Relatedness-based Multi-Entity Summarization

Kalpa Gunaratna[1], **Amir Hossein Yazdavar**[1], **Krishnaprasad Thirunarayan**[1],
Amit Sheth[1], and **Gong Cheng**[2]
[1]Kno.e.sis, Wright State University, Dayton OH, USA
[2]National Key Laboratory for Novel Software Technology, Nanjing University, China
{kalpa,amir,tkprasad,amit}@knoesis.org, gcheng@nju.edu.cn

Abstract

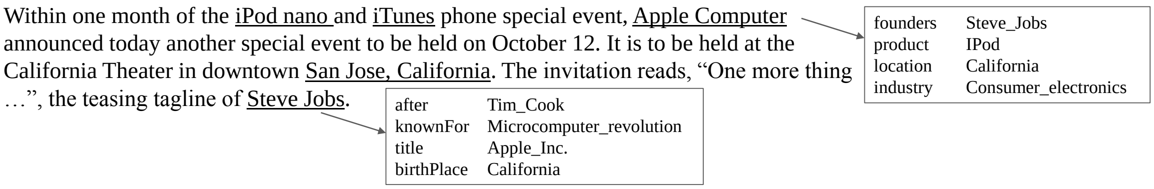
Representing world knowledge in a machine processable format is important as entities and their descriptions have fueled tremendous growth in knowledge-rich information processing platforms, services, and systems. Prominent applications of knowledge graphs include search engines (e.g., Google Search and Microsoft Bing), email clients (e.g., Gmail), and intelligent personal assistants (e.g., Google Now, Amazon Echo, and Apple's Siri). In this paper, we present an approach that can summarize facts about a collection of entities by analyzing their relatedness in preference to summarizing each entity in isolation. Specifically, we generate informative entity summaries by selecting: (i) inter-entity facts that are similar and (ii) intra-entity facts that are important and diverse. We employ a constrained knapsack problem solving approach to efficiently compute entity summaries. We perform both qualitative and quantitative experiments and demonstrate that our approach yields promising results compared to two other stand-alone state-of-the-art entity summarization approaches.

1 Introduction

The task of extracting, storing, and representing entity-related information has recently gained significant attention in academia and industry. The Linking Open Data (LOD) initiative has encouraged researchers to publish open and freely accessible entity-based structured knowledge on the Web. Similar to this, many commercial companies have started developing rich, proprietary knowledge graphs (e.g., Google knowledge graph, Microsoft Satori knowledge graph, and Amazon product graph) to support their products, services, and intelligent user interfaces. Intelligent agents like Amazon Echo, Google Now, and Microsoft Cortana utilize these structured knowledge graphs to provide a rich experience to users in the context of question answering and recommendations. Moreover, researchers have shown that knowledge graphs evolve over time by gaining more high quality knowledge [Auer *et al.*, 2013]. As a consequence, the entity descriptions grow in size and selecting a subset of the description depending on the task at hand, referred to as *Entity Summarization* in the literature [Cheng *et al.*, 2011], is necessary to avoid information overload on the data consumers.

Many flavors of techniques have been applied in creating entity summaries in the recent past. For example, the RELIN [Cheng *et al.*, 2011] and LinkSum [Thalhammer *et al.*, 2016] entity summarization systems have employed PageRank-based ranking mechanisms, the FACES system [Gunaratna *et al.*, 2015] demonstrated an incremental conceptual hierarchical clustering-based approach in creating comprehensive (diverse) summaries, and [Sydow *et al.*, 2013] investigated entity neighborhoods in the graph to generate diverse summaries. Systems such as those mentioned above focus on summarizing individual entities by giving precedence to selecting the most important facts for distinctly identifying an entity. But summarizing a collection of entities by showing related facts (retrieved from a knowledge graph) for quick understanding of the entity collection as a whole compared to individual entities in isolation is an important issue that is yet to be resolved. Such a system can help users to: (i) understand documents when browsing by presenting related facts for entities and (ii) interact with related facts and entities when searching and browsing on the Web (e.g., Google search shows related entity collections). A solution to this problem should maximize the similarity or relatedness of facts selected between the entities as it increases the understandability of the entity collection. For example, Figure 1 shows an example of such a summary creation for "Apple Computer" and "Steve Jobs". For the entity Steve Jobs, it shows more facts about computers than other topics because the majority of the entities are talking about computers or entities related to computers. Further, it shows facts related to the entire entity collection (e.g., selection of "California" for Steve Jobs). In other words, the summary generated for the entity Steve Jobs can vary from document to document depending on the other entities that appear with it. Hence, this kind of a summary is dynamic and context dependent, compared to entity summaries generated by stand-alone entity summarization systems like RELIN and FACES which are context independent and static.

