RELIN: Relatedness and Informativeness-based Centrality for Entity Summarization

Gong Cheng[1], Thanh Tran[2], and Yuzhong Qu[1]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China
[2] Institute AIFB, Karlsruhe Institute of Technology, D-76131 Karlsruhe, Germany
{gcheng,yzqu}@nju.edu.cn, ducthanh.tran@kit.edu

Abstract. Linked Data is developing towards a large, global repository for structured, interlinked descriptions of real-world entities. An emerging problem in many Web applications making use of data like Linked Data is how a lengthy description can be tailored to the task of quickly identifying the underlying entity. As a solution to this novel problem of entity summarization, we propose RELIN, a variant of the random surfer model that leverages the relatedness and informativeness of description elements for ranking. We present an implementation of this conceptual model, which captures the semantics of description elements based on linguistic and information theory concepts. In experiments involving real-world data sets and users, our approach outperforms the baselines, producing summaries that better match handcrafted ones and further, shown to be useful in a concrete task.

Keywords: Distributional relatedness, entity summarization, informativeness, PageRank, random surfer model.

1 Introduction

Linked Data can be conceived as a large collection of entity descriptions. As descriptions evolve on the Linked Data Web, they are linked to others. The result is that descriptions become increasingly lengthy. Already today, lengthy descriptions can be found in many existing data sets. For instance, the latest version of the well-known DBpedia data set[3] describes 3.5 million entities with 672 million facts (i.e. RDF triples). This means each entity description is associated with an average of 192 RDF triples. Lengthy descriptions take long time for human users to read, which is unacceptable in tasks that require quick identification of the underlying entities. For example during entity search [5, 14], users want to quickly browse through search results to identify the ones that match a given information need. Another task is pay-as-you-go data integration [11, 19], where users evaluate entity mappings computed by the matching system by identifying the referred entities and judging whether they denote the same thing. To improve the efficiency of these tasks, we aim at solving this novel problem that we

[3] http://dbpedia.org/.

call *entity summarization* to produce a version of the original description that is more concise, yet containing sufficient information for users to quickly identify the underlying entity.

The more general problem of *data summarization* has been studied by different communities. For example, *database* [2] and *graph summarization* [13] compute compact representations of data that generalize the original data elements (e.g. cells in a data cube, or a graph) to a more coarse-grained level (e.g. dimension-based regions, or an aggregated graph). That is, data elements are categorized, and then are compactly represented using the resulting categories. However, this is proposed for lossless or lossy (but with bounded errors) data representation, which is distinct from the summary pursued in our problem of entity summarization that is for facilitating quick identification of the underlying entity, or in other words, for helping to efficiently distinguish one entity from others. Thereby, rather than categorization, a solution needed here could be a way of selecting a few central data elements that are most useful in characterizing an entity. This is more similar to *extractive text* [8] and *ontology summarization* [20], the goal of which is to find the central topics of the given data (e.g. a document or an ontology). Unlike categories in database and graph summaries, a topic here is an element extracted from the original data, e.g. a text sentence or an ontology element. To find central elements, the notion of centrality is often employed. Existing approaches [8, 20] mainly simulate a random surfer's behavior (as in PageRank [15]), and incorporate data elements that are most likely to be visited by the surfer into the summary. We follow this line of research in our work.

To summarize, we propose to look at this novel (1) *problem of entity summarization*. In this first (to the best of our knowledge) solution to the problem, we elaborate on (2) *a variant of the random surfer model*. This well-known model is used as the basis to support the idea of incorporating central elements into the summary. However, it is revised by a more specific notion of centrality, called RELIN, where the computation of central elements involves relatedness (or similarity) between elements as well as their informativeness, i.e. the amount of information carried that helps to identify the entity. It extends the previous idea of capturing the main themes [8, 20] that describe the data, to find more specific central elements that identify the data. To this end, instead of a traditional random surfer, we simulate a rather goal-directed surfer that explores an entity description with the aim of identifying the underlying entity. We model two kinds of action, namely relational move and informational jump, that follow non-uniform probability distributions. The surfer, to achieve her goal, prefers related elements when she moves, and prefers informative elements when she jumps. We propose a simple but effective (3) *implementation of these notions of relatedness and informativeness* that exploits the semantic information captured by the graph structure of the data (as in [20]) as well as the labels of nodes and edges. For the latter, we apply well-known linguistic and information theory concepts. We carried out an extensive (4) *empirical study* of the proposed approach. The results show that it significantly outperformed the baseline approaches, both

in an intrinsic evaluation based on a comparison with handcrafted summaries, and in an extrinsic evaluation where the computed summaries are used for the task of confirming entity mappings.

The remainder of this paper is organized as follows. The problem is defined in Sect. 2. The approach is detailed in Sect. 3, and an implementation is given in Sect. 4. Related work is discussed in Sect. 5. Experimental results are presented in Sect. 6 before we conclude in Sect. 7.

2 Problem Statement

For the investigated problem, we employ a graph-structured data model corresponding to RDF, which describes entities in the form of attribute values and relations to other entities (collectively called property values). Let E be the set of all *entities*, L the set of all *literals*, and P the set of all *properties*.

