Combining Serendipity and Active Learning for Personalized Exploration of Knowledge Graphs

Federico Bianchi [a], Matteo Palmonari [a] Marco Cremaschi [a] and Elisabetta Fersini [a]
[a] *University of Milan - Bicocca, Viale Sarca 336, Milan, Italy*
federico.bianchi@disco.unimib.it,
palmonari@disco.unimib.it,
cremaschi@disco.unimib.it,
fersiniel@disco.unimib.it

Abstract. Knowledge Graphs (KG) are now a widely used knowledge representation method and contain a large number of Semantic Associations (SAs), i.e., chains of relations that may reveal interesting and unknown connections between entities. Information that comes from a KG can be used to help a user that is doing a familiar task like reading an online news article, by adding contextual information that can provide informative background or serendipitous new details. Because of the large number of SAs that can be extracted from the entities that are found in an article, it is difficult to provide to the user the information that she needs. Moreover, different users might want to explore different SAs and thus exploration should be personalized. In this paper, we propose a method based on the combination between a heuristic measure, namely serendipity, and an active learning to rank algorithm that is used to learn a personalized ranking function for each user; this method asks the user to iteratively score small samples of SAs to learn the ranking function while reducing the effort on the user side. We conducted user studies in which users rate SAs while reading an online news article and used this data to run an experimental evaluation. We provide evidence that users are interested in different kind of SAs, proving that personalization in this context is needed. Moreover, results not only show that our methodology provides an effective way to learn a personalized ranking function but also that this contextual exploration setting can help users learn new things.

1. Introduction

Knowledge Graphs (KGs) describe real-world entities and their properties. Some of these properties represent links to other entities, providing a rich source of relational information. Languages recommended by W3C like RDF[1] can be used to publish KGs as (open) linked data, but KGs are frequently used also in the industry. KGs support a large variety of applications, including those targeted to knowledge exploration.

Differently, from query answering approaches, designed to return information relevant to specific information needs explicitly expressed by the users, knowledge exploration approaches are designed to deliver information interesting for the users in a more proactive fashion [1].

Often knowledge exploration approaches use contextual information to provide users with interesting details. We use *contextual Knowledge Graph (KG) exploration* to refer to a KG exploration setting in which a user who is carrying out a familiar task, e.g., querying a search engine, watching media content [2], reading a text of interest [3,4],

[1]https://www.w3.org/RDF/

explores content extracted from a KG, that is selected and proactively pushed to her.

As a representation of the current user interest, textual information can be used as an entry point to retrieve and select content from the KG that matches the user interests and capture his attention. To connect the raw text with a knowledge graph we can use Named Entity Linking (NEL) techniques [5]; these algorithms allow us to find mentions of KG entities within the text. After this phase, information about these entities, selected based on its estimated relevance, is shown to the users. Google infoboxes are one example of this idea: when a user makes a web search, he will see documents that match her query and additional information retrieved from Google's KG [3].

1.1. Contextual Exploration of KG with Semantic Associations

Imagine a European user that is browsing an online news site during the 2020 United States election. She might not be informed about the main actors involved in the topics of the articles. Additional information is thus required to help her in the exploration of data. She might ask herself "Who is Bernie Sanders?". Both Wikipedia and its structured version, DBpedia, provide a lot of information about *Bernie Sanders*[2] but we can not expect the user to read a complete Wikipedia page to get a small piece of information or to analyze data in structured format inside DBpedia.

In our previous work [4], we designed a data journalism application that goes beyond showing plain properties, DaCENA (Data Context for News Articles)[3]. In DaCENA, we let users who are reading a text explore Semantic Associations (SAs) extracted from the KG. SAs are loop-free semi-walks of finite length that connect two entities in the KG [6,4]. SAs between entities extracted from the input text reveal complex connections between entities, which provide new and interesting insights into the topic of the input text.

DaCENA presents to a user a set of SAs as a data context for the article that she is reading. The reference KG for DaCENA is DBpedia[4]. Users can read the news article and explore the SAs using an interactive interface.

Figure 1 shows a screenshot of the interface, with a news article extracted from the NYT [5] (curious readers can play with the example shown in the Figure 1 online [6]). The graph shows the k-most interesting SAs, where k can be set by the user. When the user clicks on an entity node, e.g., *Separatism*, SAs from/to such nodes are shown in the lower panel and ordered by interest estimated with the use of an heuristic function.

DaCENA data extraction phase has been summarized in Figure 2. Starting from an input text, a semantic annotator is used to extract entities from text. These entities are then used to find SAs in a KG. For further details about DaCENA processing steps, we refer to [4].

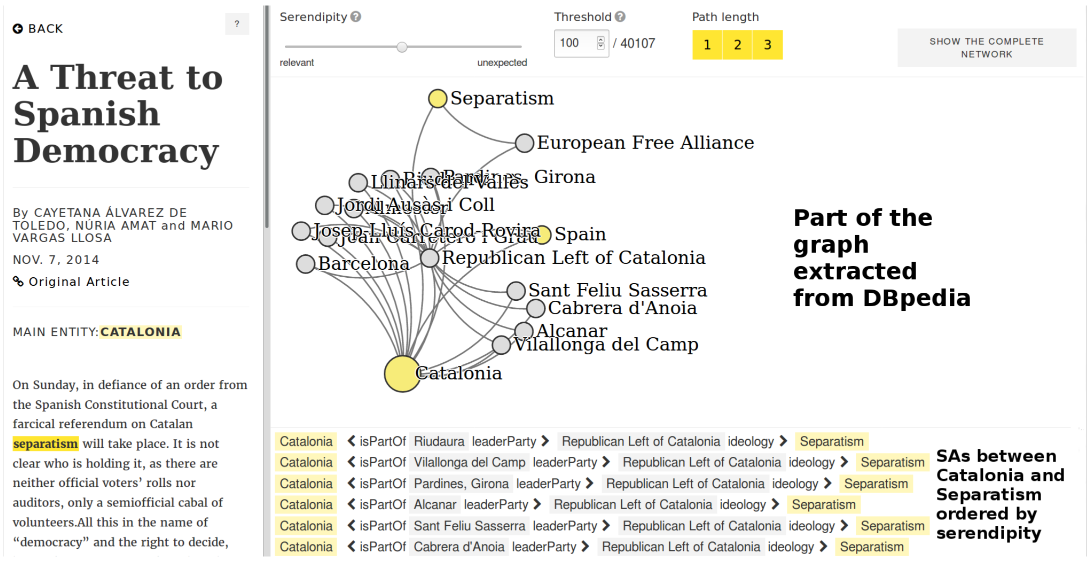
1.2. Challenges for Personalized Contextual Exploration of KG with SAs

Deciding which contextual information is valuable to be shown to users is challenging for every contextual KG exploration approach, but the problem is even more compelling when SAs are included in the information that is delivered to users: the number of SAs that can be found between entities extracted from even relatively short texts tends to be high.

For example, for the article used in the exploration use case depicted in Figure 1, we extracted 40.107 SAs from DBpedia. Otherwise, some preliminary user studies we have conducted to evaluate the usability of our DaCENA application suggest that users do not want to look at more than 100 SAs.

Moreover, supporting a user in the exploration of SAs is a key problem that we tackle by finding those SAs that are of interests for her between all the retrieved ones. In other words, the crucial problem to solve to support SAs exploration is to find an effective ranking function. Several approaches have been proposed that use context-less measures based on graph analytics to rank and filter SAs [6,7]. Others make use of machine learning methods to learn which associations are more interesting based on labels provided by a group of annotators [8]. In our previous work, we defined

[2]https://wikipedia.org/wiki/Bernie_Sanders - http://dbpedia.org/resource/Bernie_Sanders
[3]http://dacena.disco.unimib.it
[4]http://wiki.dbpedia.org/
[5]https://goo.gl/RFvqZh
[6]http://dacena.disco.unimib.it/article/84


BACK

?

A Threat to Spanish Democracy

By CAYETANA ÁLVAREZ DE TOLEDO, NÚRIA AMAT and MARIO VARGAS LLOSA

NOV. 7, 2014

Original Article

MAIN ENTITY: CATALONIA

On Sunday, in defiance of an order from the Spanish Constitutional Court, a farcical referendum on Catalan separatism will take place. It is not clear who is holding it, as there are neither official voters' rolls nor auditors, only a semiofficial cabal of volunteers.All this in the name of "democracy" and the right to decide,

Serendipity: relevant — unexpected

Threshold: 100 / 40107

Path length: 1 2 3

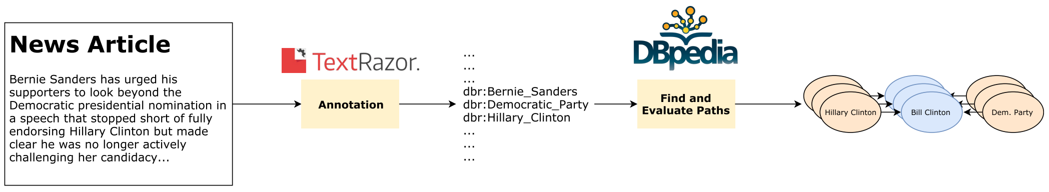
SHOW THE COMPLETE NETWORK

Part of the graph extracted from DBpedia

Catalonia	❮ isPartOf	Riudaura	leaderParty ❯	Republican Left of Catalonia	ideology ❯	Separatism	
Catalonia	❮ isPartOf	Vilallonga del Camp	leaderParty ❯	Republican Left of Catalonia	ideology ❯	Separatism	
Catalonia	❮ isPartOf	Pardines, Girona	leaderParty ❯	Republican Left of Catalonia	ideology ❯	Separatism	
Catalonia	❮ isPartOf	Alcanar	leaderParty ❯	Republican Left of Catalonia	ideology ❯	Separatism	
Catalonia	❮ isPartOf	Sant Feliu Sasserra	leaderParty ❯	Republican Left of Catalonia	ideology ❯	Separatism	
Catalonia	❮ isPartOf	Cabrera d'Anoia	leaderParty ❯	Republican Left of Catalonia	ideology ❯	Separatism	

SAs between Catalonia and Separatism ordered by serendipity

Fig. 1. An example of Contextual Exploration of KG with Semantic Associations from the DaCENA interface.