Diversity is an important characteristic that makes entity summaries comprehensive subject to the length constraints. Therefore, we should try to maximize the diversity of the facts selected for each entity summary; otherwise, they may contain redundant facts that make them less in-



Within one month of the iPod nano and iTunes phone special event, Apple Computer announced today another special event to be held on October 12. It is to be held at the California Theater in downtown San Jose, California. The invitation reads, "One more thing …", the teasing tagline of Steve Jobs.

founders	Steve_Jobs
product	IPod
location	California
industry	Consumer_electronics

after	Tim_Cook
knownFor	Microcomputer_revolution
title	Apple_Inc.
birthPlace	California

Figure 1: Showing entity summaries maximizing relatedness between them for a news item from Wikinews corpus.

teresting and useful. We propose **RE**latedness-based **M**ulti-**E**ntity **S**ummarization (REMES) approach that facilitates the above-mentioned characteristics in creating entity summaries for an entity collection. For this purpose, we adapt and map the Quadratic Multidimensional Knapsack Problem (QMKP), which is an extension of the Quadratic Knapsack Problem (QKP) [Pisinger, 2007] and utilize graph-based relatedness and semantic similarity measures. Specifically, we:

1. Generate entity summaries for a collection of entities by: (i) maximizing inter-entity related facts, (ii) maximizing intra-entity importance of facts, and (iii) minimizing intra-entity related facts, by adapting QMKP. We modify a version of the Greedy Randomized Adaptive Search Procedures (GRASP) algorithm to compute entity summaries efficiently.
2. Utilize graph-based and semantics-based relatedness measures to create entity summaries.

We evaluate the proposed approach qualitatively and quantitatively against state-of-the-art entity summarization approaches and show that it generates satisfactory summaries.

The rest of the paper is organized as follows. First, we present related work, describe the problem, necessary notation, and the proposed approach. Next, we discuss the evaluation and results of our approach. Finally, we conclude with future directions.

2 Related Work

Entity summarization work can be divided mainly into two categories in relation to the context of this paper. The first set of approaches generate entity summaries for general purpose use and the others are for various specific tasks. Orthogonal to this breakdown, summaries can be generated considering one entity at a time vs. a collection of entities at once.

Several recent efforts involve generating entity summaries for single entity and for general purpose use. RELIN [Cheng *et al.*, 2011], FACES [Gunaratna *et al.*, 2015; 2016], LinkSum [Thalhammer *et al.*, 2016], SUMMARUM [Thalhammer and Rettinger, 2014], diversity-based summaries [Sydow *et al.*, 2013], and contextual entity summaries by mining query logs [Yan *et al.*, 2016] are good examples. In these approaches, RELIN, SUMMARUM, and LinkSum approaches adapt modified random surfer models (PageRank) to rank facts and then select them for summaries. Further, LinkSum does link analysis to select important features. FACES followed a different approach by using a conceptual clustering algorithm to identify different themes of features belonging to the entity and then ranked them to pick concise and diverse entity summary. [Sydow *et al.*, 2013] also followed a diversity based entity summarization approach but they considered filtering out syntactically similar properties when traversing the graph for the entity as improving diversity. All these systems generate summaries by processing one entity at a time and hence, they are unable to capture related facts that might exist between entities in an entity collection. In contrast, our approach REMES is specifically designed to address this problem.

Entity summaries have shown to be effective in performing specific tasks (supporting human effort) like entity linking [Cheng *et al.*, 2015b] and entity resolution [Cheng *et al.*, 2015a]. Unlike previously mentioned approaches, these consider more than one entity. REMES also considers multiple entities but differs by making general purpose summaries, utilizing graph and semantics-based relatedness measures, and focusing on the diversity of each entity summary. For relatedness measures, we used a graph-based path embedding model [Ristoski and Paulheim, 2016] and lexical database-based semantic similarity computation used in [Gunaratna *et al.*, 2015]. We enforce selecting diverse features for each entity and related features among entities. In addition to modifying the pairwise greedy ranking and profit measures in the GRASP algorithm, we altered the *Restricted Candidate List (RCL)* by using a threshold for each entity's feature set.

3 Problem Description

3.1 Preliminaries

Let E, L, and P be the sets of entities, literals, and properties, respectively, in the knowledge graph G. An entity e ($e \in E$) is described using property-value pairs $(p, v) \in P$ x $(E \cup L)$. A property-value pair is called a *feature* (f) and the collection of features that belong to entity e is called the *feature set* $FS(e)$. Using the above notions, an entity summary for an entity e, $Summ(e)$, of size k is defined as selecting a subset of $FS(e)$ such that $|Summ(e)| \leq k$ and $|FS(e)| > k$, where k is a positive integer [Cheng *et al.*, 2011; Gunaratna *et al.*, 2015].

