Actively Learning to Rank Semantic Associations for Personalized Contextual Exploration of Knowledge Graphs

Federico Bianchi, Matteo Palmonari, Marco Cremaschi and Elisabetta Fersini

University of Milan - Bicocca, Viale Sarca 336, Milan, Italy
federico.bianchi,palmonari,cremaschi,fersiniel@disco.unimib.it

Abstract. Knowledge Graphs (KG) provide useful abstractions to represent large amount of relations that occur between entities of different types. Chains of such relations represented by semantic associations reveal connections between entities that may be interesting, and possibly unknown, to a user, thus resulting valuable for her to get insights into a given topic. For example, in contextual exploration of KGs, we may want to let a user explore a set of semantic associations that are deemed to be more interesting for her while she is reading an input text. Using well-known techniques to bridge the gap between the input text and the KG a associations can be found and presented to the user. However, because of the large number of diverse associations that can be found, a critical challenge is to effectively rank the associations so as to present the ones that are estimated to be most interesting. In addition, since users may have different interests, this ranking function should be adapted to match users' preferences to personalize the exploration of KG information. In this paper We describe an active learning to rank model to let a user rate a small sample of associations, which is used to learn a ranking function that optimize the user preferences. To the best of our knowledge this is the first attempt to use active learning to rank techniques to explore semantic associations. Experiments conducted with several data sets show that the approach is able to improve the quality of the ranking function with a limited number of user interactions.

1 Introduction

Knowledge Graphs (KG) provide an abstraction to describe knowledge by representing entities, their properties and binary relations that interconnect these entities. KGs are today frequently used to support interoperability among applications in the industry as well as in the research community. Languages like RDF and SPARQL have been also proposed to publish KGs and make them publicly available as linked data.

A problem that has recently gained attention is how to exploit the vast amount of knowledge available in proprietary or open KGs to deliver useful information to the users. While query answering is aimed at satisfying specific

information needs, knowledge exploration includes mechanisms to deliver information that is estimated to be interesting for the users in a proactive fashion [1]. Few approaches have been proposed to use information from a KG in the context of a task that is carried out by the user. For example, the Google Knowledge Graph is currently used to extend the results of a web search with additional information not explicitly required by a user. Other approaches use relations in the KG to enrich the information delivered to a user who is reading an input text[2]. We refer to this approach as contextual knowledge exploration, where a text is used as input to find relevant information in the KG. By using well-known entity linking techniques, we can link entities mentioned in the text to a KB. If more than one entities are found, semi-walks in the KG that connect the two entities, i.e., semantic associations of finite length between the two entities [3, 4] reveal connections between entities that may be interesting, because they provide - possibly new - insights into the topic of the input text. For example, when reading a news article about US Election 2016, from mentions of Hilary Clinton and Donald Trump, an association of length equal to two found in DBpedia reveals that Clinton and Trump have been both members of the Democratic Party. From surveys collected among users, we found that many Italians, for example, do not know that Trump has been member of the Democratic party before being elected as member of the Republican Party. The main problem arising in contextual KG exploration is that a very large amount of associations can be found between a set of entities extracted from even relatively short text. For example, from 12 entities extracted from the starting piece of an article[1] of 74 words, as many as 738 associations are found in DBpedia with DaCENA[2], a prototype for contextual KG exploration in the data journalism domain [4]. The crucial research problem that needs to be addressed to exploit the large amount of semantic associations represented in KGs is to provide effective methods to identify those few associations that are more interesting for the users. Several approaches have been proposed that use measures based on graph analytics to rank associations [5]. Given a certain ranking over the associations, users can look at the associations in the order defined by the ranking or at a set of top-k associations. A different approach proposes to use the associations ranked in order of interest by a set of users to train a learning to rank model so as to maximize users' preferences [5]. However, the key hypothesis that motivates the work presented in this paper, is that *different users may be interested in different kinds of associations*, which suggests that the ranking function should be optimized based on the preferences of individual users. In the latter case we cannot expect that users label a large number of associations, which is required to learn the ranking function. Starting from these observations the work addressed in this paper addresses the following research questions: Q1) Can we learn to rank semantic associations by collecting a small number of labels from a user, so that we can personalize the content delivered to her based on her preferences?

[1] http://www.nytimes.com/2016/12/05/business/italy-referendum-euro-markets.html

[2] www.dacena.org

Q2) Do we need personalization in contextual exploration of KGs with semantic associations, or can we assume that different users are interested in the same content?

