problem. Existing OI research strongly argues [4] that truly innovative solutions often come from solvers whose competence is not in the topics directly found in the problem description, but rather from those who are experts in a different domain and can transfer the knowledge from one domain to another. One way to identify and involve such lateral solvers is to search for the concepts lateral to the problem. Such concepts then might be contained in the user profiles of experts likely to submit solutions, or in the possibly existing solutions in the form of research publications or patents. The key challenge thus comes down to the identification of expertise topics, directly and laterally related to the problem in question.

With the emergence of the Linked Open Data (LOD) project[1], which continues stimulating creation, publication and interlinking the RDF graphs with those already in the LOD cloud, the amount of triples increased to 31 billion in 2011, and continues to grow. The value in the linked data is the large amount of concepts and relations between them that are made explicit and hence can be used to infer relations more effectively in comparison to deriving the same kind of relations from text. We propose two independently developed methods for topic discovery based on the Linked Data. The first method called hyProximity, is a structure-based similarity which explores different strategies based on the semantics inherent in an RDF graph, while the second one, Random Indexing, applies a well-known statistical semantics from Information Retrieval to RDF, in order to identify the relevant set of both direct and lateral topics. As the baseline we use the state of the art adWords keyword recommender from Google that finds similar topics based on their distribution in textual corpora and the corpora of search queries. We evaluate the performance of these methods based on solution descriptions submitted to Hypios in the last year that we use to create the 'gold standard'. In addition, we conduct the user study aimed at gaining a more fine-grained insight into the nature of the generated recommendations.

[1] `http://linkeddata.org/`

2 State of the Art

In this section we discuss the existing measures of semantic relatedness and systems that use them in different scenarios including concept recommendation, followed by the approaches which use Linked Data.

Legacy Approaches: Although our focus is semantic relatedness of concepts our challenge is quite similar to term recommendation that has been studied for decades. Semantically related terms have been used to help users choose the right tags in collaborative filtering systems [5]; to discover alternative search queries [6]; for query refinement [7]; to enhance expert finding results [8]; for ontology maintenance [9], [10], and in many other scenarios. Different techniques and different sources are used and combined to develop Measures of Semantic Relatedness (MSRs). These measures could be split into two major categories: 1) graph-based measures and 2) distributional measures. In what follows we briefly examine each category of MSRs.

Graph-based measures make use of semantics (e.g., hyponymy or meronymy) and/or lexical relationships (e.g., synonyms) within a graph to determine semantic proximity between the concepts. For example, [11] exploits the hypernym graphs of Wordnet[2], [7] uses Gallois lattice to provide recommendations based on domain ontologies, whereas [12] uses the ODP taxonomy[3]. Some approaches (e.g. [10]) rely on the graph of Wikipedia categories to provide recommendations. Different approaches use different graph measures to calculate the semantic proximity of concepts. Shortest path is among the most common of such measures. It is often enhanced by taking into account the information content of the graph nodes [13]. To the best of our knowledge these approaches have not been applied to knowledge bases of size and richness comparable to that of DBpedia[4]. Even the Wikipedia-based measures (e.g. [10]) do not go beyond exploring categories, neither leverage the rich information inherent in DBpedia. The MSR that we propose in this paper builds upon the existing graph-based measures but is highly adapted to the rich structure of Linked Data sources, as it leverages different types of relations between the concepts in the graph.

Distributional measures rely on the distributional properties of words in large text corpora. Such MSRs deduce semantic relatedness by leveraging co-occurrences of concepts. For example, the approach presented in [14] uses co-occurrence in research papers, pondered with a function derived from the tf-idf measure [15] to establish a notion of word proximity. Co-occurrence in tags [5] and in search results [16] is also commonly used. In [17], the authors introduce Normalized Web Distance (NWD) as a generalization of Normalized Google Distance (NGD) [16] MSR and investigate its performance with six different search engines. The evaluation (based on the correlation with human judgment) demonstrated the best performance of Exalead-based NWD measure, closely followed by Yahoo!, Altavista, Ask and Google. A distributional measure applied for the task similar to ours is considered in [8], where using

---

[2] http://wordnet.princeton.edu/

[3] http://www.dmoz.org

[4] While DBpedia contains more than 3.5 million concepts, the current version of Wordnet has 206941 word-sense pairs, and ODP has half a million categories.

relevance feedback the distribution of keywords in expert profiles is used to discover new keywords that could enrich the search queries used to find experts. However, since the task was focused on finding the most relevant experts (as opposed to our focus on finding people likely to propose ideas and innovative solutions), the impact of the additional keywords was not purely satisfactory, as they tended to divert the expert search from its original focus.

