1-2015

FACES: Diversity-Aware Entity Summarization using Incremental Hierarchical Conceptual Clustering

Kalpa Gunaratna
Wright State University - Main Campus, gunaratna.2@wright.edu

Krishnaprasad Thirunarayan
Wright State University - Main Campus, t.k.prasad@wright.edu

Amit P. Sheth
Wright State University - Main Campus, amit@sc.edu

FACES: Diversity-Aware Entity Summarization using Incremental Hierarchical Conceptual Clustering

Kalpa Gunaratna, Krishnaprasad Thirunarayan and **Amit Sheth**
Kno.e.sis, Wright State University, Dayton OH, USA
{kalpa, tkprasad, amit}@knoesis.org

Abstract

Semantic Web documents that encode facts about entities on the Web have been growing rapidly in size and evolving over time. Creating summaries on lengthy Semantic Web documents for quick identification of the corresponding entity has been of great contemporary interest. In this paper, we explore automatic summarization techniques that characterize and enable identification of an entity and create summaries that are human friendly. Specifically, we highlight the importance of diversified (faceted) summaries by combining three dimensions: diversity, uniqueness, and popularity. Our novel diversity-aware entity summarization approach mimics human conceptual clustering techniques to group facts, and picks representative facts from each group to form concise (i.e., short) and comprehensive (i.e., improved coverage through diversity) summaries. We evaluate our approach against the state-of-the-art techniques and show that our work improves both the quality and the efficiency of entity summarization.

Introduction

Linking Open Data (LOD) initiative encouraged data publishers to put data on the web and link them to other related datasets. The LOD bubble that originated few years ago with a limited number of datasets has now grown into a very large data space. The entity descriptions in these datasets evolve over time (Auer et al. 2013) and grow in length. For example, DBpedia is a very large and central dataset in the LOD cloud extracted from Wikipedia. The English version 3.9 of DBpedia has 4 million entities described in over 800 million RDF triples (facts), averaging about 200 triples per entity. This amount of information is too much for the quick identification of an entity for an user. Therefore, selecting a small subset of the original triples associated with an entity as a summary is necessary for quick/convenient identification/access of entity-related information. This problem has been called *Entity Summarization* (Cheng, Tran, and Qu 2011) in the literature.

Document summarization has been a topic of interest for data mining and information retrieval communities for a long time (Nenkova and McKeown 2012; Mani 2001). Unsupervised summarization techniques for text have been extensively used in the recent past due to their flexibility to adapt to the nature of datasets and ability to create dynamic summaries (Wang and Li 2010). Document summarization is different from entity summarization because documents are unstructured and contain frequent words that can be exploited for summarization. In contrast, RDF entity descriptions are structured and do not have frequent word appearances. (Cheng, Tran, and Qu 2011) proposed an algorithm to create RDF entity summaries considering these differences and taking insights from hypertext document ranking on the Web. It is based on PageRank centrality measure and utilizes relatedness and uniqueness of features (i.e., property-value pairs). Furthermore, they showed that entity summaries can be used for quick identification of the original entity.

We hypothesize that centrality measures (including popularity) and ranking mechanisms alone are not sufficient to improve the quality of entity summaries. Rather the added use of orthogonal semantic groups of facts to diversify the summaries can be more effective. To investigate our hypothesis, we propose the **FAC**eted **E**ntity **S**ummarization (FACES) approach. Our contributions are two fold:

1. We identify conceptually similar groups of facts of an RDF entity by adapting and modifying an incremental hierarchical conceptual clustering algorithm called Cobweb (Fisher 1987) and introduce an algorithm to rank facts within a group.
2. We combine three dimensions: diversity, popularity, and uniqueness, to create human friendly entity summaries in time efficient manner.

Moreover, FACES has the following distinct characteristics compared to other entity summarization tools: (1) It selects facts considering diversity which eliminates redundancy (by filtering similar features). (2) It is dynamic as it is not affected by the order of input facts and is robust with regards to evolving facts (thus applicable in streaming contexts). (3) It is relatively fast due to its hierarchical and incremental processing structure. FACES groups conceptually similar facts in order to select the highest ranked feature (based on uniqueness and popularity) from each group to form a faceted (diversified) entity summary. We show that FACES outperforms the state-of-the-art approaches using a manually created gold standard.