To answer to Q1, we propose an active learning to rank model to reduce the number of associations that need to be labeled by the user. The model comprises: 1) a pay-as-you-go workflow to incrementally collect labels from a user and learn to rank the associations based on her preferences using the RankSVM algorithm; 2) algorithms to actively select the associations that the user has to label; 3) different approaches to select the first set of associations that the user has to label, thus solving a cold start problem generated by the above mentioned active sampling algorithms, 4) a set of features based on KG analytics to represent associations and support the model. To evaluate the effectiveness of the proposed model under different configurations and against different baselines, we have built two data sets consisting of ratings given by different users on a complete set of associations extracted for different pieces of news articles. Results show that the proposed approach is feasible and provide a consistent improvement of the ranking quality with a limited number of interactions. To answer to Q2, we measure the agreement among ratings given by different users to associations found for the same articles. Results clearly show that different users are interested in different content, thus confirming the need for personalization methods in contextual KG exploration.

To the best of our knowledge this is the first attempt to use active sampling to learn to rank semantic associations, thus improving on state of the art approaches that require a large number of labels to learn a ranking function over semantic associations. The paper is organized as follows: in Section 2, we further motivate the proposed approach by discussing contextual KG exploration, with an example of application; in Section 3, we explain our active learning to rank model; in Section 4 we describe the experiments conducted to evaluate our model; in Section 5 we discuss related work, while in Section 6 we draw some conclusions and discuss future work.

2 Contextual KG Exploration

Information extracted from a reference KGs can enrich the experience of a user while she is accessing content she is interested in, for example a news article, or multimedia content. This information can expand the user knowledge with unknown information on a given topic or help her better understand the content she is accessing. In addition, the input content tell us something about current interests of the user, thus providing a starting point to select pieces of information from the KG that are valuable for the user. Named Entity Recognition and Linking (NEEL) techniques [6] help bridging the gap between an input text (e.g., a news article or text associated with multimedia content) and the KG. Starting from a set of entities extracted from the input text, semantic associations connecting any of these two entities are defined as *semi-walks in the reference KG* [3]. We consider only relations between entities and avoid loops. By merging

every relation that occurs in some of the retrieved associations, we obtain a *subgraph* of the reference KG, which is related to the input text. To better illustrate this idea, we describe DaCENA (Data Context for News Articles), an application that supports exploration of KGs in the domain of data journalism [4]. DaCENA has been developed by our team in collaboration with DensityDesign (a lab of information visualization design from Politecnico di Milano, Italy) to target one of the objectives of data journalism, described by the following quote: "*In an age of big data, the growing importance of data journalism lies in the ability of its practitioners to provide context, clarity and, perhaps most important, find truth in the expanding amount of digital content in the world*" [7]. DaCENA is aimed to provide additional information (a data context) to a user who is reading a news article, in the form of a set of semantic associations extracted from a KGs. The DaCENA framework consists of two main components: 1) Text & Data Analyzer and 2) Contextual Explorer. The first is a component responsible for processing and storing the information used to build the context of an article; Contextual Explorer, is an interactive user interface that let the user read the articles enriched with semantic associations extracted from a KG (Figure 1).

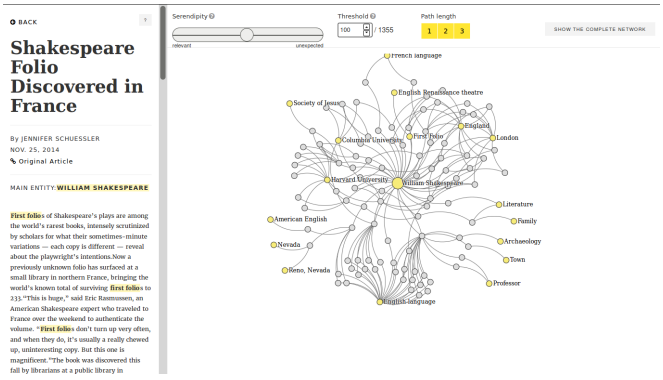

Fig. 1. DaCENA Interface

In DaCENA we are currently using TextRazor[3] as NEEL tool and DBpedia[4] as reference KG. Once entities are extracted we make several queries to the DBpedia SPARQL endpoint to extract the semantic associations that connect all these entities (we use a SPARQL endpoint to ensure that we retrieve up-to-date information). While we started with extracting associations of maximum length equal to three, now we consider only associations of maximum length equal to two, because we found - with preliminary user studies - that associations of length greater than two are seldom considered interesting by the users. Otherwise, while we previously found every association from a principal entity to every other entities, we now consider shorter associations between every entity extracted from the text. Processing an article may require significant amount of