Definition 1 (Data Graph). *A* data graph *is a digraph* $G = \langle V, A, \mathrm{Lbl}_V, \mathrm{Lbl}_A \rangle$, *where* V *is a finite set of nodes,* A *is a finite set of directed edges where each* $a \in A$ *has a source node* $\mathrm{Src}(a) \in V$ *and a target node* $\mathrm{Tgt}(a) \in V$, *and* $\mathrm{Lbl}_V : V \mapsto E \cup L$ *and* $\mathrm{Lbl}_A : A \mapsto P$ *are labeling functions that map nodes and edges to entities or literals, and properties, respectively.*

Definition 2 (Feature). *A* feature f *is a property-value pair where* $\mathrm{Prop}(f) \in P$ *and* $\mathrm{Val}(f) \in E \cup L$ *denote the property and the value, respectively. An entity* e *has a feature* f *in a data graph* $G = \langle V, A, \mathrm{Lbl}_V, \mathrm{Lbl}_A \rangle$ *if there exists* $a \in A$ *such that* $\mathrm{Lbl}_A(a) = \mathrm{Prop}(f)$, $\mathrm{Lbl}_V(\mathrm{Src}(a)) = e$ *and* $\mathrm{Lbl}_V(\mathrm{Tgt}(a)) = \mathrm{Val}(f)$.

That is, a feature of an entity corresponds to one of its associated edges in the data graph. We actually consider both incoming and outgoing edges (i.e. where e appears as target and source node, respectively). Without loss of generality, we focus on outgoing edges for the sake of clear presentation.

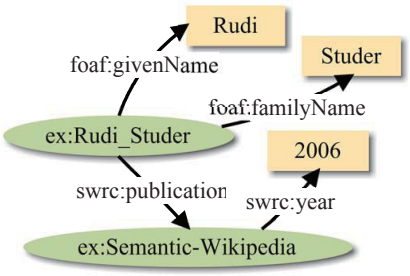
A feature is regarded as the smallest meaningful description element for an entity, based on which we characterize an entity description as a set of features:

Definition 3 (Feature Set). *Given a data graph* G, *the* feature set *of an entity* e, *denoted by* $\mathrm{FS}(e)$, *is the set of all features of* e *that can be found in* G.

The left part of Fig. 1 depicts the data graph for our running example, which describes a person and one of his publications. Given this data graph, the feature set of the entity `ex:Rudi_Studer` is shown in the right part of Fig. 1.

Finally, the problem of entity summarization is defined as extracting a subset from a lengthy feature set, subject to a cardinality constraint.

Definition 4 (Entity Summarization). *Given* $\mathrm{FS}(e)$ *and a positive integer* $k < |\mathrm{FS}(e)|$, *the problem of* entity summarization *is to select* $\mathrm{Summ}(e) \subset \mathrm{FS}(e)$ *such that* $|\mathrm{Summ}(e)| = k$. $\mathrm{Summ}(e)$ *is called a* summary *of* e.



FS(ex:Rudi_Studer)	
f_1	<foaf:givenName, "Rudi">
f_2	<foaf:familyName, "Studer">
f_3	<swrc:publication, ex:Semantic-Wikipedia>

Fig. 1. The feature set of the entity `ex:Rudi_Studer` (on the right), given the data graph (on the left) containing two entities (ellipses) and three literals (rectangles).

In the running example, valid summaries of `ex:Rudi_Studer` include $\{f_1, f_2\}$, $\{f_1, f_3\}$ and $\{f_2, f_3\}$ given $k = 2$. In the next sections, we will introduce an approach to finding a summary from $\binom{|\mathrm{FS}(e)|}{k}$ candidates that best characterizes e for quick identification.

It is worth noting that we impose a length constraint on the summary based on the number of features. In fact, the content of individual features may also be a factor that deserves consideration because, for instance, features may contain literals that significantly vary in length. However, this will not be investigated in our work. Besides, we actually concentrate on *what* information (i.e. which features) should be presented, but will not address *how* this information should be presented (e.g. by using visualization or natural language generation methods), although the latter is also an important part of the summarization task.

3 Entity Summarization

We conceive the problem of entity summarization as the one of ranking, i.e. selecting the k top-ranked features from the feature set for a summary. In this sense, entity summarization and feature ranking refer to the same task.

3.1 Centrality-based Ranking

Centrality-based ranking has been successfully applied to text [8] and ontology summarization [20], and we follow this direction to solve entity summarization. This paradigm requires constructing a graph where nodes correspond to the data elements to be ranked, i.e. sentences in text summarization [8], RDF sentences in ontology summarization [20], and features in entity summarization. Every pair of related nodes are connected by undirected [8] or directed edges [20], and such pairs could be defined based on some numerical relatedness measures with a predefined threshold [8] or problem-specific heuristics [20]. Finally, nodes are ranked according to their centralities in the graph, often computed by using PageRank [15]. Basically, PageRank simulates a surfer, who navigates from node

to node, choosing with a uniform probability which edge to follow at each step, and with a small probability, occasionally jumps to a random node; the ranking of nodes is obtained by considering the stationary distribution of such a Markov chain, and a node with a higher probability of being reached by the surfer is ranked higher. In this way, top-ranked nodes (i.e. data elements) are believed to capture the main themes of the original data, since they are central to the original data with regard to the relatedness among data elements.