In Information Retrieval, methods based on word space models can be seen as advanced distributional measures, as they are proven to be effective at finding words that appear in similar context (e.g. synonyms). That is, words that do not necessarily appear with each other, but with the same set of other words are found to be semantically related. The idea behind word space models is to use distributional statistics to generate high-dimensional vector spaces, where words are represented by context vectors. These context vectors are then used to indicate semantic similarity [18]. Examples of such methods are Latent Semantic Analysis (LSA) and Random Indexing (RI). The latter is considered more scalable and is used to discover implicit connections from large corpora such as in [19]. However, most of distributional measures are calculated based on text analysis and mining the relationships based on the distribution of words in text. In the large graphs such as the Linked Open Data cloud, the relationships already exist - the challenge is the selection of those that will lead towards more relevant concepts. Our approaches provide a ranking mechanism for this selection and finding both latent and directly related concepts, as they explore the semantics and implicit relations that exist in the large graphs.

Linked Data-Based Approaches: DBRec [20] uses Linked Data sets (DBpedia and the music-related data sets) to recommend music artists based on the specified user interest. The system proved as effective when making discoveries of relevant artists. The system uses a measure of semantic relatedness similar to our transversal strategy, but it is specific to the music domain, and works only with concepts that have the explicit type – Artist. Similarly, a video recommendation system based on DBpedia is proposed in [21] but it is also applicable for explicitly typed concept recommendations, while for our system this is not a requirement. Our general methodology is more broadly applicable, especially in cases where the desired concepts do not have explicit types.

3 Linked Data-Based Concept Recommendation Approaches

We present two Linked Data-based methods: 1) a structure-based similarity based solely on exploration of the semantics (defined concepts and relations) in an RDF graph, 2) a statistical semantics method, Random Indexing, applied to the RDF in order to calculate a structure-based statistical semantics similarity.

In general, our methods start from a set of Initial/seed Concepts (IC), and provide a ranked list of suggested concepts relevant to IC. A concept, in our context, is a Linked Data instance, defined with its URI, which represents a topic of human interest.

3.1 Structure-Based Similarity

In a typical Linked Data set covering general knowledge concepts, such as Freebase or DBpedia, links between concepts are established over two kinds of properties:

- **Hierarchical links:** The properties that help to organize the concepts based on their types (e.g., *rdf:type*[5] and *rdfs:subclassOf*) or categories (e.g., *dcterms:subject* and *skos:broader*). The links created by those properties connect a concept to a *category concept* – the one serving to organize other concepts into classes.
- **Transversal links:** The properties that connect concepts without the aim to establish a classification or hierarchy. The majority of properties belong to this group, and they create direct and indirect links between ordinary, non-category concepts.

In our concept discovery we will treat the two types of links differently, due to their different nature, and we will devise three different approaches in order to be able to work with different data sets that might or might not contain both types of links. An early version of our approach treating hierarchical links only is presented in [22].

Generic Approach. Our approach for suggesting concepts relevant to a number of dinitial seed concepts is based on two main principles:

- **Closer concepts are more relevant.** Closer concepts are those that are at a shorter distance from the seed concepts. In the sense of our work the distances in the graph are not necessarily defined as the shortest path between the two nodes, but can be measured using different distance functions. The distance functions adapted to the nature of the graph that is used are discussed later.
- **Concepts found several times are more relevant.** Concepts found by exploration of the graph proximity of several seed concepts are more relevant than those appearing in the proximity of just one starting concept.

These general principles allow a diversity of concrete approaches that differ in distance functions used as well as in the weights given to candidates found at certain distances. In the remainder of this section we examine a variety of such different approaches. The general approach to calculating our measure of semantic proximity of a concept candidate to the set of seed concepts is using Equation (1). We refer to our notion of semantic proximity as hyProximity.

$$hyP(c,IC)=\sum_{c_i \in IC} dv(c,c_i) \qquad (1)$$

HyProximity of a concept c to the set of initial concepts IC is the sum of values of the distance functions for distances between the concept c and each concept ci from the set of initial seed concepts IC. The distance value between the concept c and an initial concept ci , is denoted dv(c, ci) and is inversely proportional to the value of a chosen distance function, i.e. dv(c, ci) = p(c, ci)/ d(c, ci). Different distance functions d(c, ci) and ponderation functions p(c, ci) can be used, and we will describe some of them in the reminder of this paper. The calculation of hyProximity can be performed using

[5] All the prefixes used in this paper can be looked up at `http://prefix.cc`

the Algorithm 1. The generation of concept candidates as well as the distance value function depend on the exploration strategy used. In the following sub-sections we present a variety of strategies.