3.2 Problem Statement and Description

Problem Statement: Given a collection of entities, we select features belonging to these entities maximizing inter-entity relatedness and intra-entity importance, and minimizing intra-entity relatedness of features.

In this problem, we consider generating entity summaries for a collection of entities together by selecting features to show the relatedness among the entities and importance and diversity within entities. That is, for a given entity collection $\{e_1, e_2, ...e_n\} \subseteq E$, and summary length constraints $k_1, k_2, ..k_n$, we want to generate corresponding entity summaries $Summ(e_1), Summ(e_2), ...Summ(e_n)$ that has the maximum score according to the following objectives:

$$\begin{aligned}(Summ(e_1), .., Summ(e_n)) = & \underset{(Se_1 \subseteq FS(e_1), .., Se_n \subseteq FS(e_n))}{argmax} \\ & \Big(\alpha * (\Sigma_{x=1}^{n} \Sigma_{f_i \in Se_x}\ rank(f_i)) \\ & - \beta * (\Sigma_{x=1}^{n} \Sigma_{f_i, f_j \in Se_x}\ r(f_i, f_j)) \\ & + \gamma * (\Sigma_{i=1}^{n} \Sigma_{j=i+1}^{n} \Sigma_{f_i \in Se_i} \Sigma_{f_j \in Se_j}\ r(f_i, f_j))\Big) \\ & where\ |Se_x| \leq k_x,\ k_x \in \mathbb{Z}^+ \end{aligned} \tag{1}$$

The function r and $rank$ compute relatedness and importance scores in the range [0,1], as discussed in Section 4.3. $\alpha, \beta, \gamma \in \mathbb{R}^+$ are the weights (of the objectives) to be tuned. By maximizing the similarity of facts selected to different entity summaries, we try to provide connections between the entities in their summary descriptions for the user to better understand the coherency of the content. We maximize the selection of important features as well as related ones to make good quality summaries. Further, we avoid selecting similar features for an entity (through imposing negative values) to improve diversity and coverage of features given the summary length constraints.

4 Approach

The problem described in Section 3.2 requires maximizing relatedness, importance, and diversity of features (property-value pairs), controlled by the length of each entity summary for the entity collection. First, let's consider selecting features for an entity e from its feature set $FS(e)$. Then we discuss how to extend it to process an entity collection.

4.1 Selecting Features for an Entity

The features $f \in FS(e)$ are numbered from 1 to $|FS(e)|$. First, the important features need to be selected for the summary. For this, we utilize a tf-idf based ranking score for each feature f. Second, the selection of similar features to the summary should be discouraged to improve diversity (and hence improved coverage). To demote the selection of features that are similar to the already selected ones for the summary from the entity, we represent the relatedness between the features with the negation of the similarity value.

By defining a pairwise profit function for the features, we can map this problem as an instance of the QKP [Pisinger, 2007]. QKP is a generalization of the classical 0-1 knapsack problem where it maximizes a quadratic objective function to a linear constraint [Gallo *et al.*, 1980; Yang *et al.*, 2013]. We define the profit p_{f_i, f_j} for selecting the feature pair f_i and f_j for the summary $Summ(e)$ as in Equation 2, where α, $\beta \in \mathbb{R}^+$. The function $rank(f_i)$ calculates the importance of the feature f_i and the function $r(f_i, f_j)$ computes relatedness of the two features f_i and f_j. The intuition behind giving a negative value for the relatedness score when the two features belong to the same entity is to make their overall profit lower if they are more related to each other. This discourages selection of new features which are more related to the already selected features for the entity summary (increases diversity).

$$p_{f_i, f_j} = \begin{cases} \alpha * rank(f_i), & \text{if } i\ =\ j \\ -\beta * r(f_i, f_j), & \text{if } i\ \neq\ j \end{cases} \tag{2}$$

By introducing a series of binary variables x_a for a = 1, 2, ..., $|FS(e)|$ that indicate whether or not the feature f_a is selected to the optimal summary, the selection of $Summ(e)$ maximizing the objectives outlined above can be defined as follows in terms of QKP formulation. $w(f_a)$ is the weight of the feature of f_a and p_{f_a, f_b} defines the profit for the two features f_a and f_b.

$$\begin{aligned} maximize\ & \Sigma_{a=1}^{|FS(e)|} \Sigma_{b=a}^{|FS(e)|}\ p_{f_a, f_b} * x_a * x_b \\ where,\ & \Sigma_{a=1}^{|FS(e)|} w(f_a) * x_a \leq k,\ x_a \in \{0, 1\} \end{aligned} \tag{3}$$

In the QKP, the algorithm optimizes selecting items that maximizes profit computed between items. In other words, it can be used to select features to the entity summary to get maximum profit by analyzing pairwise profit of the selected features. When using both positive and negative weights as shown in Equation 2, QKP is NP-Hard, that is, it does not have a polynomial-time algorithm to generate solutions unless P = NP [Pisinger, 2007]. Therefore, an approximation algorithm like GRASP can be used to compute a solution.