However, applying the random surfer model like this to our scenario yields two problems. (1) The supported notion of centrality may be too general. Capturing the main themes of the original entity description is not the only goal pursued here. Recall that the summary we are looking for is the one that can best characterize the underlying entity and help to distinguish the entity from others. That is, the measurement of centrality should also give consideration to *how much information a feature carries that can contribute to the identification of the entity*. (2) To apply this random surfer model, edges are added between "significantly related" nodes, where relatedness is actually defined as a boolean-valued function: nodes are either related (and thus connected by an edge) or not (and thus not adjacent). Then, all the adjacent nodes of a node are treated as being equally related to it, since the surfer chooses from them with a uniform probability which one to visit. In other words, the model does not *represent the degree of relatedness on a more fine-grained level*. This imprecision may lead to suboptimal results, particularly when such a boolean-valued function is derived from a relatedness threshold, as it is often the case.

3.2 RELIN: Relatedness and Informativeness-based Centrality

To remedy the flaws pointed out above, we extend the standard random surfer model as follows. For the first issue, inspired by [10], we propose to embed the measurement of informativeness in the random surfer model. Recall that in the standard model, the surfer jumps to a random node with a given probability. We replace this uniform probability distribution with a non-uniform one that is *dependent on the amount of information carried by each target node that helps to identify the entity*. As a result, a feature that is informative in terms of distinguishing the underlying entity from others will more likely to be reached by the surfer, and thus will be ranked higher. For the second issue, we propose to construct an edge-labeled complete graph, as illustrated in Fig. 2 (solid lines). Then the surfer at a node chooses which edge to follow not with a uniform probability but with a probability (derived from the label of the edge) *proportional to the relatedness between the two associated nodes* (i.e. the current node and the target). In this way, we avoid the problem of finding the most appropriate threshold (which is shown to be difficult [8]) and can also fully exploit the computed numerical relatedness values.

To be specific, we propose RELIN, a variant of the random surfer model that measures RELatedness and INformativeness-based graph centrality for entity summarization. Similar to the standard model in PageRank, we simulate a random surfer's behavior using two kinds of action, one called *relational move* and

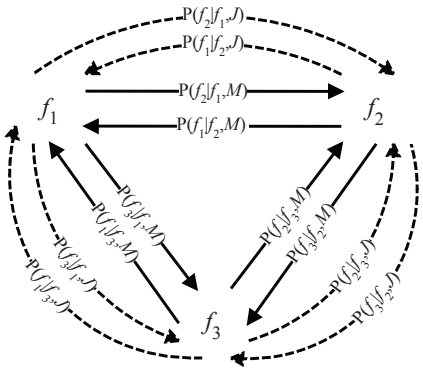


Fig. 2. A graph under the RELIN random surfer model, where nodes represent features, solid lines represent relational moves (i.e. edges) between features, and dashed curves represent informational jumps between features. Each action is associated with a non-uniform probability.

the other *informational jump*. This hypothetical surfer is a goal-directed one that navigates through a feature set in order to identify the underlying entity. To this end, the surfer either performs a relational move — more likely to a feature that carries related information about the theme currently under investigation, or performs an informational jump — more likely to a feature that provides a large amount of new information for clarifying the identity of the underlying entity. These choices are represented by two non-uniform probability distributions, one given by the relatedness between features and the other by the informativeness of features. For the running example, Fig. 2 illustrates the graph under this new random surfer model.

Now we formalize our solution using a general probabilistic framework [7]. The surfer's behavior in RELIN, namely relational move (M) and informational jump (J), is defined with respect to the current feature f_q:

- $\mathrm{P}(M|f_q)$: the probability of performing a relational move from f_q, and
- $\mathrm{P}(J|f_q)$: the probability of performing an informational jump from f_q.

There exist only two kinds of action, and thus they satisfy $\mathrm{P}(M|f_q)+\mathrm{P}(J|f_q) = 1$. Then both actions are defined with targets:

- $\mathrm{P}(f_p|f_q, M)$: the probability of performing a relational move from feature f_q to feature f_p, and
- $\mathrm{P}(f_p|f_q, J)$: the probability of performing an informational jump from feature f_q to feature f_p.

These sets of probabilities must satisfy the following normalization constraints for each $f_q \in \mathrm{FS}$, where FS is the feature set under consideration:

- $\sum_{f_p \in \mathrm{FS}} \mathrm{P}(f_p|f_q, M) = 1$, and

– $\sum_{f_p \in \mathrm{FS}} \mathrm{P}(f_p|f_q, J) = 1$.

Let $\mathbf{x}(t)$ be a $|\,\mathrm{FS}\,|$-dimensional vector where $\mathbf{x}_p(t)$ is the probability that the surfer visits feature f_p at step t. By taking all the possibilities of the surfer's behavior into account, the probability $\mathbf{x}_p(t+1)$ is updated as follows:

$$\mathbf{x}_p(t+1) = \sum_{f_q \in \mathrm{FS}} \mathbf{x}_q(t) \cdot (\,\mathrm{P}(M|f_q) \cdot \mathrm{P}(f_p|f_q, M) + \mathrm{P}(J|f_q) \cdot \mathrm{P}(f_p|f_q, J))\,. \tag{1}$$

All the above probabilities defining the RELIN random surfer model can be organized into the following $|\,\mathrm{FS}\,| \times |\,\mathrm{FS}\,|$ matrices:

– $\mathbf{M}$, where $\mathbf{M}_{p,q} = \mathrm{P}(f_p|f_q, M)$,
– $\mathbf{J}$, where $\mathbf{J}_{p,q} = \mathrm{P}(f_p|f_q, J)$,
– $\mathbf{\Delta}$, a diagonal matrix where $\mathbf{\Delta}_{q,q} = \mathrm{P}(M|f_q)$, and
– $\mathbf{\Lambda}$, a diagonal matrix where $\mathbf{\Lambda}_{q,q} = \mathrm{P}(J|f_q)$.