The structure of the paper is as follows. In the next sections, we discuss related work, define the problem, and present the FACES approach. Then we explain the evaluation of the system and discuss the results. Finally, we conclude with suggestions for future research.

Related Work

Summarization tasks can be categorized into *extractive* and *non-extractive* methods. Extractive summaries choose a subset of the features for the summary whereas non-extractive summaries include reformulation of the extracted facts. We focus on extractive summaries at the entity level.

(Cheng, Tran, and Qu 2011) introduced and defined the problem of entity summarization in RDF graphs and showed its usefulness in quick identification of an entity. Their system, called RELIN, generalizes the PageRank algorithm to select both related and informative features. The problem with RELIN's random surfer, which selects both related and informative features, is that it tends to emphasize central themes and similar features of an entity because of the centrality based ranking mechanism of PageRank. SUMMARUM (Thalhammer and Rettinger 2014) is an entity summarization system for DBpedia that is also based on PageRank and utilizes the global popularity of resources gleaned with the help of information from the corresponding Wikipedia pages. Note that, neither RELIN nor SUMMARUM focus on diversity in the summary whereas our approach does. (Thalhammer, Knuth, and Sack 2012) proposed an approach that utilizes usage data for creating entity summaries and evaluated it in the movie domain where they could find user ratings for movies. Such an approach is hard to generalize because usage data may not be readily available for entities. We do not utilize any usage data. (Xu, Cheng, and Qu 2014) created entity summaries to facilitate coreference resolution and consider pairs of entities when creating summaries. Hence, it is different from our approach, RELIN, and SUMMARUM.

Ranking in the Semantic Web is closely related to entity summarization as the latter task can be perceived as selecting the top k features from an RDF graph. Various ranking algorithms exist in the literature for RDF graphs including TripleRank (Franz et al. 2009) that ranks triples, SemRank (Anyanwu, Maduko, and Sheth 2005) that ranks associations, (Ding et al. 2005) that ranks documents, and TRank (Tonon et al. 2013) that ranks concepts. These approaches incorporate ranking algorithms for different reasons. For example, TripleRank (Franz et al. 2009) groups triples using tensors and link analysis. TripleRank's goal is to rank and identify authoritative sources for a given entity. The facet or latent grouping concept that we introduce in FACES is different from facets in TripleRank as TripleRank's grouping is based on authority whereas FACES's is based on semantic overlap of the expanded terms of the features. Also, (Cheng, Tran, and Qu 2011) pointed out that it is hard to align the TripleRank approach to the entity summarization problem but its authoritative ranking is similar to RELIN where centrality dominates. Grouping in RDF datasets has been investigated in upper level ontology creation (Zhao and Ichise 2012) and property alignment (Gunaratna et al. 2013), but these groupings are different from what we explored in FACES. We find *conceptually similar* groups (explained later) that are not just related (i.e., object value overlap of RDF triples) (Zhao and Ichise 2012) or equivalent (Gunaratna et al. 2013) groups.

Diversity has been shown to be useful in creating graphical entity summarization (Sydow, Pikuła, and Schenkel 2013), which is different from entity summarization as it produces a graph (includes neighboring entities) rather than a set of features. They pick 'lexicographically different' property names to achieve diversity in the summary using syntactic measures (e.g., *birthPlace* and *deathPlace* are different in their context) whereas FACES groups them together (i.e., significantly beyond string similarity). FACES differs from the existing entity summarization and ranking systems to create both concise and comprehensive summaries. We hypothesize that diversity makes the summary provide a more complete picture of the entity (i.e., comprehensive) when subjected to a length constraint (i.e., concise) environment.

Problem description

Informal problem statement: An entity is usually described using conceptually diverse set of facts to improve coverage. We want to select a *'representative'* subset of this set in a good summary to uniquely identify the entity.

We review below preliminaries introduced in (Cheng, Tran, and Qu 2011) for completeness.

Preliminaries

A data graph is a graph based data model, which describes entities using properties and their values. It consists of sets of entities (E), literals (L), and properties (P). An entity e ($e \in E$) is described in a data graph using property-value pairs $(a, v) \in P$ x $(E \cup L)$.