News Article
Bernie Sanders has urged his supporters to look beyond the Democratic presidential nomination in a speech that stopped short of fully endorsing Hillary Clinton but made clear he was no longer actively challenging her candidacy...
TextRazor.
Annotation
...
dbr:Bernie_Sanders
dbr:Democratic_Party
dbr:Hillary_Clinton
...
DBpedia
Find and Evaluate Paths
Hillary Clinton
Bill Clinton
Dem. Party

Fig. 2. Workflow of the DaCENA application.

a context-aware measure that considers the relevance of the SAs concerning the input text [4]. However, none of the above-mentioned ranking approaches places the individual user in the loop, supporting the personalized exploration of SAs by adapting the ranking function to her preferences. These approaches are thus inadequate if *different users are interested in different kinds of SAs*. We refer to the latter as to the "personalization hypothesis" for KG exploration with SAs.

Consider again the exploration use case depicted in Figure 1: while some users may be interested in finding out information about small municipalities associated with the topic of separatism, there are users that might be more interested in the exploration of information about more important cities.

1.3. Contribution of this Paper

In this paper, we propose an approach to personalize the contextual exploration of large KGs with SAs. With our approach a user who is reading a text can explore a set of SAs heuristically ranked by Serendipity [4], i.e., estimated to be relevant to the input text as well as unexpected (and thus interesting) for her. Then she can iteratively refine the ranking by rating few SAs at the time, thus expressing her preferences. We use a learning to rank algorithm, i.e., RankSVM [9], to learn from the individual user ratings and an active sampling method [10] to select the SAs on which ratings are collected. RankSVM is an algorithm that uses feature vectors to learn a ranking function over samples. As feature vectors, we use a combination of state-of-the-art measures that consider the graph topology, the frequency of co-occurrence of relations in SAs, relevance with respect to the text, and the temporal relevance of the entities occurring in the SAs. In order to learn better from a limited number of rates, we define an Active Learning to Rank (ALR) approach that supports the iterative improvement of the ranking quality, measured as adherence to the individual user preferences, while minimizing the user effort. On the one hand, our active learning method overcomes a limitation of a previous approach [8], which does not attempt to minimize the number of ratings collected from the users used to optimize the ranking function. On the other hand, by using *Serendipity* as heuristic function, we solve the cold-start problem that characterizes active sampling methods for ALR, i.e., the selection of informative samples requires an initial ranking, and at the same time, we can show to the user reasonably interesting SAs even before collecting any ratings.

Experiments conducted with two different datasets show that our approach is capable of improving the quality of the ranking over time using a limited amount of user ratings. We compare our approach with different baselines and alternative configurations for ALR. These configurations use different bootstrapping methods to overcome the cold start problem using different sampling methods. We show that our approach performs significantly better in terms of ranking quality improvement than all these alternative approaches, despite being more efficient and supporting a full pay-as-you-go, and thus more appealing, interaction model. Finally, the datasets constructed to evaluate our method provide insightful evidence for the personalization hypothesis, supporting the main motivation behind our approach. To the best of our knowledge, our approach provides the first application of active sampling to learning to rank for SAs and the first full pay-as-you-go approach to contextual exploration of KGs with SAs.

To summarize and better outline our contributions we define four research questions, and we describe the outcome of each one. In the experimental section we will explore in details these results:

– Q1) does contextual exploration require personalization? are different users interested in different kinds of SAs? **Outcome:** results on user scores gathered with the use of questionnaires suggest that different users are interested in different SAs.
– Q2) can we define an active learning to rank framework to personalize exploration SAs extracted from an article? **Outcome:** experiments show that the defined model can learn a personalized ranking function from a small number of feedback.
– Q3) which features are more useful to personalize the rankings? **Outcome:** features used in the model seem to capture different aspects of the SAs and are able to improve performance.
– Q4) is contextual knowledge, in the form of SAs extracted from an article, helpful for understanding articles? **Outcome:** user questionnaires suggest that SAs can help users to better understand what is written in an article.

This paper presents an extended version of a paper accepted for publication at ESWC 2017 [11]. While in the latter we explored the performance of different approaches, in this paper we describe in more details our approach based on the combination of Serendipity and Active Learning, we strength the confidence of our approach by adding up four kinds of analyses: (i) proving the significance of the results of our experiments, (ii) providing further insights about our findings, (iii) testing the effectiveness of our features and (iv) analyzing questionnaires results to provide further evidences on the usefulness of our application for Contextual KG Exploration.

The article is structured as follows: Section 2 contains the details of our approach on active learning to rank on SAs. Section 3 contains the experiments that we use to answer our research questions. In Section 4 we explore the recent advancement in the state of the art and eventually in Section 5 we discuss some conclusions related to our work.

2. Active Learning To Rank and Serendipity for Semantic Associations

In this Section, we explain in details our ALR model. We will first describe the general idea behind our model, and we will then explain each step in detail.

2.1. ALR Loop and Algorithms

We want to learn a ranking function for each user, and we use the RankSVM [9,12] algorithm to do this. RankSVM has been chosen for because it is considered a state of the art algorithm and has been widely used in the learning to rank domain [13,14], in fact, RankSVM is generally used in information retrieval settings. This force us to solve three problems: (i) first of all we need to define a set of features to train our algorithm, (ii) then

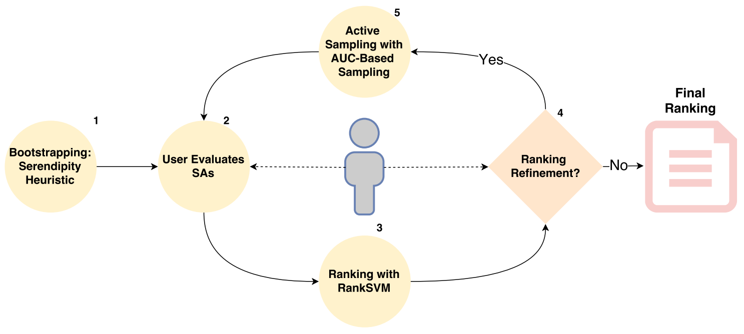


Fig. 3. Learning loop for the ALR model

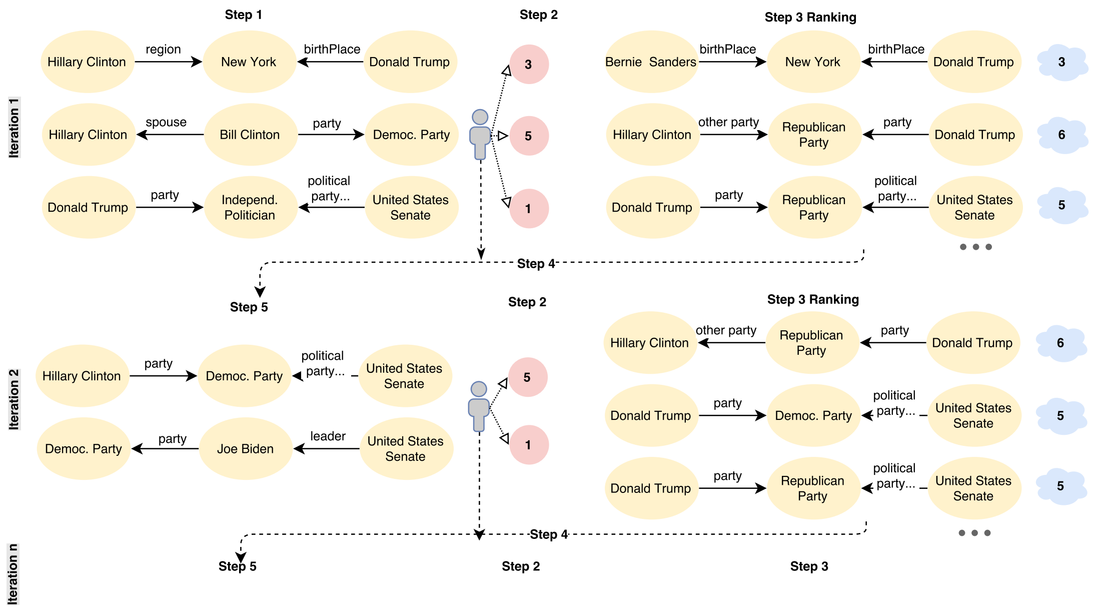


Fig. 4. Example of iterative ranking refinement with the ALR model

since we cannot expect a user to provide a large number of ratings that would be required to effectively train the RankSVM algorithm we need to find a way to reduce user effort (**user-effort problem**); (iii) finally we need to find a way to initialise the RankSVM model (**cold-start problem**).