[3] https://www.textrazor.com/
[4] https://dbpedia.org/

time (up to thirty minutes) if semantic data are fetched by querying a SPARQL endpoint as we currently do. Therefore, texts and data are processed off-line so as to make the interactive visualization features as much fluid as possible. DaCENA currently uses a measure for evaluating the *interestingness* named **serendipity** [4]. Serendipity is defined as a parametric linear combination of a measure that evaluates the relevance of an association with respect to a text, and a measure that evaluates how much an association may be unexpected for the user. An association is relevant if the virtual document built by concatenating the abstracts of each entity occurring in the association is similar to the given text. Instead, an association is unexpected when it is composed by properties that are rare, i.e., not frequently used, in the KG. Let α be a parameter used for balancing the weight of each measure, and *text* be the input text; the serendipity $S(\pi)$ of an association π, is computed by the following formula:

$$S(\pi, text) = \alpha * relevance(\pi, text) + (1 - \alpha)\ rarity(\pi)$$

In the interface shown in Figure 1 the user can see the graph and adjust the serendipity parameter to favor relatedness or unexpectedness. By using DaCENA with several articles, we could observe that a large amount of associations can be extracted even for a small text; for example from an article about politics[5] we extracted a 3500 associations; serendipity can be used to rank associations and filter the set of k most interesting associations that are shown to the users, where k can be set by the user herself (preliminary user studies suggest that users do not want to look at more than 100 associations).

3 Active Learning To Rank for Semantic Associations

To describe our approach to actively learn to rank semantic associations, we first explain the workflow and the algorithms we use and test. Then we describe the features that we use to represent semantic associations. Figure 2 describes the workflow used in our model, which is based on a learning loop. The entry point (step 1) is a *bootstrapping* phase where we select the first associations that the user has to label. The user labels the associations selected in the bootstrapping step (step 2) using a graded scale, e.g., $< 1, 2, 3, 4, 5, 6 >$, where higher grades represent higher interest for an association. We will refer to labels provided by users also as *ratings*. Then we use these labels to train a learning to rank algorithm (step 3), which ranks all the associations by assigning them a score. If the user decides that she is satisfied with the ranking obtained so far, the loop stops. Else, we proceed to further improve the ranking by collecting more labels using active sampling (step 5). In active sampling, observations are selected with the aim of optimizing the ranking function with the as few labels as possible. To find the observations for which labels are estimated to be more informative, active sampling algorithms use the scores determined by the learned ranking

[5] http://www.nytimes.com/2012/10/26/us/politics/for-obama-aides-endgame-takes-grunt-work-and-math.html

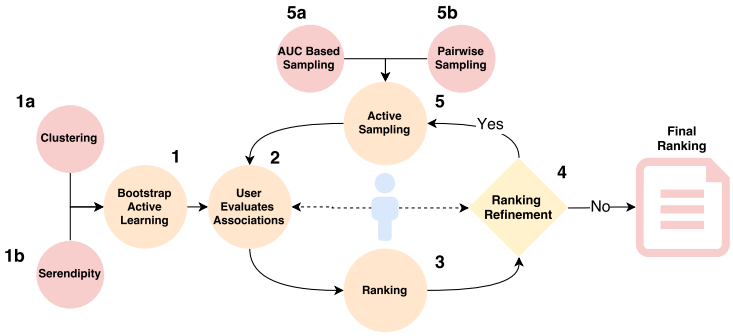

Fig. 2. High level example of our model

function. This prerequisite, motivates the need for introducing a bootstrapping step (Step 1) where labels are not selected using active sampling. After Step 5, we close the loop by repeating Step 2, in which the user labels the new associations selected in Step 5. The labels provided by the user are used to train again the learning to rank algorithm and return a new ranking (Step 3). Observe that after the first iteration, the observations that the user has to label are always selected using active sampling. Here below we further discuss the main steps of the loop, i.e., steps 1, 3 and 5.

Step 1: Bootstrap Active Learning Two approaches are proposed: the first one (alternative 1a) is based on clustering algorithms while the second one uses an heuristic ranking function (alternative 1b). The latter has the advantage that we have an ordered set of associations (and, hence a small set of interesting associations) to present to the user even before she provide any label.

Alternative 1a: the assumption at the basis of this approach is the following: observations that would be rated in a similar way by a user are spatially near in the feature space used to represent the associations, while observation rated in different ways should be distant in this space. Based on this assumption, the best way to quickly collect the training data is to cluster the data set and take the most representative observation for each cluster (the observation nearest to the mean of the cluster). Approaches similar to this have been already considered in active learning settings [8, 9], in which clustering is used to find the first observations for machine learning models. We test two clustering algorithms: the first one is the Dirichlet Process Gaussian Mixture Model [10] (Dirichlet) that has been chosen for its ability to automatically find the best number of cluster inside the data set; this is useful because we cannot know a-priori which could be the correct number of cluster for a given set of associations. We also evaluate a second clustering algorithm, the Gaussian Mixture Model (Gaussian) [11]. It is important to notice that these algorithms select associations that are representative of a data set, but that does not mean that these associations are also meaningful or interesting for a user.