Then (1) can be rewritten as:

$$\mathbf{x}(t+1) = (\mathbf{M} \cdot \mathbf{\Delta} + \mathbf{J} \cdot \mathbf{\Lambda}) \cdot \mathbf{x}(t)\,. \tag{2}$$

It has been proved [7] that:

$$\lim_{t \to \infty} \mathbf{x}(t) = \mathbf{x}^*\,, \tag{3}$$

where $\mathbf{x}^*$ is a constant vector that does not depend on the initial distribution $\mathbf{x}(0)$, if $\mathrm{P}(J|f_q) \neq 0$ and $\mathrm{P}(f_p|f_q, J) \neq 0$ for every $f_p, f_q \in \mathrm{FS}$. In practice, the iterative computation of (2) is usually configured to stop after a certain number of iterations.

Finally, features in FS are ranked by $\mathbf{x}^*$. That is, feature f_p will be ranked higher than feature f_q if $\mathbf{x}^*_p > \mathbf{x}^*_q$.

To implement this model, we need to give $\mathbf{M}$, $\mathbf{J}$, $\mathbf{\Delta}$ and $\mathbf{\Lambda}$, i.e., to define $\mathrm{P}(f_p|f_q, M)$, $\mathrm{P}(f_p|f_q, J)$, $\mathrm{P}(M|f_q)$ and $\mathrm{P}(J|f_q)$ for every $f_p, f_q \in \mathrm{FS}$. This will be discussed in the next section.

4 Implementation

Firstly, we define $\mathbf{\Delta}$ and $\mathbf{\Lambda}$ as follows:

$$\begin{aligned} \mathbf{\Delta}_{q,q} &= 1 - \lambda, \quad q = 1, \ldots, |\,\mathrm{FS}\,|\,, \\ \mathbf{\Lambda}_{q,q} &= \lambda, \quad q = 1, \ldots, |\,\mathrm{FS}\,|\,, \end{aligned} \tag{4}$$

where $\lambda \in [0, 1]$ is regarded as a parameter to be tuned and tested in experiments. Here we actually follow PageRank to assume that the surfer has a consistent probability of choosing between move and jump.

Next, as a general strategy for computing $\mathbf{M}$, the relatedness between features, and $\mathbf{J}$, the informativeness of features, we propose to exploit the information captured by the labels of nodes and edges in the original data graph. We now discuss one possible implementation.

4.1 Relatedness

Two features, i.e. property-value pairs, are related when they have related properties, e.g. "has paper" and "has research interest" which are both about academic information, or they have related values, e.g. a paper titled "Semantic Wikipedia" and a research interest "Semantic Web". Thus, we define the relatedness between two features as a combination of the relatedness (denoted by Rel) between their properties and the relatedness between their values (subjected to the mentioned probability normalization):

$$\mathbf{M}_{p,q} = \sqrt{\mathrm{Rel}(\mathrm{Prop}(f_p), \mathrm{Prop}(f_q)) \cdot \mathrm{Rel}(\mathrm{Val}(f_p), \mathrm{Val}(f_q))}\,. \tag{5}$$

As potential implementations of Rel, various notions of relatedness have been proposed in the literature. Among others, a well-known line of research [3] employs semantic network such as WordNet[4] to measure the relatedness between text phrases, usually based on the length of the shortest path between their corresponding nodes in the network. However, it is difficult to find such a source of background knowledge that has a good coverage of all the properties and values in our problem that might be encountered in practice. Therefore, we employ another notion called *distributional relatedness* [12], i.e., two phrases are more related if they more often co-occur in certain contexts (e.g. documents). We use an implementation called *Pointwise Mutual Information* (PMI). Let $\mathrm{P}(s_i)$ be the probability that phrase s_i occurs in a document, which could be estimated by counting throughout a corpus. Analogously, let $\mathrm{P}(s_i, s_j)$ be the joint probability of phrase s_i and phrase s_j. Their PMI is defined as follows:

$$\mathrm{PMI}(s_i, s_j) = \log \frac{\mathrm{P}(s_i, s_j)}{\mathrm{P}(s_i) \cdot \mathrm{P}(s_j)}\,. \tag{6}$$

Note that to estimate this for every edge in a graph under the RELIN random surfer model, we need to obtain probabilities for every entity, literal and property that might be mentioned in the nodes (i.e. features). Achieving this coverage requires a large and diverse corpus. To this end, we leverage the Google search engine to obtain the contexts in which phrases may co-occur. Let $\mathrm{Hits}(s_i)$ be the number of documents returned by the search engine that match phrase s_i (which could be the name of a property or an entity, or the lexical form of a literal), and N a predefined normalizing constant. Then, we estimate $\mathrm{P}(s_i, s_j)$, $\mathrm{P}(s_i)$ and $\mathrm{P}(s_j)$ by computing $\frac{\mathrm{Hits}(s_i, s_j)}{N}$, $\frac{\mathrm{Hits}(s_i)}{N}$ and $\frac{\mathrm{Hits}(s_j)}{N}$, respectively.