4.2 Selecting Features for Multiple Entities

The mapping of QKP above refers to creating entity summaries for individual entities. An extension of this to handle multiple entities with the addition of maximizing inter-entity relatedness features is what we require in our problem. To achieve this objective, we consider mapping this problem to an instance of QKP with multiple constraints, namely Quadratic Multidimensional Knapsack Problem (QMKP). Given a collection of entities $e_1, e_2, .., e_n$, features numbered $f_{i,1}$ to $f_{i,|FS(e_i)|} \in FS(e_i)$, and having random variables $x_{i,a}$ for $i = 1, 2, ..., n$ and $a = 1, 2, ..., |FS(e_i)|$ to denote whether the features $f_{i,a}$ is selected for the best possible summary, the optimization goals can be formalized as follows.

$$\begin{aligned} maximize\ & \Sigma_{i=1}^{n} \Sigma_{j=i}^{n} \Sigma_{a=1}^{|FS(e_i)|} \Sigma_{b=1}^{|FS(e_j)|}\ p_{f_{i,a}, f_{j,b}} * x_{i,a} * x_{j,b} \\ where,\ & \Sigma_{a=1}^{|FS(e_i)|} w(f_{i,a}) * x_{i,a} \leq k_i,\ x_{i,a} \in \{0, 1\} \end{aligned} \tag{4}$$

k_i is the capacity of knapsack belonging to entity e_i. $w(f_{i,a})$ is the weight of the a^{th} feature of e_i and $p_{f_{i,a}, f_{j,b}}$ is the profit for the two features $f_{i,a}$ and $f_{j,b}$. Note that we have n constraints to satisfy (a knapsack for each entity).

In extending QKP, we adapted a memory-based GRASP [Yang *et al.*, 2013] approach, to simply run with multiple constraints. The algorithm runs through several iterations, and in each iteration, it generates a random solution first based on a greedy ranking function and sampling from the candidate item set. The original GRASP algorithm proposes a greedy ranking function [Yang *et al.*, 2013] and we modify it to bias the selection of features to also consider future candidate selection. Given the already selected feature set S and candidate feature f, the modified greedy ranking function $Gr(S, f)$ in the construction phase of the GRASP algorithm is as shown in Equation 5. The function w gives weight of each feature and $\tau, \phi \in$ [0,1]. The component related to τ considers the current feature against already selected items and the component related to ϕ makes the algorithm to consider unselected items in scoring the current feature, making the initial selection of features in the algorithm less random.

$$Gr(S, f) = \frac{\Sigma_{i\in S}\Sigma_{j\in S, j\leq i}\ p_{i,j} + \tau\Sigma_{x\in S}\ p_{x,f} + \phi\Sigma_{x\notin S, x\neq j}\ p_{x,f} + p_{f,f}}{\Sigma_{y\in S\cup\{f\}} w(y)} \tag{5}$$

Then, in the local search phase, the memory-based GRASP algorithm tries to improve the solution by further maximizing the total profit by swapping selected items with items from the unselected item list. The total profit of the selected items in the summary is calculated by $\Sigma_{i\in S}\Sigma_{j\in S, j\leq i}\ p_{i,j}$.

Since we have more than one entity to consider in the optimization approach, the profit computation is updated to reflect this need as shown in Equation 6 below. In the equation, $\alpha, \beta, \gamma > 0$ and chosen empirically (tuned). The diagonal of the profit matrix contains the ranking scores (signifying the importance of each feature) and non-diagonal entries contain pairwise relatedness of features. We make profits negative for feature pairs belonging to the same entity so that highly similar feature pairs will not be selected for the same entity.

$$p_{f_{i,a}, f_{j,b}} = \begin{cases} \alpha * rank(f_{i,a}), & \text{if } i = j \ and \ a = b \\ -\beta * r(f_{i,a}, f_{j,b}), & \text{if } i = j \ and \ a \neq b \\ \gamma * r(f_{i,a}, f_{j,b}), & \text{if } i \neq j \end{cases} \tag{6}$$

4.3 Importance, Relatedness and Diversity

Note that we want diverse features to be selected in each entity summary and related features among entities. Further, we do not want arbitrary features to be selected for the summaries but be influenced by their importance. We try to combine these characteristics as shown in Equation 6.