Alternative 1b: another approach that we propose to use for bootstrapping the model (also in consideration of the latter remark) is to use an heuristic ranking measure. In this way, not only we can show to a user a set of associations even before she provides any label, but we can also ask their ratings on a set of associations that are heuristically believed to be interesting for them. In the context of contextual KG exploration, this may be desirable to improve the user experience, when compared to asking ratings on a set of uninteresting associations. In particular, as heuristic ranking function, we use the *serendipity* measure defined in Section 2.

Step 3: Ranking In this phase we need to train a learning to rank algorithm that can helps us ordering associations. Usually learning to rank is divided in three approaches[12]: **point-wise**, **pair-wise** and **list-wise**. In our approach we use the RankSVM [13] algorithm, which implements a pair-wise approach where the ranking problem is transformed into a pair-wise classification problem [12]. RankSVM is a variant of SVM created for learning to rank tasks, it takes ratings (based on a graded scale) over a set of the domain items as input and use these ratings to infer labels for a set of item pairs. An item pair is assigned a label equal to 1, if the first item of the pair should be ranked higher than the second one, and equal to -1, otherwise. This is the binary input of the inner algorithm used to learn the ranking function. This function assigns a score (a real number) to each item by generalizing the binary input.

Step 5: Active Sampling We implemented two supervised active sampling algorithms proposed in the document retrieval field. Both the algorithms use a pairwise approach, meaning that they can be directly used on pairwise learning to rank algorithms.

Alternative 5a: the first algorithm [14] (denominated *AUC Based Sampling*, or, shortly, *AS*, in the next sections) tries to optimize the Area Under the learning Curve (AUC), by selecting individual observations for labeling, without explicitly comparing every pair of domain items. The algorithm is thus known to be sub-optimal, runs efficiently. It is essentially based on the computation of the estimated probability of a binary class for an observation (thus, it was used in a binary setting). The algorithm uses a parameter λ to calibrate the weight of two different probability estimations.

Alternative 5b: the second algorithm [15] (denominated *Pairwise Sampling*, or, shortly, *PS*, in the next sections) explicitly compares pairs of associations to select the most informative pairs. The most informative pairs are the ones that maximize two measures: Local Uncertainty (LU), which estimates the uncertainty of the relative order within the pair, and Global Uncertainty, which estimates the uncertainty of the position of each element of the pair within the global ranking. A parameter p is used to tune the weight assigned to the LU measure. In this case, users are then asked to rate each association in the most informative pairs. With this approach, we can evaluate if the uncertainty score used to select the pair is incoherent with user ratings, thus providing more informative labels to RankSVM. The explicit generation of the observation-pairs

makes this algorithms less efficient than AUC-based Sampling, which may prevent its application to the exploration of a large number of associations.

3.1 Features

To represent the semantic associations inside our platform we used different measures. In this way we are able to define feature vectors for the active learning to rank algorithms. We normalize data extracted with these measure using a standard normalization techniques by removing the mean and scaling to unit variance.

Global PageRank We use the data in [16] to collect a global score of the PageRank inside DBpedia. In this way we are able to get an overall value of the importance of an entity inside the KG. The global pagerank of an association is computed as the average global pagerank of every entity occurring in the association.

Local PageRank We compute PageRank [17] on the sub-graph, defined by the associations extracted from an input text, to measure the *centrality importance* of each entity (hence the name of *local pagerank*). Local pagerank is computed as the global pagerank but on the sub-graph.

Local HITS We ran the HITS (Hyperlink-Induced Topic Search)[18] algorithm to compute two scores for each node of the local graph. The first, authority score, should indicate how much a node is *important*, while the second, hub score, should indicate nodes that points to nodes with an high authority score. The measure gives two score for each association: one for the average of the authority values and one for the average of the hub values.

Temporal Relevance Using the Wikimedia API we extract the number of time a wikipedia entity (page) as been accessed in a specific date (date of the publication of a given text, for example). In this way we are able to get a value of importance related to timing. For example, if we consider Wikipedia access[6] on the page Paris, we see that the entity has been accessed 8.331 times on 12-11-2015 and 171.988 times on 14-11-2015, when on 13-11-2015 there have been terrorist attacks in Paris. The temporal relevance for an association is given by the average score of the temporal relevance of the entities

Relevance This measure is based on the similarity between the concatenation of the abstract of the entities in an association and the input text that was used to extract all the associations. The measure was previously defined in [4]. The similarity can be computed with different similarity measure, in this context similarity is computed using cosine similarity weighting the terms using TF-IDF.