To solve the **user-effort problem**, we decide to use active learning techniques [15], which have been proposed to reduce the number of observations needed to train a classifier. In our case, active learning is used to reduce the number of ratings required to train the learning to rank algorithm.

To solve the **cold-start problem** we ask users to rate some SAs when they are using the application. We use *Serendipity*, an heuristic ranking function [4] to select the SAs that users have to evaluate. The advantage of using a heuristic function is that we can collect the ratings needed to initialize the RankSVM model on a set of SAs that are also (heuristically) estimated to be interesting for the user. Only a subset of SAs deemed to be more interesting for the user can be shown to her in the user interface. By using *Serendipity* to bootstrap RankSVM, we can let users rate SAs that are in the subset shown in the user interface. Unfortunately, being *Serendipity* a heuristic function not specifically introduced to select samples that are informative for training RankSVM, it is uncertain whether this method is able to sample also useful training data. In addition, as many others machine learning models, in order to initialize RankSVM we need at least two ratings with different scores (e.g., one positive and one negative) meaning that we may need to sample more than once in order to initialize RankSVM. Indeed, if a user only scores the SAs with a single value, the ranking algorithm cannot be initialized.

Alternative approaches proposed to solve the cold-start problem in literature [16] use clustering algorithms to identify representative items in the feature space and collect ratings of them. While representatives samples may be, in principle, more informative for training they might be, conversely, less interesting to evaluate for a user. Thus, we aim to ask the user to evaluate, on the beginning, information that is deemed to be interesting for him.

In the experiments of Section 3 we will show that not only Serendipity supports a more appealing application workflow, but it also leads to better performance in terms of training time.

The model proposed to personalize the exploration of KG is based on the learning loop described in Figure 3. At each iteration, user ratings collected on small samples of SAs are used to train RankSVM and to update the ranking of the whole set of SAs. To better illustrate the loop, an example with two iterations, based on an actual run of our model, is depicted in Figure 4: ratings in red circle represents rating given at each ALR step by a user, ratings blue clouds represent ideal rankings of the ranked SAs (i.e., actual ratings that are given by the user after having rated all the SAs.

In the following we describe in detail each step of the loop of our model:

Step 1 - Bootstrapping. In the bootstrapping phase, Semantic Association (SA) are ranked by Serendipity. The user can view this pre-ordered set of SAs and then decide, if she is not satisfied with the result, to start a personalisation phase. In this case, a small number of top-k SAs ranked by Serendipity are selected to be rated by the user. In our experiments, we use k=3 and k=5 on two different datasets (see Section 3 for more details about the choice of these parameters).

Step 2 - User Ratings. The user labels the SAs selected in the first step (thus the SAs obtained with the Serendipity heuristic) using ratings in a graded scale, e.g., $< 1, 2, 3, 4, 5, 6 >$. Higher grades represent higher interest for an SA.

Step 3 - Ranking. We use the ratings provided by the user to train RankSVM, which ranks all the remaining SAs by assigning them a score.

Step 4 - User Decision. The user can see the output of the ranking function, and if she is satisfied with the ranking obtained so far, the loop stops. Else, we further improve the ranking by letting the user trigger the selection of a new sample of SAs to rate.

Step 5 - Active Sampling. An active sampling algorithm proposed for document learning [10], referred to as AUC-Based Sampling in the paper, is used to find the observations (SAs) for which ratings are estimated to be more informative. The algorithm uses the scores determined by the learned ranking function, which motivates the reason for using a different algorithm for sampling the data used to bootstrap the model. After Step 5, we close the loop by repeating Step 2.

Observe that after the first iteration, the user always labels SAs selected with active sampling. In Figure 4, it can be noticed that the quality of the ranking improves after the second iteration (Iteration 2 - Step 3).

The main steps of the loop, i.e., bootstrapping, ranking and active sampling, are explained in more details here below.

2.2. Bootstrapping with Serendipity

Serendipity is defined as a parametric linear combination of two different measures: (i) **Relevance**, that is a measure that computes the relevance of a given SA with respect to a text, and (ii) **Rarity**, a measure that computes how much a given SA may be unexpected to the users. To define the unexpectedness of a SA we consider how rare its properties are in the KG.

Relevance of a SA π to an input text *text* is thus defined as follows:

$$relevance(text, \pi) = cos(v_{text}, v_\pi)$$

where v_{text} and v_π are word vectors representing the text and the SA respectively. The text from which we generate v_π is built as the concatenation of short texts describing the entities occurring in π in the KG (in particular, we used DBpedia abstracts[7]). Word vectors are weighted using TF-IDF, where for each input text (and the SA extracted from it) we build a dedicated vector

[7]http://dbpedia.org/property/abstract

space. An SA is unexpected when it is composed by properties that are not frequently used in the KG, which can be captured by a Rarity measure.

Rarity of a property p can be defined as follows:

$$rarity(p) = 1 - n\_frequency(p)$$

where, $n\_frequency(p)$ represents the frequency of p in the KG normalized into the range $[0, 1]$ using the max-min method. In other words rarity is defined as the inverse normalized frequency of a property. Rarity of a SA π is defined as the average rarity of the properties occurring in the SA.

Since **Relevance** and **Rarity** are normalized in the interval $[0, 1]$, they can be smoothly combined. Let $\alpha \in [0, 1]$ be a parameter used for balancing the weight of each measure, and *text* be the input text; the serendipity $S(\pi)$ of an SA π, is computed as:

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

We show an example of the top-2 ranked SAs and one of the bottom-2 ranked SAs, computed using Serendipity with $\alpha = 0.5$ in Figure 5. The text[8] from which those SAs were extracted from is shown on the left-hand side of the figure. The top-ranked SAs contain the entity *Hillary Clinton*, which is relevant to the article text. The order of the SAs showed is consistent with the definition of Serendipity: the predicate *birthPlace* is used frequently in DBpedia (and thus probably uninteresting) and *George W Cate* and *Joseph K Allen* are not mentioned in the article text. The *birthPlace* predicate is present also in the top-2 ranked SAs, but higher relevance of the entities occurring in these SAs to the article text lead to higher Serendipity scores.

2.3. Ranking with RankSVM

We choose to use Rank SVM to learn the ranking function based on user ratings. Rank SVM has become one of the state of the art algorithm for learning to rank documents [9,12]. Since RankSVM is an adaptation of Support Vector Machines to solve ranking problems it represents the items to rank as feature vectors. It is based on a pairwise ranking algorithm. This means that the input to the model is a set of pairs that contains two observations with their relative order. As an example, if in the training set there are two feature vectors x_i and x_j, the input to the RankSVM would be $\{(x_i, x_j), y\}$, where y is a label that indicates the relative order between x_i and x_j. The label is $y = +1$ if x_i should be ranked higher then x_j (vice-versa when $y = -1$). Thus, Labels can be derived from ratings attributed to individual items: from pairs $\{(x_i, x_j)\}$ if the rating for i is higher than the rating for j a training example $\{(x_i, x_j), 1\}$ is defined. Also in our approach labels are collected via ratings. RankSVM builds such pairs based on a set of ratings provided on an arbitrary set of observations. At iteration t the model is retrained by considering all the labels generated from the ratings collected until iteration t.

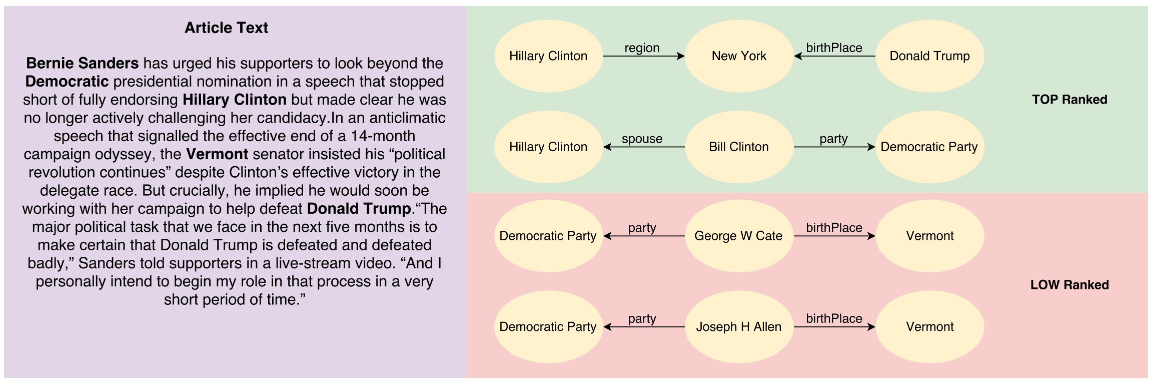
2.4. Active Sampling

Our model uses AUC-based Sampling as the active sampling algorithm [10]. This method optimizes the Area Under the ROC Curve and accepts ratings on an arbitrary set of observations. It select samples that are expected to minimize the rank loss and does not consider the native pairwise structure of the problem: this makes the algorithm sub-optimal for this task, but fast to compute. Other methods that consider the pairwise structure are more expensive to compute. An example of the SAs selected by this active learning to rank algorithm can be seen in Step 5 of Figure 4. However, we want to emphasize, that our workflow is independent on the choice from the active learning algorithm.