For instance, in the running example, f_1 is more related to f_2 than to f_3, mainly because their property names "given name" and "family name" have a higher PMI than "given name" and "publication" have.

It is worth noting that the use of a Web search engine could become a performance bottleneck that limits a practical summarization system. Solutions include completing all (or most) potential queries prior to placing the system in service, or using a local corpus instead.

[4] `http://wordnet.princeton.edu/`.

4.2 Informativeness

We use a well-known information theory concept to measure the informativeness of features. Given o an outcome of a random variable with probability $\mathrm{P}(o)$, its *self-information* is defined as:

$$\mathrm{SelfInfo}(o) = -\log(\mathrm{P}(o))\,. \tag{7}$$

That is, the smaller probability an outcome has, the more information its occurrence provides. In this sense, according to the RELIN random surfer model, we should look at $\mathrm{P}(f_p|f_q)$ — the probability that feature f_p belongs to a feature set, given f_q belongs to the same one. This can be estimated via a statistical analysis of the original data graph:

$$\mathrm{P}(f_p|f_q) = \frac{|\{e \in E \,|\, f_p, f_q \in \mathrm{FS}(e)\}|}{|\{e \in E \,|\, f_q \in \mathrm{FS}(e)\}|}\,. \tag{8}$$

Then, the amount of new information that the surfer obtains by performing an informational jump from f_q to f_p is measured by (subjected to the mentioned probability normalization):

$$\mathbf{J}_{p,q} = \mathrm{SelfInfo}(f_p|f_q) = -\log(\mathrm{P}(f_p|f_q))\,. \tag{9}$$

For instance, given f_1 in the running example, in terms of distinguishing the underlying entity from others, f_3 is more informative than f_2 because there is only one Rudi that is an author of the publication "Semantic Wikipedia", but there are probably more than one Rudi whose family name is Studer.

It is worth noting that, when computing $\mathrm{SelfInfo}(f_p|f_q)$ between all pairs of features is too costly in practice, $\mathrm{SelfInfo}(f_p)$ can be used as an approximation. This would assume that the informativeness of one feature is independent from the information provided by other features such that for informational jump, only the information of the targets plays a role.

5 Related Work

Summarization methods can be classified into *extractive* and *non-extractive* ones. Most text [18] and ontology summarization [16, 20] work employs the more popular extractive strategies, which produce a summary by choosing a subset from the original data elements. We follow it by conceiving an entity description to be summarized as a feature set. On the other hand, database [2] and graph summarization [13] usually adopt the non-extractive paradigm, and define the notion of summary on a level that is more coarse-grained than the original data.

The adoption of a PageRank-like [15] graph centrality measure for entity summarization here is motivated by research in related fields. Firstly, for the closely related problem of text summarization, *centrality*-based methods (e.g. [8]) have proven to be superior to those simple *centroid*-based statistical methods (e.g. [17]), which basically rank data elements according to their relatedness to

the "centroid" of the entire data. A detailed theoretical and empirical comparison of these two styles is given in [8]. Secondly, among various notions of graph centrality, we prefer a *principled PageRank-like measure* mainly because PageRank has shown a better performance than its competitors (e.g. degree-based measures) in the experiments on previous summarization tasks [8, 20].

Although the proposed RELIN approach also builds upon the random surfer model [15] as in previous methods [8, 16, 20], it is about computing central elements (i.e. features) that not just represent the main themes of the original data, but rather, can best *identify the underlying entity*. Thus, instead of a traditional random surfer, we simulate a goal-directed one that has a *preference for related and informative features*. Further, different from the standard model, the surfer's behavior in RELIN is characterized by *non-uniform probability distributions*. This idea of assigning different weights to different actions in the random surfer model has been investigated, among others, for dealing with the problem of Web search [10]. Besides dealing with a different problem, the proposed approach also uses information that is completely different from the notions of relatedness and informativeness implemented in our work.

Complementary to centrality, *diversity* [4] is another popular metric for evaluating a summary, by measuring its coverage of themes in the original data. In fact, this diversity aspect is partially supported by our implementation, since according to Sect. 4.2, informational jumps are dependent on the amount of "new" information. However, this matter is not elaborated in the paper because it is orthogonal to our concern in the sense that it can be easily incorporated into our approach as a re-ranking step, as proposed before [4, 20].

From another point of view, our work is also related to the topic of *ranking in RDF graphs*, for which different ideas have been studied. For instance, [6] performs a hierarchical link analysis for ranking entities; [9] applies tensor decomposition to find latent aspects of the data and generate aspect-specific rankings; [1] ranks associations (i.e. paths) between entities by means of a wide range of customizable metrics. However, none of these approaches is well fit for the problem of entity summarization here, which requires ranking data elements according to *how much they help identify the underlying entity*. In this respect, [16, 20] are the most related work: basically, an RDF graph is decomposed into a set of subgraphs called "RDF sentences"; based on the common nodes they share, links are defined between them, from which a new graph with nodes representing RDF sentences is derived; then, various graph centrality measures (e.g. PageRank) are applied to this new graph for ranking. In comparison, our work leverages the information contained in the *labels of nodes and edges in the original data graph*. This goes beyond [16, 20], which mainly employ the *graph structure* for ranking.