Path Informativeness We use a measure defined in previous work [3] of path informativeness, which is based on the concept of Predicate Frequency Inverse Triple Frequency (PF-ITF), that is used to rank path in order of informativeness.

Path Pattern Informativeness We use a measure on path patterns, defined in [3], to get the informativeness of patterns extracted from paths.

[6] http://tools.wmflabs.org/pageviews/

Rarity This measure computes how much a path is rare using the global frequency of its property inside the DBpedia KG. It was previously defined in [4].

4 Experiments

The purpose of the experimental evaluation is to validate the hypothesis that personalization is important in KG exploration, to evaluate the performance of the proposed model, and to compare alternative approaches proposed for different steps of the model. The experiments were run on a machine with a Intel Core i5 (4th Gen) (1.6Ghz).

4.1 Experimental Settings

To test our model we built two different datasets, each one consisting of triples $< text_i, A_i, ratings_{u,i} >$, where $text_i$ is a text extracted from a news article retrieved from online news platforms like NyTimes and The Guardian, A_i is the set of all associations extracted for $article_i$ with our tool DaCENA, and $ratings_{u,i}$ contains the labels assigned by a user u to every association in A_i. From each triple in a dataset, we can derive a complete ranking of the retrieved associations for one user, i.e., an ideal ranking to use in our evaluation.

Asking to a user to rate a large number of associations is demanding and tedious for her, and may downgrade the quality of the labels because of fatigue bias. Otherwise, we would like to collect a sufficient number of ratings for different articles. Thus, in the first dataset, we selected relatively short, semantically self-contained, texts (usually one long initial paragraph) extracted from five articles, for which we could retrieve a number of associations usually lower than 100. Associations were extracted from 5 different articles and for each articles we extracted, respectively 51,60,98,44 and 113 associations. Data was collected from 14 users, so some of the users took the survey more than one time. User ratings have been collected through an online form, through which users were asked to read the text and rate semantic associations based on their interest. Users were also asked questions to let us profile, e.g., their education degree, mastering of the English language, and knowledge about the text topic. For ratings we choose a graded scale from 1 to 6, following guidelines suggested in a recent study [19]. Differently from a five-valued ordinal scale, this scale provides a symmetric range that clusters scores in two sets: scores with a negative tendency (1, 2 and 3) and scores with a positive tendency (4, 5 and 6). As a result, we obtained a dataset consisting of 25 different ratings, referred to as *smaller dataset* in the following (for the small number of associations in each ranking). However, we want also to evaluate if results obtained over small association sets are comparable with results obtained with (and thus generalizable to) large association sets. To this end, we use ratings provided by two different students with a background in communication sciences, who were asked to rate thousands of associations extracted for two full-length articles, with the goal of evaluating heuristic functions used in an early versionof DaCENA. In this case, we used a three-valued

scale for ratings, from 1 to 3. We refer to this dataset, that consists in three complete rankings, as to *larger dataset* (for the larger number of associations in each ranking).

Using the ideal rankings in the two gold standards, we meaasure the quality of the rankings returned by our model at different iterations using Normalized Discounted Cumulative Gain (nDCG) computed over the top-10 ranked associations, denoted by nDCG@10. In addition, we compute the Area Under the nDCG@10 Curves (AUCs) as an aggregate performance measure. At each iteration of active sampling, we use two and six user labels in the smaller and the larger datasets, respectively. The motivation for this specific choice is given below, when discussing details about the model configurations. We carry out experiments in two different settings: in *Contextual Exploration Settings*, we consider the workflow as implemented in a system that supports contextual exploration: the set from which we select the observations to label is the same set used to evaluate the performance of the model. In this settings, we make sure that observations labeled during previous iterations are not labeled a second time by the user). In *Cross Validation Settings*, we split the associations in two sets: training and test. In this settings, which was used also in previous work [14], active sampling always picks associations from the training set. Although not amenable in contextual KG exploration, this approach is helpful to evaluate the robustness of the model. In fact, we can use 2Fold-Stratified Cross Validation to make sure that results can be reasonably generalized and do not depend on specific data. Random algorithms will be run multiple times to stabilize values.