Algorithm 1.

1. *get initial topic concepts IC*
2. *for each seed concept c in IC:*
   a. *while distance_level++ < maxLevel:*
      i. *generate concept candidates for the current distance_level*
      ii. *for each concept candidate c_i:*
         1. *value(c_i) = dv(c,c_i)*
         2. *get previousValue(c_i) from Results*
         3. *put <c_i, previousValue(c_i)+value(c_i)> to Results*
3. *sort Results in decreasing order of hyProximity*

Hierarchical Distance Functions. Hierarchical approaches exploit the links established over hierarchical properties. They focus on a subset of a given data set's graph constructed only of hierarchical properties and the concepts that they connect.

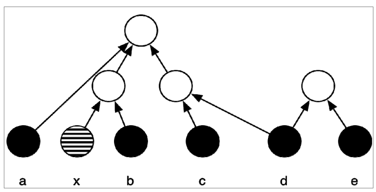

Fig. 1. A sample structure of a graph of concepts and categories

In finding candidate concepts using the hierarchical links, we can distinguish several ways to calculate distances. Our previous studies [22] allowed to isolate one particular function that gives best results, and that we will use here. Figure 1 represents an example graph of concepts (black nodes) and their categories/types[6] (white nodes), and it will help us illustrate the distance function. Our hierarchical distance function considers all the non-category concepts that share a common category with x (in the case of our example – only the concept b) to be at distance 1. To find candidate concepts at distance n, we consider each category connected to the starting concept (x) over n links, and find all concepts connected to it over any of its subcategories. In our example, this approach would lead to considering {b,c,d} as being at distance 2 from x. Different ponderation schemes can be used along with the distance functions. A standard choice in graph distance functions is to use the informational content [13] of the category (-log(p) where p is the probability of finding the category in the graph of DBpedia categories when going from bottom up) as a pondering function. Applied to our case the pondering function p(c, ci) would take as a value the informational content of the first category over which one may find c when starting from ci.

[6] For the sake of simplicity, we will refer to both categories and types, as well as other possible grouping relations used to construct a hierarchy, as categories.

As the higher level categories normally have lower informational content, this function naturally gives higher hyProximity values to concept candidates found over categories closer to the initial concepts.

Transversal Distance Function. Our transversal function relies on a subset of the data set's graph constituted of transversal properties relevant for a particular use case of interest, and the concepts that they connect. As the total number of properties in a data set might be high retrieving all the transversal links may yield time-consuming SPARQL queries. It is therefore useful to focus on those transversal properties that make connections relevant to a use case. The ways of identifying the set of useful properties for expert search are discussed in Section 4. The transversal distance function asserts the distance 1 for each link (direct or indirect) created between two concepts over one of the transversal properties. In our experiments we use the following ponderation function along with the transversal distance function: p(c,ci)= -log(n/M) where n is the number of concepts to which the candidate concept is connected over the same property that connects it to the initial concept. M is a large constant, larger than the maximum expected value of n. We use the total number of concepts in DBpedia as M in order to make the hyProximity values of the transversal strategy comparable to those of the hierarchical strategy where this same number is used to calculate the probabilities of finding a category in the graph. With such pondering function we give more importance to the concepts having a lower number of connections than to those acting as general connection hubs.

Mixed Distance Function. The mixed distance function asserts the distance n to all the concepts found at the distance n by the hierarchical function and those found at the same distance by the transversal function.