6 Experiments

In the experiments, two real-world data graphs were used: (1) the English version of the DBpedia 3.4 core data sets, which collectively contain 124,404,962 RDF

triples,[5] and (2) the December 2009 Link Export of the Freebase data set,[6] which contains 325,158,504 RDF triples.[7] Both are domain-independent encyclopedic data sets, which belong to the largest that have been made publicly available on the Web as part of the Linked Data initiative. They cover a broad range of descriptions of entities such as people, cities and music albums.

We implemented RELIN as described previously in the paper. For the parameter λ in (4), which assigns the importance of informational jump relative to relational move, we tested 5 different values, namely 0.00, 0.15, 0.50, 0.85 and 1.00. Particularly, with $\lambda = 0.00$, our approach relies only on relatedness between features, which can then be regarded as an application of traditional text summarization methods (e.g. [8]) to entity summarization. On the contrary, with $\lambda = 1.00$, it amounts to one that employs informativeness of features only. Besides, the iterative computation in RELIN was set to stop after 10 iterations.

We intend to compare our approach with other work on ranking RDF data. However, as discussed in Sect. 5, no existing method is well-suited to our problem. Thus, to establish baselines, on the one hand, we implemented *OntoSum*, which is an adaptation of the most related approach given in [20] to our problem of ranking features.[8] To be specific, given the data graph comprising all the features of the entity under consideration, the notion of RDF sentence proposed in [20] amounts to one single RDF triple, which further corresponds to a feature of the entity. Thereby, a ranking of RDF sentences produced by [20] would naturally induce a ranking of features. On the other hand, we also implemented *RandomRank* that always produces a random ranking of features.

We ran two independent evaluations. In an *intrinsic* one, automatically computed summaries were compared with ideal ones. The other *extrinsic* evaluation aimed at investigating the usefulness of the summaries in a practical task.

6.1 Intrinsic Evaluation

In the intrinsic evaluation, 24 participants (comprising graduate and undergraduate students majoring in computer science) were invited to manually construct ideal entity summaries as the gold standard. A sample of 149 entities were selected at random from DBpedia under the constraint that the cardinality of each one's feature set is inside the interval [20,40], such that it is neither too small to be significant for a summarization task nor too lengthy for manual investigation. Then, each entity was randomly assigned to an average of 4.43 participants; given an entity description presented as a list of features sorted in random order, a participant was asked to return two ideal summaries — one containing

[5] The data sets *Links to Wikipedia Article* and *External Links* were not imported since they are less relevant to the summarization task.

[6] `http://www.freebase.com/`.

[7] The RDF triples that involve non-English literals were removed since our participants cannot read.

[8] Among several centrality measures compared in [20], we chose the best-preforming one, namely PageRank.

5 features and the other containing 10 — that could best clarify the identity of the underlying entity. That is, given $k \in \{5, 10\}$ and an entity e, we would receive, from n different participants, n ideal summaries, denoted by $\mathrm{Summ}_i^I(e)$ for $i = 1, \ldots, n$.

Firstly, we report the level of agreement between ideal summaries. Given $k \in \{5, 10\}$, an entity e and n ideal summaries received, their *agreement* is defined by their average overlap:

$$\frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} |\,\mathrm{Summ}_i^I(e) \cap \mathrm{Summ}_j^I(e)|\,. \tag{10}$$

In our experiments, when $k = 5$, the agreement averaged over all the entities is 2.91 features, and when $k = 10$, the overall agreement is 7.86. These indicate a *significant level of agreement between participants about ideal summaries.*

Secondly, based on ideal summaries, for a summary automatically computed, denoted by $\mathrm{Summ}(e)$, its *quality* is evaluated by looking at the average overlap between $\mathrm{Summ}(e)$ and each $\mathrm{Summ}_i^I(e)$:

$$\mathrm{Quality}(\mathrm{Summ}(e)) = \frac{1}{n} \sum_{i=1}^{n} |\,\mathrm{Summ}(e) \cap \mathrm{Summ}_i^I(e)|\,. \tag{11}$$

Table 1 presents the quality of summaries computed under each approach setting, averaged over all the entities. The best quality values are highlighted.

Table 1. Quality of summaries computed under each approach setting

	$k = 10$	$k = 5$
OntoSum	3.69	1.01
RandomRank	3.36	0.76
RELIN, with $\lambda = 0.00$	3.58	1.61
RELIN, with $\lambda = 0.15$	3.84	1.73
RELIN, with $\lambda = 0.50$	4.40	1.99
RELIN, with $\lambda = 0.85$	**4.88**	2.29
RELIN, with $\lambda = 1.00$	4.86	**2.40**

Our approach, under almost all the tested values of λ, outperformed the two baselines. When $k = 10$, compared with *OntoSum* and *RandomRank*, the quality achieved by RELIN increases by up to 32.2% and 45.2%, respectively. When $k = 5$, the highest increases are 137.6% and 215.8%, respectively. These results suggest that w.r.t. entity summarization, *our approach is clearly superior to the most related competitor in the literature, and both are better than a random selection.*