2.5. Features

According to the data model used in RankSVM we selected different features to represent SAs with the idea of capturing relevant information of each SA. Since the measures used in this context return values in different ranges we decide to normalize values by scaling them to have zero mean and unit variance [17]. To better capture the information described by each SA we used three different kinds of features: (i) topological features, (ii) predicate-based features and (iii) relevance features.

[8]https://www.theguardian.com/us-news/2016/jun/16/bernie-sanders-will-work-with-clinton-donald-trump-speech



Article Text

Bernie Sanders has urged his supporters to look beyond the **Democratic** presidential nomination in a speech that stopped short of fully endorsing **Hillary Clinton** but made clear he was no longer actively challenging her candidacy.In an anticlimatic speech that signalled the effective end of a 14-month campaign odyssey, the **Vermont** senator insisted his "political revolution continues" despite Clinton's effective victory in the delegate race. But crucially, he implied he would soon be working with her campaign to help defeat **Donald Trump**."The major political task that we face in the next five months is to make certain that Donald Trump is defeated and defeated badly," Sanders told supporters in a live-stream video. "And I personally intend to begin my role in that process in a very short period of time."

Fig. 5. Example of top-2 and bottom-2 SAs ordered by Serendipity.

2.5.1. Topological Features

Here we present the features that are based on the topological structure of the graphs used to represent the data. These features are mostly based on centrality measures that evaluates the importance of the entities in the SAs and can be computed considering two different graphs: the first one is the KG from which we extract the SAs, which will be referred to as *global graph*; the second one is the subgraph of the KG that consists of the SAs extracted from the input text, which will be referred to as the *local graph.* Using these two graphs we can compute *global* and *local* scores of centrality for the entities. An entity with a high centrality score can be considered important due to the high number of links they have (and thus, more or less interesting for a user depending on her preferences). Once we have computed centrality score for all the entities in a SA we average their score and use the average as a feature for the SA.

Global PageRank. We use the data in [18] to collect a global PageRank score inside DBpedia. In this, way we are able to get an overall estimation of the importance of an entity inside the KG. With this feature, entities that are central in DBpedia (like United States of America[9] and Barack Obama[10]) will be considered important even if they have a limited number of links in the subgraph extracted from the text.

Local PageRank. We use the PageRank *centrality* algorithm on the local graph defined by the SAs extracted from the input text.

Local HITS. We ran another centrality algorithm, HITS (Hyperlink-Induced Topic Search)[19], to collect two scores for each node of the local graph. The authority score indicates how much a node is *important*, while hub score indicates nodes that point to nodes with a high authority score. The algorithm gives two scores in the feature vectors describing a SA: one for the average of the authority values and one for the average of the hub values.

2.5.2. Predicate-Based Features

In this Section, we present features that consider predicates occurring in the SAs. We decided to use features that are focused on finding the most important/interesting predicates inside the SAs.

Path Informativeness. It is a measure defined in [6], based on the concept of Predicate Frequency Inverse Triple Frequency (PF-ITF). This measure tries to identify discriminative paths in a way that is similar to the ones commonly defined for text like TF-IDF (Term Frequency - Inverse Document Frequency).

Path Pattern Informativeness. It is a measure based path patterns, defined in [6], to get the informativeness of patterns extracted from paths. The path pattern is used to generalize the paths in a dataset.

Rarity. This measure is the unexpectedness factor presented in the Serendipity Heuristic, more details can be found in Section 2. For example, predicate frequently used in DBpedia, like birthPlace, have low rarity score.

2.5.3. Relevance Features

This last section illustrates the features that have been defined to find those SAs that are more related to the context of the article that was analyzed.

[9]http://dbpedia.org/page/United_States
[10]http://dbpedia.org/page/Barack_Obama

Relevance. This measure is the relevance factor presented in the Serendipity Heuristic more details can be found in Section 2.
Temporal Relevance. Using the Wikimedia API we extract the number of times a Wikipedia entity (page) has been accessed in a specific date (e.g., the date of the publication of a given text). With this new value we are able to compute a value of importance that is temporally defined. Consider for example, access[11] for the page Paris in Wikipedia, we can see that the page has been accessed 8.331 times on 12-11-2015 and 171.988 times on 14-11-2015. This is because 13-11-2015 is the date of terrorist attacks in Paris. As in other measure, the final score used as a feature for the SA is computed as the average of the scores obtained by the entities of the SA.

3. Experiments

We run different experiments to verify the performance of the selected algorithms and to evaluate the effectiveness of our features answering the research questions outlined in the Introduction.

3.1. Datasets and Methodology

Two different datasets were built to test the performance of our approach. We collected these datasets ourselves because we were not able to find an open dataset that could be used to evaluate contextual exploration of SAs: in our approach, the context is given by the text, that is fundamental for our exploration scenario. Each dataset consists of triples $< text_i, A_i, Ratings_{u,i} >$, where $text_i$ is a piece of text that was extracted from an article retrieved from online news platforms like the New York Times[12] and The Guardian[13], A_i is the set of all SAs extracted from $text_i$ with the DaCENA application, and $Ratings_{u,i}$ contains the ratings assigned by a user u to every SA that is present in A_i. Each triple in a dataset represents a complete ranking of a retrieved SAs for one user, i.e., a personal ideal ranking. We describe the creation of each dataset here below.

[11]http://tools.wmflabs.org/pageviews/
[12]www.nytimes.com
[13]https://www.theguardian.com

Short Articles Many Users (SAMU). We collected user ratings for this data set using an online web application that we implemented for this task. The users were asked to read an article and to specify how much they were aware of the topics of the article. Then, the user was asked to give a rating to all the SAs. The questionnaire asked the user to give a new rank about how much the topic and the elements of the article were clear after they have been able to read each SAs (using the same scale between 1 to 6 as explained before); this allows us to understand if the contextual exploration settings is useful for users.

For ratings, we choose a graded scale from 1-to-6 following guidelines suggested in a recent study [20]. Five-valued ordinal scale are not symmetric while the 1-to-6 scale provides a symmetric range that can be in principle used to divide scores in two sets: scores with a negative tendency (1, 2 and 3) and scores with a positive tendency (4, 5 and 6). Moreover, users cannot give a neutral score and have to decide if they like the SA or not. Each user had to evaluate all the SAs extracted from a text. For this reason, we choose texts small enough to let users perform the task without being subject to fatigue bias [20]. We extracted the first self-contained paragraphs of news articles from the New York Times and The Guardian with the following features: the article's topic concerns politics and is reasonably well-known and engaging for foreign (Italian) educated users; the number of SAs extracted by the DaCENA application is between 50 and 100. The time required to complete the task was on average of 12 minutes - a value that is little below the fatigue bias threshold mentioned in a recent study on the topic [20].

We stopped gathering users for the evaluation when we collected evaluations by at least 3 users on each article (for a total number of 5 articles), which resulted in a total of 14 different users, and 25 gold standards (personal rankings). The average number of SAs for each article present in the dataset is equal to 73.2. Graded scale questions on article understanding and knowledge about the topic were given using cardinal values: Not Knowledge, Poor Knowledge, Slightly Confident, Confident, Very Confident, Expert

Long Articles Few Users (LAFU). We wanted also to evaluate if results obtained over small SA sets are comparable with results obtained with (and thus generalizable to) large SA sets. To this end, two users

In this case, we used a three-valued scale for ratings, from 1 to 3. The two users involved in the evaluation of the longer articles were Communication Sciences students with no background in Computer Science. They were granted several days for completing the task and asked to complement their task with a qualitative analysis. At the end of this evaluation, we had three datasets (coming from two articles) on which experiment on, with an average number of SAs equal to 2656 SAs.

3.1.1. Measures for the Evaluation

Ranking Quality: using the ideal rankings in the two datasets, we decided to measure the quality of the rankings returned by our model at different iterations using Normalized Discounted Cumulative Gain (nDCG) computed over the top-10 ranked SAs, denoted by nDCG@10. nDCG@10 was used to give more importance to the first retrieved SAs. In addition, we wanted to have an aggregate performance measure, and thus we computed the Area Under the nDCG@10 Curve (AUNC), the curve is based on the nDCG@10 values at each iteration.

Similarity between Rankings: we also use Kendall's τ to evaluate the similarity between two different rankings. The definition is based on the concept of concordant pairs: two rankings are concordant on the ranking of a pair (x_i, x_j) if they both agree on the relative order of the elements, thus if x_i ¿ x_j or x_j ¿ x_i for each of the two. The formula to compute Kendall's τ appears below. Kendall's τ has already been used to evaluate rank orderings in learning to rank tasks [21].

$$\tau = \frac{ConcordantPairs - DiscordantParis}{ConcordantPairs + DiscordantPairs}$$

The values of this measure range from 0 to 1, where 1 means that ordering are the same, while 0 means that there is no relationship between the ratings.