Importance of a feature

The diagonal of the profit matrix has the importance score for each feature f calculated by $rank(f)$ as shown in Equation 9. We rank features based on how informative the property-value pairs are and how popular the values are [Cheng *et al.*, 2011; Gunaratna *et al.*, 2015]. We try to achieve a trade-off between the two measures similar to tf-idf score in Information Retrieval. $Inf(f)$ computes the inverse logarithmic feature frequency as shown in Equation 7 where N is the total number of entities in the knowledge graph G. The popularity (frequency) of value v of the feature f is computed by Equation 8. $Prop(f)$ and $Val(f)$ are two functions that return the property and value of the feature f. Function $rank(f)$ facilitates selection of important features in the GRASP based summary generation as it can add higher profits for some features which are considered to be important in addition to the pairwise feature profit computed based on relatedness.

$$Inf(f) = log(\frac{N}{|\{e|f \in FS(e)\}|}) \tag{7}$$

$$\begin{aligned} Po(v) = log|\{triple\ t|\exists\ e, f : t \text{ ``}appears\ in\text{''}\ G \\ and\ t \equiv (e\ Prop(f)\ Val(f))\ and\ Val(f) = v\}| \end{aligned} \tag{8}$$

$$rank(f) = Inf(f)\ *\ Po(Val(f)) \tag{9}$$

Relatedness of a feature pair

We calculate the relatedness of a feature pair by utilizing two measures. First, we employ semantics based measurement to analyze the relatedness between two properties by computing the overlap of terms that represent the two properties. Second, we utilize a graph and co-occurrence based measure to compute relatedness between two values (entities), specifically a vector space model similar to word embedding for graphs.

For the *semantics based relatedness measure*, we process the property of each feature, with the help of a lexical database, namely WordNet [1]. For a given feature f, we get its property name (label of the property URL) and retrieve hypernyms from the lexical database. We also pre-process them (e.g., remove camel-case and stop words). Then we combine all the extracted terms and original terms for property label of the feature f into a Set S_f. Then the semantics based relatedness $SemRel_p(f_i, f_j)$ of the two features f_i, f_j is computed by getting the jaccard co-efficient of the two sets of the features S_{f_i} and S_{f_j} as shown in Equation 10. We chose to get hypernyms from the lexical database instead of synonyms or hyponyms because we need to compute the relatedness instead of strong similarity.

$$SemRel_p(f_i, f_j) = \frac{|S_{f_i} \cap S_{f_j}|}{|S_{f_i} \cup S_{f_j}|} \tag{10}$$

We consider a *co-occurrence based relatedness* to be computed between values of the features. Similar to word embedding models like Word2Vec, we utilize a graph based model called RDF2Vec [Ristoski and Paulheim, 2016] for this purpose. The model was developed using path based co-occurrence and showed promising results in data mining and similarity computation applications [Ristoski and Paulheim, 2016]. We employ a pre-trained model on DBpedia knowledge graph and compute cosine similarity of any given two entities over their vector representation as shown in Equation 11. Given two features $f_{i,a}$ and $f_{j,b}$ belonging to entities e_i and e_j and the corresponding vector representation of their values ($Val(f_{i,a})$ and $Val(f_{j,b})$) shown as $\vec{Val}(f_{i,a})$ and $\vec{Val}(f_{j,b})$, respectively, and the relatedness measure $r(f_{i,a}, f_{j,b})$ is defined as in Equation 12.

$$GraphRel_v(\vec{Val}(f_{i,a}), \vec{Val}(f_{j,b})) = \frac{\vec{Val}f_{i,a} \cdot \vec{Val}f_{j,b}}{|\vec{Val}f_{i,a}||\vec{Val}f_{j,b}|} \tag{11}$$

[1] https://wordnet.princeton.edu/

$$r(f_{i,a}, f_{j,b}) = \frac{SemRel_p(f_{i,a}, f_{j,b}) + GraphRel_v(\vec{Val}(f_{i,a}), \vec{Val}(f_{j,b}))}{2} \tag{12}$$

Improving feature diversity in entities

We implemented the GRASP algorithm presented by [Yang *et al.*, 2013] and adapted it to fit our problem solution. We do not provide a detailed description of the algorithm due to limited space. The GRASP approach constructs random solutions and then improves them in the local search phase, in several iterations and returns the best result found so far based on the total profit. Recall that one of our objectives in this problem is to improve diversity of features selected for each entity. In order to achieve that, in addition to the introduction of negative profits, we make changes in the candidate feature selection step. The GRASP approach keeps a *candidate set* and *remaining set* of features for the collection of entities. The remaining set contains all the unselected features from the entity collection and candidate set is a random sample of this set. By introducing a threshold value η, we filter out features belonging to the remaining set where their maximum pairwise profit value with any already selected feature is greater than η. That is, for a candidate feature f and the set of selected features S, we filter out f if $max(p_{f,f_{i\in S}}) > \eta$. With this modification, we are able to introduce better diversity in the results for each entity by forcing the combinatorial optimization algorithm to not have access to similar features that have already been selected.