By testing different values of λ, we found that informativeness (i.e. when $\lambda = 1.00$) is more effective than relatedness (i.e. when $\lambda = 0.00$), particularly

in generating extremely short summaries ($k = 5$), where it achieved the best result. That means, it seems the participants preferred only to jump from one informative feature to one another. However, it changed when the summaries became longer. For instance, when $k = 10$, the best result was achieved when $\lambda = 0.85$, i.e. the use of informativeness in combination with relatedness. Thus, these results suggest that *whereas both actions are useful, the choice of λ should be tested and tuned in experiments, and the defined length of summary is one important factor that determines this trade-off.*

6.2 Extrinsic Evaluation

In the extrinsic evaluation, 19 participants (comprising graduate and undergraduate students majoring in computer science) were invited to confirm entity mappings. That is, given two (summaries of) descriptions, a participant was asked to judge whether they refer to the same underlying entity. The accuracy and efficiency of these judgments would reflect the usefulness of automatically computed summaries when being applied to assist users in this particular task. A sample of 47 pairs of entities were used in the experiments. Each pair, consisting of one entity from DBpedia and the other from Freebase, is either correct (i.e. referring to the same real-world entity) or incorrect (i.e. referring to different real-world entities). These pairs were constructed as follows. Firstly, a sample of 47 entities were selected at random from DBpedia under the constraint that when submitting each one's name as a keyword query against the Freebase search engine, at least two entities could be retrieved. Then, the DBpedia entity and one Freebase entity randomly selected from the top-2 search results formed an entity mapping. If such a mapping could be found in the DBpedia extended data set *Links to Freebase*, which explicitly defines entity mappings (in the form of `owl:sameAs` relation) between DBpedia and Freebase, it was deemed correct, or otherwise incorrect. In this way, we obtained 24 correct mappings and 23 incorrect ones. These judgments were used as gold-standard answers.

For each mapping, under each of the five approach settings as shown in Table 2, the two entity descriptions were summarized and then were randomly and blindly assigned to an average of 3.62 participants to judge. Each summary was presented as a list of features sorted by their ranking values. In particular, under the *ReturnsAll* setting, all the features in a description would be presented in random order without summarization. To compare different approach settings, we examined the (1) *accuracy* of the judgments made by the participants, and the (2) *time* they spent. The accuracy of a judgment is 1.0 if it coincides with the gold standard, or otherwise 0.0. To eliminate the difference in participants' intrinsic efficiency,[9] before aggregation, every time value spent by a participant was normalized by the average time per judgment spent by this participant. In this sense, 1.0 would mean medium efficiency, when smaller values indicate higher efficiency. Table 2 summarizes the experimental results averaged over all the mappings and participants, where better results are highlighted.

[9] For example, an inefficient participant would unfairly increase the overall time spent under those approach settings that she was involved in.

Table 2. Accuracy and time for judgments using summaries computed under each approach setting

	k	Accuracy	Time
OntoSum	5	0.56	**0.84**
RandomRank	5	0.60	**0.87**
RELIN, with $\lambda = 0.85$	5	**0.70**	**0.92**
RELIN, with $\lambda = 0.85$	10	**0.68**	1.12
ReturnsAll	n/a	0.60	1.41

By looking at the three settings under the same $k = 5$, we found *our approach achieved the highest accuracy*, whereas *OntoSum* performed even worse than *RandomRank*. Considering the other two settings as well, we can see the effect of summary length on the time spent: with summarization enabled (i.e. other than *ReturnsAll*), the time is significantly shorter. This corresponds to our expectation that *participants' efficiency in carrying out tasks can be improved when using concise entity descriptions*. By comparing the results of RELIN under $k = 5$ and $k = 10$, we further found that even when being generated by the same approach, *longer summaries required noticeably more time*.

An interesting finding reflected by the last three rows of Table 2 is that, with longer summary length, although the time increases as expected, the accuracy actually decreases. That is, the *accuracy of judgments does not positively correlate with the amount of presented data*. Many participants reported in post-experiment interviews that it was because they got rather lost when facing a large amount of (often low-quality and confusing) information. This indicates that *providing a concise entity description could also improve the user experience and effectiveness (e.g. accuracy here)*.

6.3 Discussion

Although our approach performed better than the baselines, the results are still far from perfect. For example, in the intrinsic evaluation, the overall levels of agreement between a computed summary and an ideal summary (i.e. quality) are 4.88 and 2.40 at best when $k = 10$ and $k = 5$, respectively, which are much lower than the ones between ideal summaries (7.86 and 2.91, respectively). That means, *automatically computed summaries still cannot replace handcrafted ones*.

The experimental results also revealed some factors that deserve consideration when further improving our approach. Firstly, although some features were ranked high because of their high informativeness and notable relatedness, e.g. features that stand for the longitude and latitude of a city, they were not preferred by most participants because the information they carry were deemed too "domain-specific" to be exploited. That is, these features are highly informative for domain experts that can deal with this particular kind of knowledge, but are not as valuable when presented to average users. This suggests a *user-specific notion of informativeness*, which could be implemented by leveraging user profiles

or feedback. Secondly, *information redundancy* was observed in entity descriptions, which should be reduced during summarization. For example, the location of a city in DBpedia is usually not only described by the properties "longitude" and "latitude" but also redundantly described by an additional "point" property. Although our implementation has partially addressed the issue of diversity (as discussed in Sect. 5), other strategies are still needed to cope with more general cases. Thirdly, as described in Sect. 2, we focus on ranking and selecting features, rather than presenting. However, several participants reported that they could hardly understand some features, whereas some others suggested that they would prefer to see summaries presented using a richer widget, as opposed to simply a list of features as we did in the experiments. Thus, we can conclude that besides selecting the best features, *methods used for presenting entity summaries also have an impact on the user-perceived quality.*