4.2 Configurations and Baselines

We evaluate different configurations of the model, based on the alternative algorithms proposed in two steps of the loop. For Bootstrap Active Learning, we consider three approaches: two clustering algorithms (Gaussian vs. Dirichlet), and the Serendipity heuristic function, for which we set $\alpha = 0,5$. For Active Sampling, we consider two algorithms: AUC Based Sampling [14] (AS) and Pairwise Sampling [15] (PS). Parameters of these two algorithms have been determined experimentally, and set to $\lambda = 0.8$ and $p = 1$. The six configurations of the active learning to rank workflow described above are compared also against three different baselines:

- *Random + Random*: RankSVM is still used to learn a ranking function, but is trained using ratings assigned to associations that are randomly selected, both in the bootstrap and active sampling steps.
- *Serendipity No-AL*: we consider the ranking determined with Serendipity, which is not based on active learning and does not change across iterations.
- *Random No-AL*: we consider random rankings of associations, which are not based on active learning and do change across iterations.

Configuration Details In the small data sets Dirichlet Clustering and Gaussian Clustering, in the first iterations, selected an average number of clusters equal

to 3 (and thus, an average number of 3 associations are selected from this two methods in the first iteration); for this reason, to feed the model with an equal number of observation, on the average, for both Serendipity and Random Active Learning we choose to select 3 associations to be labeled. In the smaller data sets, for the supervised active learning to rank, we select 2 associations to be evaluated for each iteration. The clustering algorithms in the larger data sets selected an average number of cluster equal to 5, so, in this case, for both Serendipity and Random Active Learning we choose to select 5 associations to be labeled. After the first iteration, from both AUC Based Sampling a techniques we selected 6 observation to be labeled (since we have more data). The small dataset were obtained by recent article and thus we could use the *temporal relevance measure*. This wasn't possible for the larger ones. We used a RankSVM with polynomial kernel on the large data sets that was able to output the results of a single iteration in what we considered interactive time (less then 2 seconds).

4.3 Results and Discussion

Before discussing the results obtained in contextual exploration and cross validation settings, we discuss valuable insights gained from the analysis of the user ratings collected while creating the gold standards.

User interests and personalization. We have measured Inter-Rater Reliability [20] (IRR) to assess the usefulness of personalization within this context. Our idea is based on the assumption that different users are interested in different things. IRR was computed on the data sets that had the same associations rated by different users, results in Figure 2. We used two measures: Krippendorff's alpha, weighted using an ordinal matrix, and Kendall's W. We can see that for all the five texts used in this experiment, IRR is low and distant from 1, the value that usually represent unanimity between the raters. We also show the distribution of the ordinal scores for the data sets in table 1.

Degree	**D1**	**D2**
1	23.7%	67.4%
2	14.5%	30.1%
3	22.3%	2.5%
4	20.9%	NaN
5	10.1%	NaN
6	5.5%	NaN

Table 1. data sets' degrees distribution

Measure	**Average on Data sets**
Krippendorff's alpha	0.06154
Kendall's W	0.2608

Table 2. IRR score

Contextual Exploration Settings In this setting the active learning algorithms Figure 3 where able to perform better than the baseline considered (we can notice that active learning approaches completely outperform the non active learning ones). We then computed the area under the curve, considering the nDCG@10, of the algorithms to summarize the graph (Table 4), the algorithm that performs better is the one that uses serendipity for the bootstrap step and AUC based sampling for the active sampling step.

Cross Validation Result we obtained in the cross validation settings reported

that the best measure was the serendipity heuristic, combined with the use of the AUC based measure; this is similar to what we have seen in the previous experiment settings, with a decrease in performance probably due to the fact that the active learning algorithms are not able to access to the test data. The plots can be found in Figure 4 while the computed areas are in Figure 4.

Algorithm	**Area D1**	**Area D2**
Gaussian AS	3.0168	2.6455
Dirichlet AS	3.0011	2.6872
Gaussian PS	2.9975	NaN
Dirichlet PS	3.0009	NaN
Serendipity AS	**3.0742**	**2.711**
Serendipity PS	3.0302	NaN
Random Random	2.976	2.6013

Table 3. Area of the lines of the algorithms for Cross Validation

Algorithm	**Area D1**	**Area D2**
Gaussian AS	3.0242	2.747
Dirichlet AS	3.0711	2.7174
Gaussian PS	2.9629	NaN
Dirichlet PS	3.019	NaN
Serendipity AS	**3.2018**	**3.0817**
Serendipity PS	3.1399	NaN
Random Random	2.9359	2.673
Serendipity No-AL	2.7199	2.734
Random No-AL	2.3199	1.7971