3.2 Structure-Based Statistical Semantics Similarity

Latent Semantic Analysis (LSA) [23] is one of the pioneer methods to automatically find contextually related words. The assumption behind this and other statistical semantics methods is that words which appear in the similar context (with the same set of other words) are synonyms. Synonyms tend not to co-occur with one another directly, so indirect inference is required to draw associations between words used to express the same idea [19]. This method has been shown to approximate human performance in many cognitive tasks such as the Test of English as a Foreign Language (TOEFL) synonym test, the grading of content-based essays and the categorisation of groups of concepts (see [19]). However, one problem with this method is scalability: it starts by generating a term x document matrix which grows with the number of terms and the number of documents and will thus become very large for large corpora. For finding the final LSA model, Singular Value Decomposition (SVD) and subsequent dimensionality reduction is commonly used. This technique requires the factorization of the term-document matrix which is computationally costly. Also, calculating the LSA model is not easily and efficiently doable in an incremental or out-of-memory fashion. The Random Indexing (RI) method [18] circumvents these problems by avoiding the need of matrix factorization in the first place. RI can be seen as an approximation to LSA which is shown to be able to reach similar results (see [24] and [25]). RI can be incrementally updated and also, the term x document matrix does not have to be loaded in memory at once –loading one row at the time is enough for computing context vectors. Instead of starting with the full term x document matrix and then reducing the dimensionality, RI starts by creating almost orthogonal random vectors (index vectors) for each document. This random vector is created by setting a certain number of randomly selected dimensions to either +1 or -1. Each term is represented by a vector (term vector) which is a combination of all index vectors of the document in which it appears. For an object consisting of multiple terms (e.g. a document or a search query with several terms), the vector of the object is the combination of the term vectors of its terms.

In order to apply RI to an RDF graph we first generate a set of documents which represent this graph, by generating one virtual document for each URI in the graph. Then, we generate a semantic index from the virtual documents. This semantic index is then being searched in order to retrieve similar literals/URIs. Virtual documents can be of different depth, and in the simplest case, for a representative URI S, a virtual document of depth one is a set of triples where S is a subject - in addition if any object in the set of triples is a URI we also include all triples where that URI is the subject and the object is a literal. The reason for this is the fact that literals such as labels are often used to describe URIs. A sample virtual document of depth one is shown in Figure 2, where the graph is first expanded down one level from node S. Further on, we also expand the graph from nodes O1 and O2 to include only those statements where objects are literals. A sample raw that will be added to the term x document matrix is illustrated in Table 1.

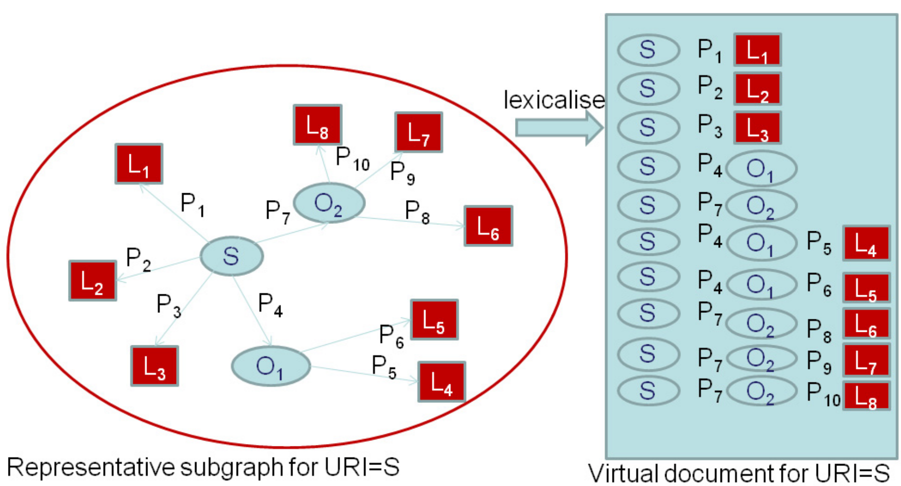

Fig. 2. From a representative subgraph to the virtual document for URI S: L - literals, O - non-literal objects (URIs), P - RDF properties

Table 1. A sample raw in the term x document matrix for the virtual document in Figure 2. The number of documents is equal to the number of URIs in the graph, and the number of terms is equal to the number of URIs and literals.

	S	P_1	..	P_{10}	L_1	..	L_8	O_1	O_2
S	10	1	..	1	1	..	1	3	4
..	..	..	..	..	..	..	..	..	..

Traditionally, the semantic index captures the similarity of terms based on their contextual distribution in a large document collection, and the similarity between documents based on the similarities of the terms contained within. By creating a semantic index for an RDF graph, we are able to determine contextual similarities between graph nodes (e.g., URIs and literals) based on their neighbourhood – if the two nodes are related with a similar set of other nodes, they will appear as contextually related according to the semantic index. We use the cosine function to calculate the similarity between the input term (literal or URI) vector and the existing vectors in the generated semantic index (vector space model). While the generated semantic index can be used to calculate similarities between all combinations of term/document-term/document, we focus on document-document search only: suggesting a set of representative URIs related to a set of seed URIs or ICs.