5 Evaluation and Results Discussion

We discuss details of our experiments and results below.

5.1 Implementation Details

In our implementation of memory-based GRASP algorithm, we set $\gamma, \beta, \lambda, \sigma$ to 1, 3, 5, and 5, respectively (as suggested by authors of GRASP). We normalized profit values by dividing them using the maximum profit. We also added average similarity between the value of each feature and the entity collection to the diagonal of the profit matrix (to improve relatedness). In the greedy ranking function shown in Equation 5, we set $\tau = 1$ and $\phi = 0.5$. In the profit matrix, we used $\alpha = 2$, $\beta = 1$, and $\gamma = 1.5$. We set the threshold $\eta = 0.45$. The parameter values in the greedy ranking function and profit computation needed to be tuned for this task (we used a separate document sample). Further, in this implementation, we consider feature weights to be uniform and equal to 1. Therefore, the length of the summary for each entity denotes the knapsack size for that entity. We used DBpedia (version 2016-04) encyclopedic dataset as our knowledge graph to retrieve entity descriptions and ran the RDF2Vec model on it. For the semantic relatedness measure, we used the WordNet lexical database.

5.2 Datasets and Evaluation

We evaluated our system using qualitative and quantitative measures. For the qualitative evaluation, we requested a set of judges to rank systems on the Likert scale [2] 1 to 5 (1 for strongly disagree and 5 for strongly agree) for a given set of questions. For the quantitative evaluation, we evaluated REMES against other systems for their level of relatedness. We used two document samples taken from two popular entity linking benchmark datasets: (i) Wikinews [3] (20 documents) and (ii) AQUAINT [4] (10 documents).

Qualitative evaluation

We compared REMES with two state-of-the-art stand-alone entity summarization systems: FACES and RELIN. The goal of this evaluation is to measure how successful is each system in selecting summaries for each entity in a collection of entities to maximize inter-entity relatedness and intra-entity diversity and importance of features. We constructed 5 questions to evaluate on a Likert scale (1 strongly disagree and 5 strongly agree). We asked 13 judges to answer these questions for each dataset and each question had at least 5 different judges. The evaluation contains 850 question instances scored by the judges. The questions and the results are shown in Table 1. REMES achieved higher mean scores for all the questions used in the evaluation on the Likert scale. We measured its statistical significance by first performing one-way ANOVA and then using Least Significant Difference (LSD) post-hoc analysis.

Quantitative evaluation

To further evaluate the robustness of our model, we processed the summaries generated by the three systems and compared how effective they were in picking related features between entities. To measure the relatedness between features in the generated summaries, we measured semantic similarity of the entities (by processing their labels in the graph) in those features. In particular, we assessed the relatedness of these entities by employing two state-of-the-art NLP semantic similarity techniques, namely, UCI [Newman *et al.*, 2010] and UMass [Mimno *et al.*, 2011]. UCI was measured by a sliding window and the Point-wise Mutual Information (PMI) of all entity pairs. The entity co-occurrence counts were calculated utilizing a sliding window with the size 10. For every value pair the PMI is calculated on Wikipedia articles as shown in Equation 13.

$$UCI(W_i, W_j) = log\frac{p(W_i, W_j) + \epsilon}{p(W_i)p(W_j)} \tag{13}$$

where W_i, W_j are the labels of the entities e_i, e_j and the word probabilities ($p(W)$) are calculated by counting word co-occurrence in a sliding window over Wikipedia. On the other hand, UMass is measured based on document co-occurrence counts as shown in Equation 14.

$$UMass(W_i, W_j) = log\frac{D(W_i, W_j) + \epsilon}{D(W_i)} \tag{14}$$

where $D(W_i, W_j)$ counts the number of documents containing both W_i and W_j words and $D(W_i)$ counts the ones containing W_i, and ϵ is the smoothing factor. We used Palmetto[5] for measuring the UMass and UCI measures (using