3.2. Q1: Users' interests and personalization

One of our assumptions was that the personalization was needed because different users are interested in different SAs. We measured Inter-Rater Reliability [22] (IRR) among users who provided ratings in the SAMU dataset, to prove that a personalized exploration is not only important but needed. IRR is used to compute the rate of agreement between the users, in our case a low rate of the agreement would indicate a disagreement of the interest of different users with respect to the same SAs. We used two measures for evaluating the IRR of the dataset: Krippendorff's alpha (weighted using an ordinal matrix, since our context uses ordinal values), which output was 0.062, and Kendall's W, from which we obtained a score of 0.2608. The Table 1 shows these results. IRR is low and distant from 1, the value that usually represents unanimity between the raters. We show, in Table 4, the distribution of the ordinal ratings for both datasets. We remark again that the SAs in the SAMU dataset have a rating scale that ranges from 1 (low interest) to 6 (high interest) while the one from LAFU are from 1 (low interest) to 3 (high interest).

Krippendorff's alpha	0.062
Kendall's w	0.268

Table 1
Inter-Rater Reliability on the SAMU data set

3.3. Q2: Active Learning to Rank Performance

Settings of the experiments We carry out experiments in two different settings. 1) In *Contextual Exploration Settings*, we consider the workflow as implemented in a system that supports contextual exploration: the set from which we select the SAs to label is the same set that is then used to evaluate the performance of the model. In these settings, SAs labeled during previous iterations are not labeled a second time by the user. 2) In *Cross Validation Settings*, we use a different workflow for evaluation purposes: SAs are split in training set and test set similarity to what is done in active learning contexts [10]. We can use 2Fold-Stratified Cross Validation (CV) to make sure that results can be reasonably generalized and do not depend on specific data. 2Fold CV was used because the cardinality of the dataset was low. With 2Fold CV we were able to have enough data on which the active sampling algorithms were able to select the observations to rate and enough data to test the model.

3.3.1. Configurations and Baselines

Direct comparison with other state-of-the-art approaches is difficult because to the best of our knowledge we could not find an active learning to rank approach for SAs. Moreover, other approaches that deal with SAs are evaluated in a context-free settings, while in our contextual exploration scenario, input text needs to be considered. Thus, we compare our approach to several alternative approaches based on different methods applied in the three steps of the feedback loop: **Bootstrapping**, **Active Sampling** and **Ranking**. Table 2 shows a summary of the configurations. The algorithms that have been signed with the blue are the ones that use active sampling techniques to reduce the number of observations needed to learn the model, while the ones in red do not use active sampling. We will illustrate the details of the algorithms in the remaining part of this section.

The **Bootstrapping** phase of our model is currently guided by the serendipity heuristic (with $\alpha = 0.5$). We decided to compare it with the use of two clustering algorithms since clustering has been used in active learning settings [16,23]. The choice of clustering is motivated by the fact that an observation near to the cluster means can be considered representative of the cluster and thus it can be informative to learn preferences from that observation. We consider two clustering algorithms: the Gaussian Mixture Model and the Dirichlet Gaussian Mixture Model. Dirichlet Gaussian Mixture Model was chosen due to its ability to automatically find the number of clusters for the dataset. For the Gaussian Mixture Model, we use the silhouette coefficient [17] to detect the best number of clusters in each dataset. The first SAs to show to the user for evaluation are those nearer to the mean of each cluster found by the clustering algorithm, thus we selected one SAs for each cluster.

The **Active Sampling** phase of our model is done using AUC Based Sampling [10] (AS), we thus selected Pairwise Sampling [24] (PS) as an alternative to it. PS is based on the pairwise structure of RankSVM, and it tries to find those observations that could be more interesting to evaluate by estimating the uncertainty of the observations with respect to the ranking problem. PS combines two different kinds of uncertainty a Global Uncertainty and Local Uncertainty and uses a parameter p to regulate the importance of Local Uncertainty. Local uncertainty estimates how the ordering of a pair of instances is not clear to the function used in the ranking phase. While the global uncertainty estimates how uncertain is a single instance with respect to all the other instances. The general idea is to prevent the selection of outliers in the active sampling phase. More details about this algorithm can be found in the original work [24] The following parameters of these two algorithms have been determined experimentally in cross validation, we set $\lambda = 0.8$ for AS and $p = 1$ for PS.

Three baseline algorithms that do not use proper active learning were also tested:

- *Random + Random*: we use RankSVM to learn a ranking function, but the model is trained using ratings over randomly sampled SAs. Randomly selected SAs are thus used for the first step (initialization) and the active learning step. This random algorithm has been run multiple times to stabilize the results (100.000 times).
- *Serendipity No-AL*: we consider the ranking determined with Serendipity, which is not based on active learning and does not change across iterations. In this case, RankSVM is not used.
- *Random No-AL*: we consider random rankings of SAs, which are not based on active learning and do change across iterations. In this case, RankSVM is not used.

The last two algorithms are considered to understand if an incremental learning with active learning to rank can outperform an order given by a simple heuristic function and a random approach.
Configuration Details. To compare the approaches with the serendipity heuristic we define a few configurations in such a way that the training set for the various algorithms is of the same size and thus balanced; this is done to obtain a fair comparison between the algorithms. In the SAMU data sets Dirichlet and Gaussian Clustering, in the first iterations, detected an average number of clusters equal to 3 (and thus, an average number of 3 SAs are selected from these two methods in the first iteration); for this reason, on average, to feed the model with a balanced number of SAs we choose to select 3 SAs when using Serendipity and Random in the bootstrapping step. The active sampling step for this dataset extracts the top-2 ranked SAs as determined by the active sampling techniques used in the configuration. For the Random Ran-

	Algorithm	**Bootstrapping**	**Sampling**	**Learning**
🟦	Serendipity AS	Serendipity Heuristic	AUC-Based Sampling	RankSVM
🟦	Serendipity PS	Serendipity Heuristic	Pairwise Sampling	RankSVM
🟦	Dirichlet AS	Dirichlet Gaussian Mixture Model	AUC-Based Sampling	RankSVM
🟦	Dirichlet PS	Dirichlet Gaussian Mixture Model	Pairwise Sampling	RankSVM
🟦	Gaussian AS	Gaussian Mixture Model	AUC-Based Sampling	RankSVM
🟦	Gaussian PS	Gaussian Mixture Model	Pairwise Sampling	RankSVM
🟥	Random Random	Random	Random	RankSVM
🟥	Random - No AL	No Bootstrapping	No Active Sampling	No Learning to Rank
🟥	Serendipity - No AL	No Bootstrapping	No Active Sampling	No Learning to Rank

Table 2
Details of the algorithms configurations

dom approach we again select 2 random SAs. The number of observations collected at each iteration was increased in LAFU since this dataset is bigger. The clustering algorithms in this dataset detected an average number of cluster equal to 5, leading to 5 SAs to be labeled when using Serendipity and Random. In the active sampling, for the LAFU data set, we selected 6 observations to be labeled at each iteration for both AS and PS. Table 3 shows the number of SAs that each algorithm selects for each step. The *Temporal Relevance* could not be used in LAFU as a feature because articles in this dataset were not recent enough to be able to extract the page views. We used a RankSVM with polynomial kernel (degree equal to 2) on LAFU, since dataset were bigger and nonlinearity was more probable, that was able to output the results of a single iteration in what we considered interactive time (less than 2 seconds); on the SAMU dataset RankSVM was run with a linear kernel.

3.3.2. Results and Discussion

Contextual Exploration Settings The reader can view the results in Figures 6 and 7, where we plot the average nDCG@10 for the first five iterations of the model. On the average, a user at the 5th iteration has given ratings to 12 SAs for the SAMU dataset and 30 for the LAMU dataset. On the SAMU dataset, the best algorithm appears to be the Serendipity AS one. One interesting thing to notice is that active learning, performs better than methods that do not use an incremental learning method (Random No-AL and Serendipity No-AL). This is important because it proves that active sampling is useful for maximizing the ranking function. In the LAFU dataset Serendipity No-AL was able to get a good result and, if we do not consider the Serendipity AS configuration, active learning required three iterations to perform better then Serendipity No-AL. The Random Random approach, that uses a random selection method for active sampling, performs better with each iteration, but the overall performance is low, compared to the other algorithms that use active sampling. This is an indication of how much the initial training set and the SAs selected actively are important for ranking algorithms. The areas computed can be found in Table 6 and show that Serendipity AS is the algorithm with the biggest area.

Cross Validation Setting. Cross validation has been used to provide evidence that the module is robust and the main result obtained is that the model with the best performance is ours. The conclusions are similar to the one already explained in the section above, the general performance in this case is slightly less good. The plots can be found in Figures 8 and 7 while the computed areas are in Table 5.

Nevertheless, while AS [10] was originally designed and developed in binary learning to rank setting, it was able to obtain good results even in a multi-rating setting (our case).