4 Gold Standard-Based Evaluation

In this section we describe the experiments conducted in order to compare our different approaches for concept recommendation. We used 26 real innovation problems from Hypios for which the solutions submitted in the past were available. Our assumption is that a good method for concept recommendation should be able to suggest concepts that appear in the actual solutions. Although the set of concepts appearing in the solutions does not necessarily correspond to the complete set of concepts relevant for solving a problem, it constitutes a reasonable list of concepts against which we can test performance. However, in order to better confirm our results we complement this evaluation with the user study presented in Section 5.

We use 3 **performance measures**: *precision*, *recall* and the combined *F1 measure*. In the sense of this experiment, precision is the number of relevant solution concepts suggested by the system that was found in the actual solutions divided by the number of concept suggestions proposed by the system. By recall we consider the number of relevant solution concepts suggested by the system that was found in the actual solutions divided by the total number of solution concepts known for the particular problem. The F1 score is the harmonic mean between precision and recall. It serves to compare the strategies in the case when precision and recall are equally important, and can point to approaches with the best balance of the two measures.

In order to generate the suggestions using Linked Data-inspired similarity metrics described in Section 3, we used the **DBpedia data set**, as it is arguably the most complete source of topics related to the general human knowledge, with more than 3.5 million concepts. It should be noted that our methods are also applicable to other Linked Data sets. The full DBpedia dataset is also known to have a large number of properties and hence any structure-based method is expected to be more effective if some pre-selection is conducted prior to calculating similarities. In our case, we were able to select a number of properties relevant to the Open Innovation-related scenario by analyzing the problems and solutions collected on hypios.com in the past (note that this dataset is different from the 26-problems dataset which we used in our evaluation). In order to determine this set of properties we performed DBpedia concept extraction from the text of problems and their respective solutions, using Zemanta. We then queried DBpedia to discover all the paths that connect concepts found in problems with those in the respective solutions. The output of this exercise was only a small number of properties: *dbo:product, dbp:pruducts, dbo:industry, dbo:service, dbo:genre*, and properties serving to establish a hierarchical categorization of concepts, namely *dc:subject* and *skos:broader*. We therefore boosted the concepts participated in links created over those properties in comparison to the others in DBpedia. The same method for discovering relevant subset of properties could be used to adapt the approach to other domains, provided that an initial set of input concepts and desired outputs is available.

To set up the experiment and create the 'gold standard' against which we can test our methods we prepared the data as follows:

- **Extract problem URIs.** We took the 26 problem descriptions and extracted their key concepts using a natural language processing service that links the key concepts in a given English text to the DBpedia entities. We use Zemanta[7] for this extraction, but other services such as OpenCalais[8] or DBpedia Spotlight[9] may also be used. This service has been shown to perform well for the task of recognizing Linked Data entities from text in recent evaluations [26].
- **Extract solution URIs.** For each problem we collected the submitted solutions (142 total), extracted the key concepts in the same way we did for problem texts.

The key concepts extracted by Zemanta were not verified by human users. While in the case of key concept extraction from problems this verification was feasible, in the case of solutions it was not, as it would violate the confidentiality agreement. We therefore had to work with automatically extracted and non-validated concepts, trusting that Zemanta's error rate would not affect the correctness of our further study, and that the potential impact of potential errors would equally affect all approaches. Note that when evaluating the baseline, we did not need to extract the key concepts, as the Google Keyword tool would generate a set of keywords that we could then compare to the words in the submitted solutions without any need for linking them to URIs.
As the **baseline** we used Google Adwords Keyword Tool[10]. This tool is a good candidate for baseline because it is the state of the art commercial tool employing some of the best Information Retrieval practices to text. In a legacy platform that Hypios uses for finding solvers, such a tool plays the crucial role as it is capable of suggesting up to 600 similar terms which then can be used to search for solvers. This large number

[7] `developer.zemanta.com`
[8] `http://www.opencalais.com/`
[9] `http://dbpedia.org/spotlight`
[10] `https://adwords.google.com/select/KeywordToolExternal`

of suggested terms is important for the task of Web crawling in order to find relevant experts. Hypios crawls the Web in order to identify and extract the expert information and thus enrich the existing database of experts. Google Adwords is also widely used in tasks with similar purposes such as placing the adverts for consumers relevant to the page they are viewing. Using the methods for ranking concept recommendations inspired by Linked Data, our aim is to improve the baseline. Our hypothesis is that linked data-based similarity metrics described in this paper can improve the baseline. In what follows we detail the experiments conducted to test this hypothesis.