[2] https://en.wikipedia.org/wiki/Likert_scale

[3] http://www.newsreader-project.eu/results/data/wikinews

[4] http://www.nzdl.org/wikification/docs.html

[5] http://aksw.org/Projects/Palmetto.html

Question	Wikinews					AQUAINT				
	Response: Mean (SD)			F(2,357)	LSD post-hoc	Response: Mean (SD)			F(2,147)	LSD post-hoc
	REMES	FACES	RELIN	(p-value)	(p <0.05)	REMES	FACES	RELIN	(p-value)	(p <0.05)
Q1: Summaries assisted me to get some relationships between the entities in the entity collection.	3.98 (1.16)	3.66 (1.08)	2.78 (1.18)	35.798 (6.772e-15)	REMES >FACES >RELIN	4.50 (0.65)	3.92 (0.92)	3.06 (1.13)	30.866 (6.427e-12)	REMES >FACES >RELIN
Q2: The facts in each summary are diverse.	4.12 (0.93)	3.93 (1.08)	3.79 (1.28)	2.747 (6.500e-2)	REMES >RELIN	4.26 (1.01)	3.98 (0.89)	3.22 (1.36)	11.879 (1.700e-5)	REMES, FACES >RELIN
Q3: The summaries helped me to better understand the document.	3.69 (0.71)	3.38 (1.41)	2.84 (0.71)	17.868 (4.022e-8)	REMES >FACES >RELIN	4.36 (0.60)	3.76 (0.96)	2.92 (1.32)	25.927 (2.267e-10)	REMES >FACES >RELIN
Q4: The summaries provide me an overview of the entire entity collection.	3.78 (1.07)	3.48 (1.07)	2.91 (1.20)	19.148 (1.260e-8)	REMES >FACES >RELIN	4.26 (0.63)	3.74 (0.80)	2.88 (1.19)	30.123 (1.086e-11)	REMES >FACES >RELIN
Q5: I like the summaries generated.	4.05 (0.89)	3.72 (0.91)	3.18 (1.20)	22.586 (5.805e-10)	REMES >FACES >RELIN	4.22 (0.68)	3.32 (1.04)	2.54 (1.20)	35.611 (2.447e-13)	REMES >FACES >RELIN

Table 1: Evaluating system summaries using questionnaire.

System	UCI		UMASS	
	Wikinews	AQUAINT	Wikinews	AQUAINT
REMES	**0.064**	**0.056**	**-0.301**	**-0.257**
FACES	-0.083	-0.259	-0.971	-0.428
RELIN	-0.221	-0.148	-0.984	-0.589

Table 2: Average coherency of different models.

Summaries generated by REMES	Summaries generated by FACES
Dmitry Medvedev **dbo:title-dbr:President_of_Russia** dbo:otherParty-dbr:Communist_Party_of_the_Soviet_Union dbo:birthPlace-dbr:Saint_Petersburg dbo:predecessor-dbr:Valadimir_Putin	*Dmitry Medvedev* **dbo:title-dbr:President_of_Russia** dbo:otherParty-dbr:Independent_(politician) dbo:almaMater-dbr:Saint_Petersburg_State_University dbo:deputy-dbr:Igor_Shuvalov
Russia dbo:establishedEvent-dbr:Russian_Empire **dbo:leaderName-dbr:Dmitry_Medvedev** dbo:currency-dbr:Russian_ruble dbo:capitol-dbr:Moscow	*Russia* dbo:establishedEvent-dbr:Russian_Empire dbo:leaderName-dbr:Vladimir_Putin dbo:southwest-dbr:Black_Sea dbo:capitol-dbr:Moscow

Figure 2: Example entity summaries for two entities.

Wikipedia as the external corpus). Table 2 shows the semantic relatedness of the generated summaries for the three different systems based on above metrics.

5.3 Discussion

In the qualitative evaluation, REMES ranked higher than the other two systems for both the datasets, except for question 2, where p-value (0.18 for Wikinews and 0.20 for AQUAINT) was not significant enough to make a decision between REMES and FACES. This is not totally unexpected because FACES has shown superior capabilities in achieving high quality diversity in generating entity summaries (by using a comprehensive hierarchical clustering approach). In all other questions, REMES outperforms the others and achieved higher mean scores, confirming its ability to generate summaries while maximizing inter-entity relatedness and intra-entity importance (and comparable to FACES in diversity). Figure 2 shows summaries generated for two entities using the REMES and FACES systems. While REMES tried to make a connection between the entities (by selecting the leader for Russia), FACES could not get such relatedness. This is mainly because FACES cannot and do not consider other entities in the entity collection.

In the quantitative evaluation, we further confirmed that our approach generates summaries that maximizes relatedness of features for entity collections. We utilized an external knowledge source (Wikipedia) to capture relatedness of facts selected for the summaries. The higher the semantic similarity score, the more related facts are in the summaries generated for the entity groups. Clearly, the summaries generated by REMES are more related according to both the measures and they further confirm the achievement of our objective of creating relatedness based entity summaries for entity collections. Further, we intend to investigate more on what properties to select and similarities and alignments among the properties and entities [Gunaratna *et al.*, 2013; 2014] in creating high quality summaries.

6 Conclusion

Summarizing a collection of entities is challenging since it involves processing all the entities in the collection simultaneously. In this paper, we proposed an approach called REMES to select related features among entities while keeping the diversity and saliency of features within each entity. The approach utilizes a graph-based RDF2Vec model to compute relatedness of two entities and semantic expansion based measure to compute relatedness of two properties. Further, we adapted a QMKP problem instance with implementation of a memory-based optimization algorithm called GRASP. The proposed approach is evaluated against two state-of-the-art stand-alone entity summarization systems in two different settings: qualitative and quantitative. Extensive set of experiments using statistical tests (one-way ANOVA and LSD post-hoc) on two different datasets, confirmed our model outperformed the others in generating high quality summaries for collections of entities.