Interactive Time. We designed an approach that should interact with users. One of the most important requirements from the user side is the interactive time. A user is willing to use an application only if the response he gets from the platform is given in a short period of time. We computed the average number of seconds the model needs to train the ranking algorithms and provide a ranking to the user; in this case we tested the Serendipity AS algorithm. For the SAMU dataset the model

	SAMU	LAFU
Step 1 #SAs		
Serendipity	Top-3 SAs with serendipity	Top-5 SAs with serendipity
Random	3 randomly extracted SAs	5 randomly extracted SAs
Clustering	3 SAs found on average	5 SAs found on average
Step 3 #SAs		
AS&PS	Top-2 SAs found with AL	Top-6 SAs found with AL
Random	2 randomly extracted SAs	6 randomly extracted SAs

Table 3
Number of observations selected at each steps by the algorithms

Rating	**1**	**2**	**3**	**4**	**5**	**6**
SAMU	23.7%	14.5%	22.3%	20.9%	10.1%	5.5%
Rating	**1**		**2**		**3**	
LAFU	67.4%		30.1%		2.5%	

Table 4
Rating distribution in the two data sets

Configurations	**SAMU**	**LAFU**
Serendipity AS	**3.0742**	**2.711**
Serendipity PS	3.0302	NaN
Gaussian AS	3.0168	2.6455
Dirichlet PS	3.0009	NaN
Dirichlet AS	3.0011	2.6872
Gaussian PS	2.9975	NaN
Random Random	2.976	2.6013

Table 5
AUNC in Cross Validation

Configurations	**SAMU**	**LAFU**
Serendipity AS	**3.2018**	**3.0817**
Serendipity PS	3.1399	NaN
Gaussian AS	3.0242	2.747
Gaussian PS	2.9629	NaN
Dirichlet AS	3.0711	2.7174
Dirichlet PS	3.019	NaN
Random Random	2.9359	2.673
Serendipity No-AL	2.7199	2.734
Random No-AL	2.3199	1.7971

Table 6
AUNC in Contextual Exploration

was able to compute the result in 0.35 seconds while for the LAFU dataset the average number of seconds required is around 2. This means that as the dataset gets bigger, the algorithm requires a longer time to compute the result.

Initial training set analysis. If a user in the first step evaluates all the SAs with the same ordinal degree, we can not start training the Rank SVM model. We thus evaluate the *time for the first iteration*: this value corresponds to the average number of iterations needed to have a training that can be used to train the learning to rank algorithm; we compute this for each bootstrap method. Table 7 shows the result of this experiment, where we show the results for both Cross Validation (CV) and Contextual Exploration Setting (CE). In LAFU data set it is more difficult for the algorithm to find the first observations that are needed to initialize the ALR model; the Dirichlet Clustering method can probably adapt itself to the size of the dataset in an easier way and has better performance in this task, a result that is consistent with how this clustering algorithm is defined. However, looking at the plots (Figures 6,7,8 and 9) we can see that while Dirichlet clustering is able to rapidly gain an initial training set for the RankSVM model (typically around 1 iteration), this training set makes the model obtain a performance that is not has good as the one obtained trough the use of serendipity.

Independence Test of nDCG@10 results To verify the results obtained by our model we performed the Wilcoxon test, on the various nDCG@10 obtained by the **last** iteration of the Cross-Validation, to prove that the final result of each algorithm is statistically significant with respect to the others. We generated a matrix with the algorithms configurations on the columns and the datasets on the row. An element $x_{i,j}$ of this matrix is the nDCG@10 of the last iteration of the Cross Validation for the i-th dataset obtained with the j-th algorithm. P-values obtained with the Wilcoxon

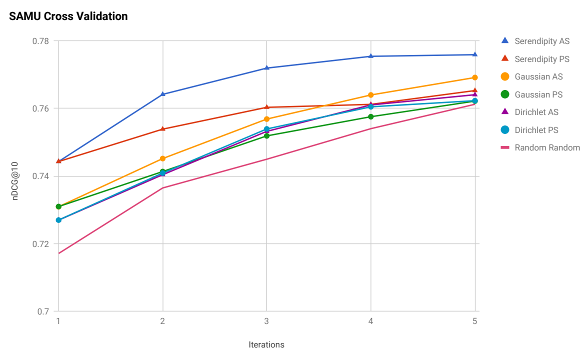


Fig. 6. nDCG@10 for SAMU Contextual Exploration

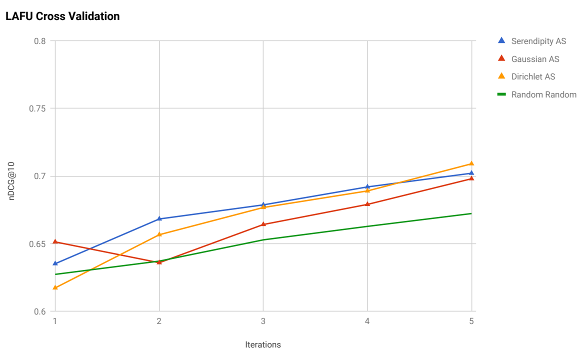


Fig. 7. nDCG@10 for LAFU Contextual Exploration

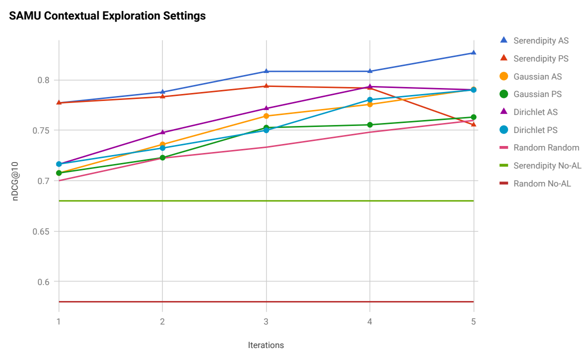


Fig. 8. nDCG@10 for SAMU Cross Validation

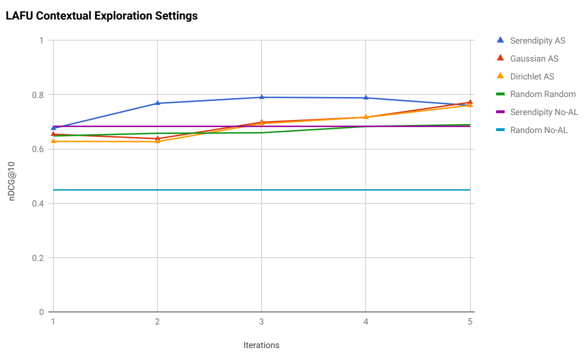


Fig. 9. nDCG@10 for LAFU Cross Validation

Algorithm	**Average #Iter. SAMU CV**	**Average #Iter. LAFU CV**	**Average #Iter. SAMU CE**	**Average #Iter. LAFU CE**
Serendipity	1.08	1.5	1.35	1.14
Dirichlet	1.16	1.11	1.01	1.06
Gaussian	1.08	1.16	1.01	1.38
Random	1.25	1.50	1.19	1.23

Table 7

Average Iterations to find the first training set for the ranking model

test (that was run with an $\alpha = 0.05$) can be seen in Table 8, we signed with a "*" the values that are statistically significant. We can conclude that using Active Sampling is significant with respect to PS and Random. When we use AS we can see that results with Serendipity, Dirichlet or Gaussian are correlated, this is because the only difference between Serendipity AS, Gaussian AS and Dirichlet AS lies in the first iteration. However, from an applicative point of view, Serendipity provides an approach that is smoother and cleaner for a user, because the SAs retrieved in the first iteration are interesting, while using a clustering algorithm could force the user to evaluate SAs that are not interesting. This has been proved by good performance in the resulting section. Results of the test suggest that not only results obtained with AS seem more significant of those obtained with PS, but even of those obtained with a Random approach.