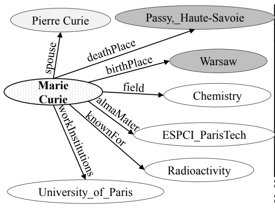
Definition 1 ***(data graph)*** *A data graph is a digraph G = $\langle V, A, Lbl_V, Lbl_A\rangle$, where (i) V is a finite set of nodes, (ii) A is a finite set of directed edges where each $a \in A$ has a source node Src(a) $\in$ V, a target node Tgt(a) $\in$ V, (iii) $Lbl_V : V \mapsto E \cup L$ and (iv) $Lbl_A : A \mapsto P$. Lbl_V and Lbl_A are labeling functions that map nodes to entities or literals and edges to properties, respectively.*

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

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

An entity summary is a subset of all features that belong to that entity.

Definition 4 ***(entity summary)*** *Given an entity e and a positive integer $k < |FS(e)|$, summary of entity e is Summ(e, k) $\subset$ FS(e) such that $|Summ(e, k)| = k$.*



Entity - Marie Curie				
Feature Set	**Facets**	**Features**	**Property**	**Value**
FS	F1	f1	spouse	Pierre_Curie
	F2	f2	birthPlace	Warsaw
		f3	deathPlace	Passy,_Haute-Savoie
	F3	f4	almaMater	ESPCI_ParisTech
		f5	workInstitutions	University_of_Paris
		f6	knownFor	Radioactivity
		f7	field	Chemistry

Summary 1 (normal) : {f2, f3, f4}
Summary 2 (faceted) : {f1, f2, f6}

Figure 1: Facets of entity - Marie Curie. Values for conceptually similar features are in the same color pattern.

Consider the entity Marie Curie taken from DBpedia as shown in Figure 1. We omit namespaces of the URIs for the clarity of presentation. Then a summary for the entity Marie Curie can be created selecting a subset of the features. E.g., Summary 1 and Summary 2 are two summaries of length 3 that can be created for the entity Marie Curie.

Faceted entity summaries

In general, different properties represent different aspects of an entity. E.g., Summary 2 in Figure 1 has two features having *knownFor* and *spouse* properties where they represent different aspects of the entity. The first represents an intangible value and the second represents a human; one talks about the entity's professional background and the other about its personal life. Based on this observation, we can formalize facets of an entity in terms of its feature set.

Definition 5 ***(facet)*** *Given an entity e, a set of facets $F(e)$ of e is a partition of the feature set $FS(e)$. That is, $F(e)$ = $\{C_1, C_2, ...C_n\}$ where $C_i \subseteq FS(e)$. Each C_i is called a facet of e.*

Our hypothesis is that the feature set of an entity can be divided into conceptually orthogonal groups, approximated by the facets, using a partitioning (clustering) algorithm (we will explain the details in the next section). Furthermore, a facet can be viewed as a *hidden variable*. See Figure 1. {F1, F2, F3} is a partition of the feature set FS. Note that the features within each facet are similar than those between facets. I.e., the features that are expressed through *KnownFor* and *Field* properties are *conceptually similar* because they both talk about the entity's professional life. But features having *birthPlace* and *knownFor* properties are conceptually dissimilar as they represent completely different information to each other. Next, we define *Faceted Entity Summary* for an entity based on facets.

Definition 6 ***(faceted entity summary)*** *Given an entity e and a positive integer $k < |FS(e)|$*, faceted entity summary of *e, FSumm(e, k), is a collection of features such that FSumm(e, k) $\subset$ FS(e), $|$FSumm(e, k)$|$ = k, and if $k > |F(e)|$ then $\forall X \in$ F(e) $\Rightarrow X \cap$ FSumm(e, k) $\neq \emptyset$ else $\forall X \in$ F(e) $\Rightarrow |X \cap$ FSumm(e, k) $| \leq$ 1.*

Informally, if the number of facets is n and the size of the summary is k, *at least one feature* from each facet is included in the summary when $k > n$. If $k \leq n$, then *at most one feature* from each facet is included in the summary. E.g., a faceted summary of length 3 for the entity Marie Curie can be {f1, f2, f6} as shown in Figure 1.