Table 4. Area of the lines of the algorithms for Contextual Exploration Setting

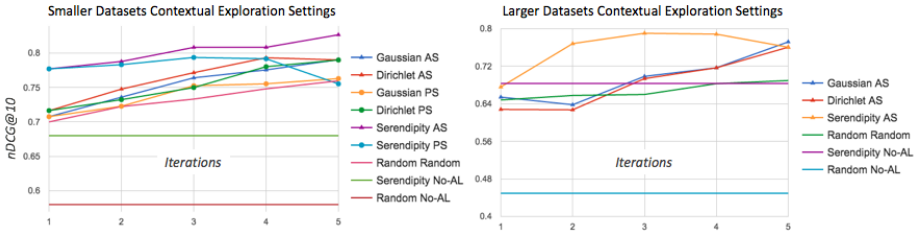

Fig. 3. Contextual exploration setting results

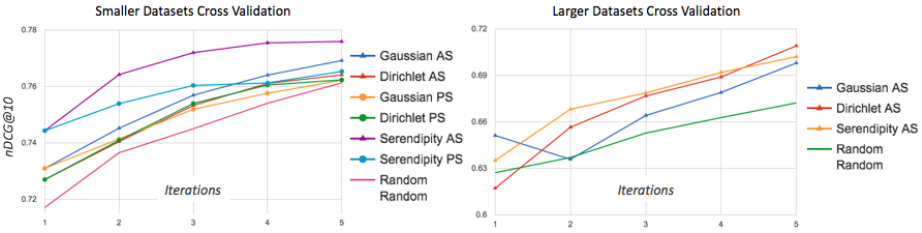

Fig. 4. Cross Validation setting results

4.4 Bootstrap Active Learning Analysis

Finding observation with different labels (operation that is needed to build the learning to rank model) since the first iteration isn't always possible. We evaluated the *time for first iteration* value, that corresponds to the average number of iteration needed for each method to have a training useful for training the learning to rank algorithm. The worst case of these algorithms is represented by a user who gives the same score to every observation. The result are visible

in table 5. We show data for both the Cross Validation (CV) and Contextual Exploration Setting (CE). We can see that with the larger data sets the finding of the first observation becomes more difficult for the algorithm except for the Dirichlet approach that can probably adapt itself to the dataset in an easier way.

Algorithm	**#Iter. D1 CV**	**#Iter. D2 CV**	**#Iter. D1 CE**	**#Iter. D2 CE**
Dirichlet	1.16	1.11	1.010	1.063
Gaussian	1.08	1.16	1.066	1.381
Serendipity	1.08	1.5	1.346	1.1363
Random	1.25	1.5	1.1866	1.229

Table 5. Time for first iteration on the data sets

5 Related Work

We compare our work to previous work in the field of interactive KG exploration and of learning to rank approaches to KG exploration.

Interactive Knowledge Graph Exploration There has been a lot of interest in literature for KG exploration, in particular in the context of linked data. Several methods, described and compared in a recent survey [1], combine navigation, filtering, sampling and visualization to let users explore large data sets. For example, RelFinder is a web application that can be used to find relation of interest between two selected entities [21]. An interesting approach to KG Exploration in the context of the Linked Data is Refer [2][7], which is an application that can be integrated in blogs as Wordpress Plugin and that scans an article to find entities and other meaningful text elements. After this operation, the Yovisto Semantic Framework is used to analyze the text to find those elements that are meaningful w.r.t the context. Next, external information, coming from other knowledge bases (like Wikipedia) is used to enrich the content with additional information. Refer is an example of contextual exploration of KG; the main difference between their approach and our approach is that we include semantic associations of length greater than one and introduce a model to order all associations, introducing a machine learning model personalize the exploration. None of the approaches mentioned above or surveyed in [1] introduces methods to learn information to show to the users based on their explicit feedback.

Learning to Rank and Active Learning for KG Exploration Learning to rank, which has been extensively applied to document retrieval [22], has been proposed only in one approach to KG exploration [5], where a function to rank semantic associations is proposed like in our approach. This approach use a variant of SVM on associations extracted from Freebase. Since the approach does not try to minimize the inputs needed to learn the ranking function, it can be hardly used to personalize the exploration of KG as our approach. In addition, some of their feature are specifically tailored on the Freebase structure while we propose features that can be easily applied to any KG (with a possible exception being time relevance, which requires bridges from the KG to Wikipedia). Active learning to rank introduces techniques to select the observations for which

[7] http://refer.cx/

labels are more informative. In our approach, we have implemented and tested two different techniques proposed for document retrieval. A first approach collects labels over individual observations (associations in our case) and solves the cold-start problem by randomly selecting positive and negative instances from a training set (which is split from the data ranked for the user). In our approach, we pick the associations that are labeled by a user from the set of associations ranked, which is more coherent with the contextual KG exploration workflow. However, we have also conducted tests with data split in a training and a test set to show the robustness of the model. In addition, we provided a principled approach to solve the cold-start problem in our domain. The second approach [15], which collects labels over pairs of observations, seem to be not only less efficient, but also less effective for ranking semantic associations. To the best of our knowledge, this is the first attempt to apply active learning to rank to the problem of exploring semantic associations. One approach that has proposed the application of active learning in the context for KG exploration, has been applied to a classification problem, i.e., to decide which nodes should be included in a graph summary [23], which is very different from the learning to rank problem discussed in this paper.