3.4. Q3: Features Analysis

3.4.1. Correlation Analysis on Features

To understand the effectiveness of the features we introduced into the model we computed the correlation (using Pearson Correlation coefficient) between all of our features. The heatmap with

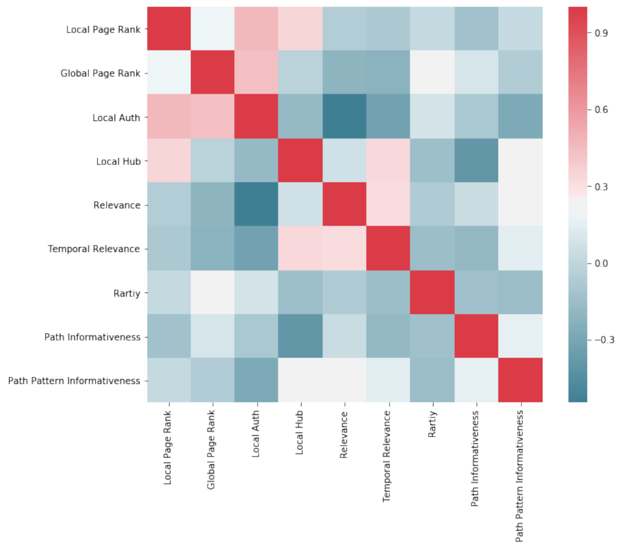
Table 8

p-values for the Wilcoxon signed-rank test on the SAMU dataset

	Dirichlet AS	Gaussian PS	Dirichlet PS	Serendipity AS	Serendipity PS	Random
Gaussian AS	0.07548	0.003781*	0.02365*	0.7915	0.173	0.0008081*
Dirichlet AS		0.09032*	0.5077	0.615	0.5782	0.05158*
Gaussian PS			0.1014	0.002255*	0.1073	0.731
Dirichlet PS				0.01247*	0.9578	0.4908
Serendipity AS					0.09573	0.02748*
Serendipity PS						0.5965

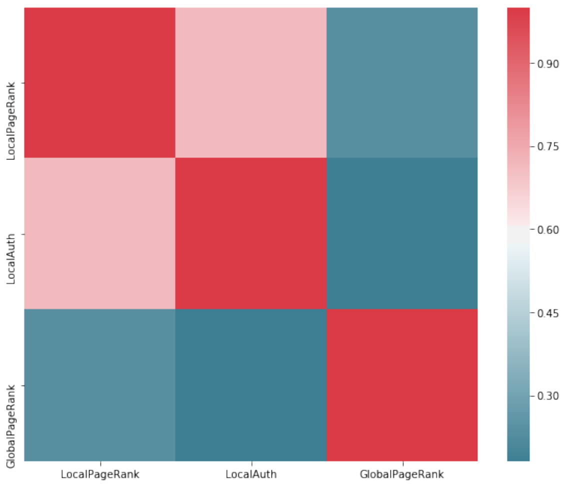
the results can be seen in Figure 10: heatmap squares' color range from blue, meaning uncorrelated/slightly negative correlation, to red, meaning positive correlation. Most of the measures seem to be lowly correlated/uncorrelated (values ranging between -0.3 and 0.2). As expected, measures that estimate the centrality of nodes seems to be slightly more correlated.

Fig. 10. Correlation heatmap of the feature used in ALR



To further investigate the relations between the centrality measures we run a test computing the Kendall's τ coefficient: for each article we collect the orderings of SAs with Local Page Rank, Local Auth and Global Page Rank, ranking them from the one with the highest value to the one with the lowest value. For each pair of algorithms we then computed the τ coefficient. A heatmap that summarizes the dependencies can be seen in Figure 11.The ordering computed with Global PageRank is different from the other two, this result can be explained by the fact that the Global Page Rank is computed on a different graph (that is the DBpedia KG). Rank ordering obtained by Local Page Rank and Local Auth are similar (average Kendall's $\tau = 0.713$). For some articles the τ has reached a value of 0.9. This seems in line with other results in the literature that report that PageRank and HITS might give similar ordering, where the main differences between the ranking sequence can be related to the fact that, during computation, HITS considers both inbound and outbound links in the computation, while PageRank focuses on inbound links [25]. Anyway since the average Kendall's τ is still far from 1, Local Auth can be still used as a feature. Also, we remark that our measures are computed on aggregated values: Local PageRank of an association is computed as the sum of the PageRank of each node in the association, this can also be a hint on why the Local PageRank and Local Auth ordering are not always the same.

Fig. 11. Kendal τ heatmap for the centrality measures



3.4.2. Features Sensitivity on Ranking

We also conducted an analysis of our features with respect to the ranking problem using the insights and the measures provided by the literature [21]. At the best of our knowledge there is no existent method to perform an in-depth analysis of the features used in an active learning to rank context. In this section we thus investigate the performance of every single measure by computing the importance of a feature and the similarity between features in the same way as [21]. While these two measure where defined to be part of a features selection algorithm, we use them to evaluate our own features.

– importance: we rank, using RankSVM, SAs using one feature at time and we compute the nDCG@10 of each feature. The value of the nDCG@10 is the importance score of one feature.
– similarity: we compute Kendall's τ against the ranking of each pair of features. Kendall's τ, as explained in the preceding section is used to evaluate the similarity between the ranking orders.

For both measures we run the experiments with the following settings: we run a 2-Fold Cross Validation on the data. The training set is used entirely to train the model, while we compute nDCG@10 on test data. We run experiments multiple times and average results.

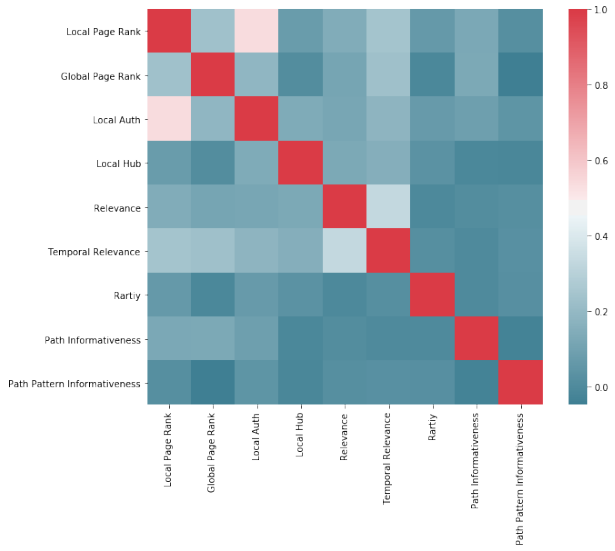
Importance For each measure we report mean nDCG@10 and the standard deviation. Results are visible in Table 9. We also evaluate the performances of a random measure: we generate random orderings of SAs and evaluate the (average) nDCG@10, this will allow us to have a baseline to compare the other measures with.

All the features have higher importance than the random measure. All the features evaluated tend to have an amount of variation the depends on the different set of SAs on which they are tested on. This might indicate that there's no best measure, since the behavior varies with respect to each user and each article considered.

3.5. Q4: Effectiveness of Contextual Exploration

In this last section we want to give preliminary insights on the SAs usefulness for the readers of

Fig. 12. Kendal τ heatmap for the similarity between features using kendall's tau



an article of additional content in the form of SAs. Our hypothesis is that a user can find interesting information that allows them to better understand the context of the article. Users were asked to give a score on a graded scale between 1 and 6 on the confidence with respect to the topic of the articles before and after evaluating the SAs. The value user could select for the answer were (No Knowledge, Poor Knowledge, Slightly Confident, Confident, Very Confident and Expert). The average value of understanding of the articles before reading SAs was 3.2 (little above the *Slightly Confident* value), while the after reading SAs score was 3.8. The increase in value suggests that the SAs extracted are useful to understand the article. To verify this hypothesis we used a non-parametric version of the ANOVA test, the Kruskal-Wallis H-test [26] on all the questionnaires we did. The test null hypothesis is that the median of the populations is equal. The test on the sample we gathered reported a p-value of 0.03 that made us reject the null hypothesis. The difference between the before and after evaluation status is thus significant. Moreover if we consider only those users (15 of them) that declared to be less confident about the topic of the article (No Knowledge, Poor Knowledge, Slightly Confident), the average value was 2.46, while the average value after was of 3.4, in this case there is a point span between the two values, meaning that the SAs reading is able to increase the un-

Table 9

Importance score of the features used in the model

	L. PRank	G. PRank	L. Auth	L. Hub	Relev.	Rarity	P. Info.	P. P. Info.	Page Views	Random
Mean	0.715	0.731	0.739	0.702	0.740	0.679	0.716	0.655	0.753	0.637
SD	0.065	0.0598	0.059	0.0643	0.0529	0.0729	0.0673	0.065	0.0542	0.08

derstanding value of the article of one point. This obliviously means that the impact of SAs is less valuable for user that are already knowledgeable about the topic: the other 10 users with gave a higher score value to the topic understanding reported an average value of understanding equal to 4.3, while the after value was 4.4, meaning that there was little increasing of knowledge for those users. Results brought us to state that SAs can be helpful for those users that are not really knowledgeable about the topic in place in the article. However, consider that the knowledge in DBpedia is quite basic. If we used a KG with more insightful facts we may enrich the user experience. In a post-questionnaire interview, users that were not expert about the political events were surprised to find that Donald Trump was formerly a member of the Democratic Party (SAs that can be found in DBpedia and was also found in some of the experimented articles). The low increase in value given by already confident users is due to the information extracted from the KG that contains general knowledge that a user already confident with the topic might know.

Discussion. We observed that different users have different interests, which motivates the need for personalization in KG exploration approaches. The serendipity heuristic with the AS algorithm model introduced in this paper shows remarkable improvement over other configurations. While the difference between Serendipity, Gaussian and Dirichlet is not statistically significant, the serendipity heuristic is able to show a pre-ordered set of SAs from the first iteration, and this might be of big interest for the user. This is important from a user experience point of view, since a user might not be interested in evaluating SAs that do not interest her.

Summary The results we got leave us with the following assertions:

– ALR is a good way to rapidly gain feedback from the user while optimizing their own ranking function
– The serendipity heuristics is a good method to initialize the ALR model and is able to show to the users a pre-ordered set of SAs from the beginning
– Our experiments showed that users are interested in different kinds of SAs and thus the personalization of the exploration is an important and fundamental task

4. Related Work

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

Interactive Knowledge Graph Exploration.