Approach

The FACES approach generates faceted entity summaries that are both *concise* and *comprehensive*. Conciseness is about selecting a small number of facts. Comprehensiveness is about selecting facts to represent all aspects of an entity that improves coverage. Diversity is about selecting facts that are orthogonal to each other so that the selected few facts enrich coverage. *Hence, diversity improves comprehensiveness when the number of features to include in a summary is limited*. Conciseness may be achieved by ranking and filtering techniques. But creating summaries that satisfy both conciseness and comprehensiveness constraints simultaneously is not a trivial task. It needs to recognize facets of an entity so that the summary can represent as many facets (diverse and comprehensive) as possible without redundancy (concise). The number and nature of facets (corresponding to abstract concepts) in a feature set is not known a priori for an entity and is hard to guess without human intervention or explicit knowledge. Therefore, a supervised or unsupervised clustering (partitioning) algorithm with prescribed number of clusters to seek cannot be used in this context. Next, we describe the three main algorithms in our approach: *partitioning feature set into facets, ranking features within facets, and generating faceted entity summaries*.

Partitioning feature set

Our goal is to partition the feature set $FS(e)$ of an entity e. To achieve this goal, we adapted the Cobweb (Fisher 1987), an incremental system for hierarchical conceptual clustering. The algorithm embodies a top-down approach and creates a concept hierarchy where each node represents a concept with a probabilistic explanation/description. I.e., it clusters items utilizing attribute-value pairs associated with them and the clustering depends on the probability of attribute-value pairs for the items in each part of the partition. Further, it maximizes intra-cluster similarity and inter-cluster dissimilarity of items being clustered. This is closely related to the decisions that humans make in partitioning tasks (Fisher 1987). We chose this algorithm because: (1) it is hierarchical and hence the number of clusters need not be known a priori, (2) it creates groups for specific concepts, and (3) it is influenced by heuristics that humans use for grouping based on probability. Cobweb is also efficient compared to other hierarchical clustering algorithms (e.g., single-link, complete-link) because it is incremental.

Cobweb has been designed to work with attribute-value pairs for objects such as 'height - 6 ft' and 'weight - 120 pounds' for a person. In our problem, there are two attributes associated with each feature f: property and value. Values of these attributes are $Prop(f)$ and $Val(f)$, respectively. Moreover, Cobweb uses a heuristic measurement called *Category Utility (CU)* to determine partitions in the hierarchy by measuring partition quality. Let C_p be a node in the hierarchy with a set of features and has child clusters (partition) $C_1, C_2, .., C_n$. Then *CU* of partition $\{C_1, C_2, .., C_n\}$ of C_p can be computed as in Equation 1.

$$CU(C_p) = \frac{\sum_{x=1}^{n} P(C_x) \sum_{i=1}^{2} [P(A_i, V_i|C_x)^2 - P(A_i, V_i|C_p)^2]}{n} \quad (1)$$

$P(C_x)$ is the probability of a random feature belonging to child cluster C_x and (A_i, V_i) is the i^{th} attribute-value pair of the new feature being clustered. When a new feature is to be positioned in the hierarchy, Cobweb starts from the root node and performs one of the following operations based on the maximum *CU* score at each level until the new feature is settled in the hierarchy. They are: (1) *Insert* operation that inserts the feature into an existing cluster, (2) *Create* operation that creates a new cluster for the feature, (3) *Merge* operation that combines two clusters into one, and (4) *Split* operation that divides an existing cluster into several clusters. Cobweb uses a cut-off threshold to decide when to stop in the hierarchy and inherits agglomerative and divisive characteristics from merge and split operations, respectively. Furthermore, the split operation can be used to undo a merge operation performed earlier and vice versa.

CLASSIT (Gennari, Langley, and Fisher 1989) is a variation of Cobweb that is used in the text clustering domain, but requires a normal distribution for the values of the attributes. CLASSIT has been shown applicable in the document clustering context with the use of Katz's distribution where rich term distributions can be found (Sahoo et al. 2006). We do not have such a rich term distribution as in (Wang and Li 2010; Sahoo et al. 2006) for documents and hence it is difficult to use the CLASSIT implementation. Therefore, we adapt and modify the original Cobweb for our problem.