7 Conclusions and Future Work

We have studied the problem of entity summarization, which is related to but different from the problems of extractive text and ontology summarization, since it is more about identifying the entity that underlies a lengthy description. To this novel problem, we have proposed a solution called RELIN. As a variant of the random surfer model, it is based on non-uniform probability distributions, and embeds informativeness into the traditional relatedness-based centrality measure. We have presented an implementation that rests on the information captured by the labels of nodes and edges in the data graph. It goes beyond related methods for ontology summarization which mainly build upon the graph structure. The experimental results of applying our approach to entity summarization are quite promising. It performs better than the baselines in terms of producing summaries that are closer to handcrafted ideal summaries, and that assist users in confirming entity mappings more accurately.

The experimental results and feedback obtained from the participants have indicated directions for future research. We will study "human factors" in the context of entity summarization. For instance, we will look at user feedback. We are also interested in application-specific entity summaries, such as query-biased summaries for entity search.

Acknowledgments. This work was supported in part by the NSFC under grant 61003018 and 61021062, and from the AIFB part, by the German Federal Ministry of Education and Research (BMBF) under the CollabCloud project grant (01IS0937A-E). We would like to thank Dr. Xiang Zhang, Saisai Gong, and all the participants in the experiments.

References

1. Aleman-Meza, B., Halaschek-Wiener, C., Budak Arpinar, I., Ramakrishnan, C., Sheth, A.P.: Ranking Complex Relationships on the Semantic Web. IEEE Internet Comput. 9(3), 37–44 (2005)

2. Bu, S., Lakshmanan, L.V.S., Ng, R.T.: MDL Summarization with Holes. In: 31st International Conference on Very Large Data Bases, pp. 433–444. ACM, New York (2005)
3. Budanitsky, A., Hirst, G.: Evaluating WordNet-based Measures of Lexical Semantic Relatedness. Comput. Linguist. 32(1), 13–47 (2006)
4. Carbonell, J., Goldstein, J.: The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In: 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 335–336. ACM, New York (1998)
5. Cheng, G., Qu, Y.: Searching Linked Objects with Falcons: Approach, Implementation and Evaluation. Int. J. Semant. Web Inf. Syst. 5(3), 49–70 (2009)
6. Delbru, R., Toupikov, N., Catasta, M., Tummarello, G., Decker, S.: Hierarchical Link Analysis for Ranking Web Data. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part II. LNCS, vol. 6089, pp. 225–239. Springer, Heidelberg (2010)
7. Diligenti, M., Gori, M., Maggini, M.: A Unified Probabilistic Framework for Web Page Scoring Systems. IEEE Trans. Knowl. Data Eng. 16(1), 4–16 (2004)
8. Erkan, G., Radev, D.R.: LexRank: Graph-based Centrality as Salience in Text Summarization. J. Artif. Intell. Res. 22, 457–479 (2004)
9. Franz, T., Schultz, A., Sizov, S., Staab, S.: TripleRank: Ranking Semantic Web Data By Tensor Decomposition. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 213–228. Springer, Heidelberg (2009)
10. Haveliwala, T.H.: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. IEEE Trans. Knowl. Data Eng. 15(4), 784–796 (2003)
11. Jeffery, S.R., Franklin, M.J., Halevy, A.Y.: Pay-as-you-go User Feedback for Dataspace Systems. In: 2008 ACM SIGMOD International Conference on Management of Data, pp. 847–860. ACM, New York (2008)
12. Mohammad, S., Hirst, G.: Distributional Measures of Concept-distance: A Task-oriented Evaluation. In: 2006 Conference on Empirical Methods in Natural Language Processing, pp. 35–43. ACL, Sydney (2006)
13. Navlakha, S., Rastogi, R., Shrivastava, N.: Graph Summarization with Bounded Error. In: 2008 ACM SIGMOD International Conference on Management of Data, pp. 419–432. ACM, New York (2008)
14. Nie, Z., Ma, Y., Shi, S., Wen, J.-R., Ma, W.-Y.: Web Object Retrieval. In: 16th International World Wide Web Conference, pp. 81–90. ACM, New York (2007)
15. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford InfoLab (1999)
16. Penin, T., Wang, H., Tran, T., Yu, Y.: Snippet Generation for Semantic Web Search Engines. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) ASWC 2009. LNCS, vol. 5926, pp. 493–507. Springer, Heidelberg (2009)
17. Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-based Summarization of Multiple Documents. Inf. Process. Manag. 40(6), 919–938 (2004)
18. Spärck Jones, K.: Automatic Summarising: The State of the Art. Inf. Process. Manag. 43(6), 1449–1481 (2007)
19. Tran, T., Wang, H., Haase, P.: Hermes: Data Web Search on a Pay-as-you-go Integration Infrastructure. J. Web Semant. 7(3), 189–203 (2009)
20. Zhang, X., Cheng, G., Qu, Y.: Ontology Summarization Based on RDF Sentence Graph. In: 16th International World Wide Web Conference, pp. 707–716. ACM, New York (2007)