A recent survey [1] outlines different methods that, combine navigation, filtering, sampling and visualization to let users study and explore large data sets.

Applications of contextual exploration are present in literature; approaches in the entity expansion field are an example [2]. RelFinder allows users to select two entities and to see the SAs between them, but does not provide a personalization method [27]. More recent application add measures to evaluate and explain SAs between entities [7,6,28]. inWalk is an application that allows interactive linked data exploration based on thematic graphs, whose nodes represent clusters of similar data, and edges are proximity relations between these clusters [29]. Instead, Aeemo is example of a tool for knowledge exploration [30] that uses a keyword-based search that allows users to find summarized information about an entity using Wikipedia, Twitter and Google News. Another interesting approach offers users the possibility of extending their knowledge about a domain by exploring data graphs [31]. Closer to our exploration setting is Refer [3] that is a Wordpress Plugin that helps a user enrich an article with additional information extracted from a KB. The plugin finds entities in the article and recommends SAs that are estimated by unknown to the user. Refer is an example of the contextual exploration of KG; the main difference between their approach and ours is that we introduce a model that learns a personalized ranking function for each user that allows them to interactively explore semantic associations. None of the approaches mentioned above or surveyed in [1] introduces methods to learn which information to show to the users based on their explicit feedback. An interesting approach seen in the literature [32] uses methods from genetic programming to select strong relationships in linked data. Eight judges were asked to evaluate the relationships found from the algorithm, but those relationships with low inter-user agreement were removed and were not considered positive examples for training because not interesting for all users. Our approach is different, since we have found that our personalzation hypothesis holds. Thus, we train our model on the preferences of individual users using an active learning to rank approach.

Learning to Rank and Active Learning for KG Exploration Learning to rank has been extensively applied in document retrieval [33] but only in one approach to KG exploration [8], in which a variant of SVM is used to rank SAs extracted from the Freebase KB. Differently from ours, this approach does not introduce a personalization method based on learning to rank techniques. Moreover, some of the features used are specifically tailored on the Freebase KB; our features are general and can be applied to any KG (with the exception of the measure *Temporal Relevance*, which requires to find a link from the KG to Wikipedia). Active learning to rank introduces techniques to select the most informative observations to label in such a way to speed up the training of a model. In our approach, we have implemented and tested two different methods proposed for document retrieval. The first approach [10] collects labels over individual observations (that are the SAs in our case) and solves the cold-start problem, mentioned before, by randomly selecting positive and negative instances from a subset of the data reserved for training. In our interactive approach, we select the SAs that are labeled by the user from the same set that has to be ranked, which is coherent with contextual KG exploration scenarios. However, we have also conducted a cross validation test to show that the model is robust. Moreover, we have introduced a principled approach that solves the cold-start problem in this contextual exploration. The second approach we have tested collects labels over pairs of observations [24]; this approach seems to be not only less efficient but also less effective for ranking SAs. To the best of our knowledge, ours is the first attempt to apply active learning to rank to the problem of exploring SAs. While many active learning to rank techniques and methods have been proposed in literature [34,35,10,24], to the best of our knowledge none of them has been applied for SAs active ranking. We choose to concentrate our attention only on two of them, mainly because they could be easily applied in pairwise learning to rank setting with RankSVM, a well-known state-of-art algorithm in this context. One approach that has been 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 [36], which is very different from the learning to rank problem discussed in this paper.

5. Conclusion

Experimental results show that our approach, based on a serendipity heuristic and with the use of an AUC based active learning to rank algorithm can increase the ranking of the SAs while keeping the number of feedback requested to the user low. Moreover, we have been able to prove that personalization of KG exploration is necessary since users are interested in different kinds of contextual information and that the use of an active learning to rank approach can help to optimize the personalization function.

In future work, we plan to analyze the impact of individual features in an actual active learning context bypassing the static evaluation based on learning to rank only. In addition, we want to implement our active learning to rank workflow in the DaCENA application. This will require tackling the challenge of designing human-data interaction pipelines that can be effectively used by the end users. Another research area that we plan to explore is the one of online learning to rank with the use of implicit feedback (like clicks) given by the users on the SAs; combining implicit and explicit feedback could significantly improve the performance of the models while lowering the effort requested to the user to get personalized results. Another step that we need to take is to make our application faster in such way tat it can be used in real time contexts. So far, we preferred to collect fresher information via the use of SPARQL queries to the DBpedia endpoint despite the longer processing time. In journalism, the freshness of information is relevant, and we plan to study further techniques that will allow us to update SAs even after they have been processed.

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. *preprint arXiv:1601.08059*, 2016.

[2] José Luis Redondo-García, Michiel Hildebrand, Lilia Perez Romero, and Raphaël Troncy. Augmenting TV newscasts via entity expansion. In *ESWC*, pages 472–476. Springer, 2014.

[3] 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, 2016.

[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] 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.

[6] Giuseppe Pirrò. Explaining and suggesting relatedness in knowledge graphs. In *ISWC*, pages 622–639. Springer, 2015.

[7] Gong Cheng, Yanan Zhang, and Yuzhong Qu. Explass: exploring associations between entities via top-k ontological patterns and facets. In *ISWC*, pages 422–437. Springer, 2014.

[8] Na Chen and Viktor K Prasanna. Learning to rank complex semantic relationships. *IJSWIS*, 8(4):1–19, 2012.

[9] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[10] Pinar Donmez and Jaime G Carbonell. Active sampling for rank learning via optimizing the area under the ROC curve. In *ECIR*, pages 78–89. Springer, 2009.

[11] Federico Bianchi, Matteo Palmonari, Marco Cremaschi, and Elisabetta Fersini. Actively learning to rank semantic associations for personalized contextual exploration of knowledge graphs. In *ESWC*, 2017.

[12] Ching-Pei Lee and Chih-Jen Lin. Large-scale linear ranksvm. *Neural computation*, 26(4):781–817, 2014.

[13] R Busa-Fekete, György Szarvas, Tamás Elteto, and Balázs Kégl. An apple-to-apple comparison of learning-to-rank algorithms in terms of normalized discounted cumulative gain. In *20th European Conference on Artificial Intelligence (ECAI 2012): Preference Learning: Problems and Applications in AI Workshop*, volume 242. Ios Press, 2012.

[14] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374, 2010.

[15] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[16] 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 *PAKDD*, pages 384–388. Springer, 2004.

[17] Pang-Ning Tan et al. *Introduction to data mining.* Pearson Education India, 2006.

[18] 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, 2016.

[19] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 46(5):604–632, 1999.

[20] 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.

[21] Xiubo Geng, Tie-Yan Liu, Tao Qin, and Hang Li. Feature selection for ranking. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 407–414. ACM, 2007.

[22] Kilem L Gwet. *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters.* Advanced Analytics, LLC, 2014.

[23] Weiwei Yuan, Yongkoo Han, Donghai Guan, Sungyoung Lee, and Young-Koo Lee. Initial training data selection for active learning. In *ICUIMC*, page 5. ACM, 2011.

[24] 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.

[25] Ong Kok Chien, Poo Kuan Hoong, and Chiung Ching Ho. A comparative study of hits vs pagerank algorithms for twitter users analysis. In *Computational Science and Technology (ICCST), 2014 International Conference on*, pages 1–6. IEEE, 2014.

[26] William H Kruskal and W Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.

[27] Philipp Heim, Sebastian Hellmann, Jens Lehmann, Steffen Lohmann, and Timo Stegemann. Relfinder: Revealing relationships in rdf knowledge bases. In *SAMT*, pages 182–187. Springer, 2009.

[28] Lujun Fang, Anish Das Sarma, Cong Yu, and Philip Bohannon. Rex: explaining relationships between entity pairs. *Proceedings VLDB*, 5(3):241–252, 2011.

[29] Silvana Castano, Alfio Ferrara, and Stefano Montanelli. inwalk: Interactive and thematic walks inside

the web of data. In *EDBT*, pages 628–631. Citeseer, 2014.

[30] Andrea Giovanni Nuzzolese, Valentina Presutti, Aldo Gangemi, Alberto Musetti, and Paolo Ciancarini. Aemoo: exploring knowledge on the web. In *ACM Web Science Conference*, pages 272–275. ACM, 2013.

[31] Marwan Al-Tawil, Vania Dimitrova, and Dhavalkumar Thakker. Using knowledge anchors to facilitate user exploration of data graphs. *Semantic Web*, (Preprint):1–30, 2019.

[32] Ilaria Tiddi, Mathieu dAquin, and Enrico Motta. Learning to assess linked data relationships using genetic programming. In *ISWC*, pages 581–597. Springer, 2016.

[33] Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[34] Bo Long, Olivier Chapelle, Ya Zhang, Yi Chang, Zhaohui Zheng, and Belle Tseng. Active learning for ranking through expected loss optimization. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. ACM, 2010.

[35] Wei Chu and Zoubin Ghahramani. Extensions of gaussian processes for ranking: semisupervised and active learning. *Learning to Rank*, page 29, 2005.

[36] Meng Fang, Jie Yin, and Xingquan Zhu. Active exploration for large graphs. *Data Mining and Knowledge Discovery*, 30(3):511–549, 2016.