4.1 Results

We took the key concepts extracted from the problems, and fed them to our methods and to the baseline system, which all generated an independent set of recommended concepts. We then calculated the performance for each method by comparing the results with those collected in the gold standard. The results, shown in Figure 3, indicate that the mixed hyProximity measure performs best with regard to precision. This measure should therefore be used in the end-user applications, as the users can typically consult only a limited number of top-ranked suggestions. With regard to recall, Random Indexing outperforms the other approaches for 200 top-ranked suggestions. It is especially useful in cases when it is possible to consider a large number of suggestions which include false positives - such as the case when the keyword suggestions are used for expert crawling. The balanced F-measure indicates that the transversal hyProximity method might be the best choice when precision and recall are equally important, and for less than 350 suggestions. After this threshold the mixed hyProximity is a better choice. HyProximity measures improve the baseline across all performance measures, while Random indexing improves it only with regard to recall and F-measure for less than 200 suggestions. The significance of differences is confirmed by the T-test for paired values for each two methods ($p<0.05$).

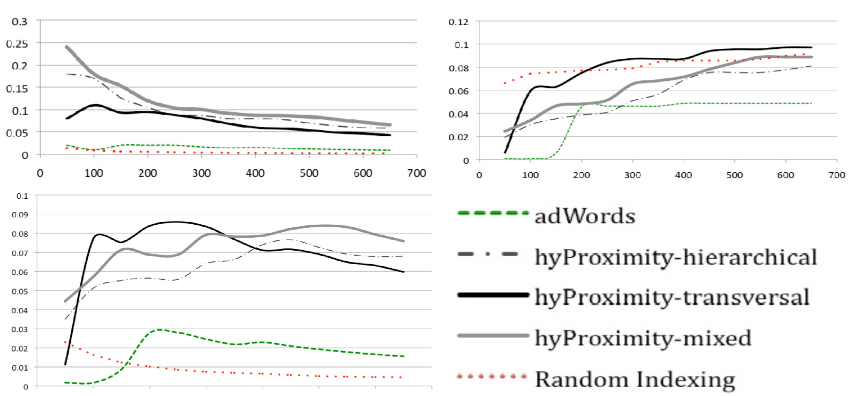

Fig. 3. Comparison of methods: precision (top-left), recall (top-right), F-measure (bottom left). On x axis: the number of suggestions provided by the systems.

The relatively low precision and recall scores for all methods, including the baseline, can be explained by the fact that our 'gold standard' is not complete : some concepts might not appear in solutions, even if relevant, as not all relevant experts were motivated to propose a solution. This is a natural consequence of the difficulty of the task. However, our evaluation with such an incomplete dataset still gives an insight into different flavors of our similarity measures, and to compensate for this incompleteness, we conduct a user-centric study in order to test the quality of the generated suggestions.

5 User Evaluation

We conducted a user study in order to cover the aspects of the methods' performance that could not have been covered by the previous evaluation. The reason is that relying on the solutions received for a particular problem gives insight into a portion of the relevant topics only, as some correct and legitimate solutions might not have been submitted due to the lack of interest in the problem prize, and in such cases our gold standard would not take such topics into account. Further on, the user study allowed a more fine-grained view on the quality of recommendations, as we focused on the following two aspects:

- **Relevancy:** the quality of a concept suggestion being relevant to the given innovation problem in the sense that the concept might lead to a potential solver of a solution of this problem if used in the expert search. We used the scale from 1 to 5: (1) extremely irrelevant (2) irrelevant, (3) not sure (4) relevant (5) extremely relevant.
- **Unexpectedness:** the degree of unexpectedness of a concept suggestion for the user evaluator on the scale from 1 to 5: (1) evident suggestions e.g. those that appear in the problem description (2) easy– suggestions that the user would have easily thought of based on the initial seed concepts (3) neutral (4) unexpected - for keywords that the user would not have thought of in the given context, however the concept is known to him (5) new unexpected - for keywords that were unknown to the user as he had to look up their meaning in a dictionary or encyclopedia.

Suggestions being both relevant and unexpected would represent the most valuable discoveries for the user in the innovation process, and a good concept recommendation system for this use case should be capable of providing such suggestions.