In future, we plan to investigate on improving summary quality by analyzing diversity and relatedness and use REMES in real world applications like facilitating Web search and social media text and Web document understanding. The proposed approach can also be improved by deciding what features to select (importance vs. relatedness) based on the neighboring entities.

Acknowledgments

Research reported in this publication was supported in part by NIMH of the National Institutes of Health (NIH) under award number R01MH105384-01A1. Gong Cheng was supported in part by the NSFC under Grant 61572247. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH or NSFC.

References

[Auer *et al.*, 2013] Sören Auer, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, and Amrapali Zaveri. Introduction to linked data and its lifecycle on the web. In *Reasoning Web. Semantic Technologies for Intelligent Data Access*, pages 1–90. Springer, 2013.

[Cheng *et al.*, 2011] Gong Cheng, Thanh Tran, and Yuzhong Qu. Relin: relatedness and informativeness-based centrality for entity summarization. In *The Semantic Web–ISWC 2011*, pages 114–129. Springer, 2011.

[Cheng *et al.*, 2015a] Gong Cheng, Danyun Xu, and Yuzhong Qu. C3d+ p: A summarization method for interactive entity resolution. *Web Semantics: Science, Services and Agents on the World Wide Web*, 35:203–213, 2015.

[Cheng *et al.*, 2015b] Gong Cheng, Danyun Xu, and Yuzhong Qu. Summarizing entity descriptions for effective and efficient human-centered entity linking. In *Proceedings of the 24th International Conference on World Wide Web*, pages 184–194. ACM, 2015.

[Gallo *et al.*, 1980] Giorgio Gallo, Peter L Hammer, and Bruno Simeone. Quadratic knapsack problems. In *Combinatorial optimization*, pages 132–149. Springer, 1980.

[Gunaratna *et al.*, 2013] Kalpa Gunaratna, Krishnaprasad Thirunarayan, Prateek Jain, Amit Sheth, and Sanjaya Wijeratne. A statistical and schema independent approach to identify equivalent properties on linked data. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 33–40. ACM, 2013.

[Gunaratna *et al.*, 2014] Kalpa Gunaratna, Sarasi Lalithsena, and Amit Sheth. Alignment and dataset identification of linked data in semantic web. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(2):139–151, 2014.

[Gunaratna *et al.*, 2015] Kalpa Gunaratna, Krishnaprasad Thirunarayan, and Amit Sheth. Faces: diversity-aware entity summarization using incremental hierarchical conceptual clustering. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 116–122. AAAI Press, 2015.

[Gunaratna *et al.*, 2016] Kalpa Gunaratna, Krishnaprasad Thirunarayan, Amit Sheth, and Gong Cheng. Gleaning types for literals in rdf triples with application to entity summarization. In *Extended Semantic Web Conference*, pages 85–100. Springer, 2016.

[Mimno *et al.*, 2011] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics, 2011.

[Newman *et al.*, 2010] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. Association for Computational Linguistics, 2010.

[Pisinger, 2007] David Pisinger. The quadratic knapsack problema survey. *Discrete applied mathematics*, 155(5):623–648, 2007.

[Ristoski and Paulheim, 2016] Petar Ristoski and Heiko Paulheim. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*, pages 498–514. Springer, 2016.

[Sydow *et al.*, 2013] Marcin Sydow, Mariusz Pikuła, and Ralf Schenkel. The notion of diversity in graphical entity summarisation on semantic knowledge graphs. *Journal of Intelligent Information Systems*, 41(2):109–149, 2013.

[Thalhammer and Rettinger, 2014] Andreas Thalhammer and Achim Rettinger. Browsing dbpedia entities with summaries. In *The Semantic Web: ESWC 2014 Satellite Events*, pages 511–515. Springer, 2014.

[Thalhammer *et al.*, 2016] Andreas Thalhammer, Nelia Lasierra, and Achim Rettinger. Linksum: using link analysis to summarize entity data. In *International Conference on Web Engineering*, pages 244–261. Springer, 2016.

[Yan *et al.*, 2016] Jihong Yan, Yanhua Wang, Ming Gao, and Aoying Zhou. Context-aware entity summarization. In *International Conference on Web-Age Information Management*, pages 517–529. Springer, 2016.

[Yang *et al.*, 2013] Zhen Yang, Guoqing Wang, and Feng Chu. An effective grasp and tabu search for the 0–1 quadratic knapsack problem. *Computers & Operations Research*, 40(5):1176–1185, 2013.