6 Conclusion

Experimental results presented in this paper suggest that active learning approaches can be effectively used to optimize the ranking of semantic associations extracted from KGs, thus supporting personalized exploration of complex relational knowledge made available in these graphs. We have also found that, in this context and for the selected set of features, AUC-based sampling performs better than pairwise sampling, both in terms of effectiveness and efficiency. As a result, an approach the combines a serendipity measure and AUC-based sampling, outperforms different alternative configurations. In future work, we plan to analyze the impact of individual features on the performance of an active learning to rank model for semantic associations, and evaluate the use of additional measures. In addition, we want to incorporate our active learning to rank model into the DaCENA application, by tackling the challenge of designing human-data interaction patterns that can engage the users.

References

1. Nikos Bikakis and Timos Sellis. Exploration and visualization in the web of big linked data: A survey of the state of the art. *arXiv preprint arXiv:1601.08059*, 2016.
2. Tabea Tietz, Joscha Jger, Jrg Waitelonis, and Harald Sack. Semantic annotation and information visualization for blogposts with refer. In *VOILA '16*, volume 1704, pages 28 – 40. V. Ivanova, P. Lambrix, S. Lohmann, C. Pesquita, 2016.
3. Giuseppe Pirrò. Explaining and suggesting relatedness in knowledge graphs. In *ISWC*, pages 622–639. Springer, 2015.

4. Matteo Palmonari, Giorgio Uboldi, Marco Cremaschi, Daniele Ciminieri, and Federico Bianchi. Dacena: Serendipitous news reading with data contexts. In *ESWC*, pages 133–137. Springer, 2015.
5. Na Chen and Viktor K Prasanna. Learning to rank complex semantic relationships. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(4):1–19, 2012.
6. Marieke Van Erp, Giuseppe Rizzo, and Raphaël Troncy. Learning with the web: Spotting named entities on the intersection of nerd and machine learning. In *# MSM*, pages 27–30, 2013.
7. Jonathan Gray, Lucy Chambers, and Liliana Bounegru. *The data journalism handbook*. " O'Reilly Media, Inc.", 2012.
8. Jaeho Kang, Kwang Ryel Ryu, and Hyuk-Chul Kwon. Using cluster-based sampling to select initial training set for active learning in text classification. In *Advances in knowledge discovery and data mining*, pages 384–388. Springer, 2004.
9. Weiwei Yuan, Yongkoo Han, Donghai Guan, Sungyoung Lee, and Young-Koo Lee. Initial training data selection for active learning. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, page 5. ACM, 2011.
10. Samuel J Gershman and David M Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.
11. Pang-Ning Tan et al. *Introduction to data mining*. Pearson Education India, 2006.
12. LI Hang. A short introduction to learning to rank. *IEICE TRANSACTIONS on Information and Systems*, 94(10):1854–1862, 2011.
13. Ching-Pei Lee and Chih-Jen Lin. Large-scale linear ranksvm. *Neural computation*, 26(4):781–817, 2014.
14. Pinar Donmez and Jaime G. Carbonell. *ECIR 2009. Proceedings*, chapter Active Sampling for Rank Learning via Optimizing the Area under the ROC Curve, pages 78–89. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
15. Buyue Qian, Hongfei Li, Jun Wang, Xiang Wang, and Ian Davidson. Active learning to rank using pairwise supervision. In *SIAM Int. Conf. Data Mining*, pages 297–305. SIAM, 2013.
16. Andreas Thalhammer and Achim Rettinger. PageRank on Wikipedia: Towards General Importance Scores for Entities. In *ESWC 2016, Revised Selected Papers*, pages 227–240. Springer International Publishing, Cham, October 2016.
17. Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.
18. Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.
19. Federico Cabitza and Angela Locoro. Questionnaires in the design and evaluation of community-oriented technologies. *International Journal of Web-Based Communities (to appear)*, 13(1), 2017.
20. Kilem L Gwet. *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics, LLC, 2014.
21. Philipp Heim, Sebastian Hellmann, Jens Lehmann, Steffen Lohmann, and Timo Stegemann. Relfinder: Revealing relationships in rdf knowledge bases. In *International Conference on Semantic and Digital Media Technologies*, pages 182–187. Springer, 2009.
22. Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.
23. Meng Fang, Jie Yin, and Xingquan Zhu. Active exploration for large graphs. *Data Mining and Knowledge Discovery*, 30(3):511–549, 2016.