Twelve users familiar with OI scenarios (employees of OI companies and PhD students in OI-related fields) participated in the study. They were asked to choose a subset of innovation problems from the past practice of hypios.com and evaluate the recommended concepts. This generated a total of 34 problem evaluations, consisting of 3060 suggested concepts/keywords. For the chosen innovation problem, the evaluators were presented with the lists of 30 top-ranked suggestions generated by adWords, hyProximity (mixed approach) and Random Indexing. We then asked them to rate the relevancy and unexpectedness of suggestions using the above described scales.

The choice of our subjects was based on the two criteria. Their ability to judge the relevancy in this particular sense came out of their experience with OI problems, and at the same time they were not domain experts, but had rather general knowledge so the topics that they would judge as unexpected would most likely be also unexpected for an average innovation seeker from a client company.

Table 2. Average note ± standard deviation obtained in the study

Measure	adWords	hyProximity (mixed)	Random Indexing
Relevance	2.930±0.22	3.693±0.23	3.330±0.25
Unexpectedness	2.859±0.16	2.877±0.25	3.052±0.22
Unexpectedness (relevancy >=4)	*2.472±0.31*	*2.542±0.36*	*2.635±0.36*
Unexpectedness (relevancy =5)	*1.760±0.22*	*1.842±0.31*	*1.767±0.36*

As shown in Table 2, the Linked Data measures outperform the baseline system across all criteria. While hyProximity scores best considering the general relevance of suggestions in isolation, Random Indexing scores best in terms of unexpectedness. With regard to the unexpectedness of the highly relevant results (relevancy>=4) Random indexing outperforms the other systems, however hyProximity offers a slightly more unexpected suggestions if we consider only the most relevant results (relevancy=5). We tested the differences in relevance for all methods using the paired T-test over subjects individual means, and the tests indicated that the difference in relevance between each pair is significant ($p < 0.05$). The difference in unexpectedness is significant only in the case of Random Indexing vs. baseline. This demonstrates the real ability of Linked Data-based systems to provide the user with valuable relevant concepts.

In the follow up study, we asked the raters to describe in their own words, the suggestions they were presented with from each system (identified as System 1, 2, and 3). The adjective most commonly used to describe adWords suggestions was "redundant" and "Web-oriented". This indeed corresponds to the fact that the system is not fully adapted to the OI scenario, but also to the fact that it is based on a statistical approach, which is more influenced by the statistical properties of Web content, than by the meaning of things. HyProximity suggestions were most commonly described as "really interesting" and "OI-oriented", while the suggestions of Random Indexing were most often characterized as "very general". According to the preference towards more general or more specific concepts, it is therefore possible to advise the user with regard to which of the two methods is more suitable for the specific use case.

To illustrate the qualitative aspects of suggestions we provided an example of concept suggestions from all 3 systems on our website[11].

6 Conclusion

We presented two Linked Data-based concept recommendation methods and evaluated them against the state of the art Information Retrieval approach which served

[11] `http://research.hypios.com/?page_id=165`

as our baseline. We argue that our methods are suitable in an Open Innovation scenario where the suggested concepts are used to find potential solvers for a given problem. Our results show that both proposed methods improve the baseline in different ways, thus suggesting that Linked Data can be a valuable source of knowledge for the task of concept recommendation. The gold standard-based evaluation reveals a superior performance of hyProximity in cases where precision is preferred; Random Indexing performed better in case of recall. In addition, our user study evaluation confirmed the superior performance of Linked Data-based approaches both in terms of relevance and unexpectedness. The unexpectedness of the most relevant results was also higher with the Linked Data-based measures. Users also indicated that Random Indexing provided more general suggestions, while those provided by hyProximity were more granular. Therefore, these two methods can be seen as complementary and in our future work we will consider combining them as their different nature seem to have a potential to improve the properties of the query process.

Acknowledgments. The work of Milan Stankovic is partially funded by the grant CIFRE N 789/2009 given by a French research funding agency ANRT. Special thanks to employees of Open Innovation companies Hypios and Bluenove, as well as the PhD students from the same field from the University Paris Dauphine, for their participation in the evaluation.

References

1. Chesbrough, H.W.: Open Innovation: The New Imperative for Creating and Profiting from Technology. Harvard Business Press (2003)
2. Speidel, K.-P.: Problem-Description in Open Problem-Solving. How to overcome Cognitive and Psychological Roadblocks. In: Sloane, P. (ed.) A Guide to Open Innovation and Crowdsourcing. Advice from Leading Experts. KoganPage, London (2011)
3. Stankovic, M., Jovanovic, J., Laublet, P.: Linked Data Metrics for Flexible Expert Search on the Open Web. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 108–123. Springer, Heidelberg (2011)
4. Jeppesen, L.B., Lakhani, K.R.: Marginality and Problem Solving Effectiveness in Broadcast Research. Organization Science 20 (2009)
5. Sigurbjörnsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: Proceeding of the 17th International Conference on World Wide Web, WWW 2008, p. 327. ACM Press, New York (2008), doi:10.1145/1367497.1367542
6. Mei, Q., Zhou, D., Church, K.: Query suggestion using hitting time. In: Proceeding of the 17th ACM Conference on Information and Knowledge Mining - CIKM 2008, New York, USA, p. 469 (2008)
7. Safar, B., Kefi, H.: OntoRefiner, a user query refinement interface usable for Semantic Web Portals. In: Proceedings of Application of Semantic Web Technologies to Web Communities Workshop, ECAI 2004, pp. 65–79 (2004)
8. Macdonald, C., Ounis, I.: Expertise drift and query expansion in expert search. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management - CIKM 2007, p. 341. ACM Press, New York (2007)
9. Cross, V.: Semantic Relatedness Measures in Ontologies Using Information Content and Fuzzy Set Theory. In: Proc. of the 14th IEEE Int'l Conf. on Fuzzy Systems, pp. 114–119 (2005)

10. Gasevic, D., Zouaq, A., Torniai, C., Jovanovic, J., Hatala, M.: An Approach to Folksonomy-based Ontology Maintenance for Learning Environments. IEEE Transactions on Learning Technologies (2011) (in press)
11. Burton-Jones, A., Storey, V.C., Sugumaran, V., Purao, S.: A Heuristic-Based Methodology for Semantic Augmentation of User Queries on the Web. In: Song, I.-Y., Liddle, S.W., Ling, T.-W., Scheuermann, P. (eds.) ER 2003. LNCS, vol. 2813, pp. 476–489. Springer, Heidelberg (2003)
12. Ziegler, C.-N., Simon, K., Lausen, G.: Automatic Computation of Semantic Proximity Using Taxonomic Knowledge Categories and Subject Descriptors. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM 2006, Arlington, Virginia, USA, pp. 465–474. ACM, New York (2006)
13. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy (1995)
14. Matos, S., Arrais, J.P., Maia-Rodrigues, J., Oliveira, J.L.: Concept-based query expansion for retrieving gene related publications from MEDLINE. BMC Bioinformatics (2010)
15. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
16. Cilibrasi, R.L., Vitanyi, P.M.B.: The Google Similarity Distance. IEEE Transactions on Knowledge and Data Engineering 19(3), 370–383 (2007), doi:10.1109/TKDE.2007.48
17. Gracia, J., Mena, E.: Web-Based Measure of Semantic Relatedness. In: Bailey, J., Maier, D., Schewe, K.-D., Thalheim, B., Wang, X.S. (eds.) WISE 2008. LNCS, vol. 5175, pp. 136–150. Springer, Heidelberg (2008)
18. Sahlgren, M.: An introduction to random indexing. In: Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005 (2005)
19. Cohen, T., Schvaneveldt, R., Widdows, D.: Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. Journal of Biomedical Informatics (2009)
20. Passant, A.: dbrec — Music Recommendations Using DBpedia. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part II. LNCS, vol. 6497, pp. 209–224. Springer, Heidelberg (2010)
21. Waitelonis, J., Sack, H.: Towards Exploratory Video Search Using Linked Data. In: 2009 11th IEEE International Symposium on Multimedia, pp. 540–545. IEEE (2009), doi:10.1109/ISM.2009.111
22. Stankovic, M., Breitfuss, W., Laublet, P.: Linked-Data Based Suggestion of Relevant Topics. In: Proceedings of I-SEMANTICS Conference 2011, Gratz, Austria, September 7-9 (2011)
23. Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41, 391–407 (1990)
24. Karlgren, J., Sahlgren, M.: From words to understanding. In: Uesaka, Y., Kanerva, P., Asoh, H. (eds.) Foundations of Real-World Intelligence, pp. 294–308. CSLI Publications, Stanford (2001)
25. Cohen, T.: Exploring medline space with random indexing and Pathfinder networks. In: Annual Symposium Proceedings/AMIA Symposium, pp. 126–130 (2008)
26. Rizzo, G., Troncy, R.: NERD: Evaluating Named Entity Recognition Tools in the Web of Data. In: ISWC 2011 Workshop on Web Scale Knowledge Extraction (WEKEX), Bonn, Germany (2011)