Input to the Cobweb is the feature set $FS(e)$ of an entity e and each feature f should have sufficient attribute-value pairs so that the *CU* function can group them accurately in the hierarchy. Each feature, modeled as two attribute-value pairs, does not have adequate details to group conceptually similar features. Hence, we expand the property and value of each feature f to get a set of words $WS(f)$ in such a way that expanded terms can be used to glean higher level abstract meaning. The terms in the word set can be mapped to attribute-value pairs required by the algorithm by mapping words in the word set to 1 and those not in the word set to 0.

Our partitioning algorithm uses labels associated with URIs. If labels are not available, the local name of the URIs are utilized. For the property name expansion, we first tokenize property name and remove stop words. Then we retrieve higher level abstract terms for the tokenized words. For the value expansion, we retrieve typing information (classes assigned to the resource), tokenize them, remove stop words, and expand them to include higher level abstract terms of the tokenized words. Tokenizing includes processing camelcase, spaces, punctuations, underscores etc. Then we add output of the property name and value expansion steps of feature f into a single word set $WS(f)$. See Figure 2 for example. In our implementation, we used hypernyms as the abstract terms taken from WordNet[1], a widely used online lexical database. Note that any service similar to WordNet could be used for this purpose. We are interested in using hypernyms because we need conceptually similar feature groups as facets from the clustering algorithm. If typing information is not available in the dataset or for the value (such as data-type property), an external knowledge source such as Wikipedia[2] can be used.

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

Feature (f)	Property expansion	Value expansion	Word set (WS(f))
birthPlace:Warsaw	{*birthPlace*, *birth*, *place*, beginning, point, area, locality, ..}	{*place*, *PopulatedPlace*, populated, point, area, locality, ..}	{*birthPlace*, *birth*, *place*, *PopulatedPlace*, beginning, populated, point, area, locality, ..}

Figure 2: Feature expansion example. Terms taken from WordNet as hypernyms are underlined.

We modified the original Cobweb *CU* implementation designed for nominal attributes shown in Equation 1 and adapted it to use only the terms in the word set $WS(f)$ for a feature f (omit 0 or 1). See Equation 2. W_i is a word appearing in $WS(f)$.

$$CU(C_p) = \frac{\sum_{x=1}^{n} P(C_x) \sum_{i} [P(W_i|C_x)^2 - P(W_i|C_p)^2]}{n} \quad (2)$$

Ranking features

Our approach ranks features that appear within each facet locally, and hence, related features do not affect the ranking result, unlike graph based ranking algorithms. A specific ranking algorithm influenced by the *tf-idf* technique has been used to get the top features from each facet to form the faceted summary. For a feature f and the value v of f, the ranking takes into consideration the informativeness/uniqueness (reflected using idf) of f ($Inf(f)$), and popularity (reflected using tf) of v ($Po(v)$). $Inf(f)$ is defined in Equation 3. N is the total number of entities in the data graph G. $Po(v)$ is the number of distinct triples in the data graph G that has the matching value v. Following the IR tradition we take the log of this value as expressed in Equation 4.

The ranking of features within a facet is then the product of the informativeness of the feature and popularity of the value as defined in Equation 5. Our intuition is that a feature (property-value pair) should be relatively rare (i.e., informative) to be interesting and not the property alone. Also when the value is popular in the dataset, it tends to help form a more human readable summary.

$$Inf(f) = log(\frac{N}{|\{e|f \in FS(e)\}|}) \quad (3)$$

$$\begin{aligned} Po(v) = log|\{triple\ t|\exists\ e, f : t \text{ ``}appears\ in\text{''}\ G \\ and\ t \equiv (e\ Prop(f)\ Val(f))\ and\ Val(f) = v\}| \end{aligned} \quad (4)$$

[2]http://www.wikipedia.org/ has category hierarchy that can be used similar to class labels

$$Rank(f) = Inf(f) \ * \ Po(Val(f)) \quad (5)$$

Generating faceted entity summary

Given the feature set $FS(e)$ of an entity e and a positive integer $k < |FS(e)|$, the process of faceted entity summary creation can be stated as follows. (1) First, each feature f in $FS(e)$ is enriched to have a wordset $WS(f)$. (2) The enriched feature set $FS(e)$ is input to the partitioning algorithm to create facets. The algorithm yields a dendrogram (hierarchical tree) for $FS(e)$ and it is cut at a desired level to get the facet set $F(e)$ of $FS(e)$. (3) Then, features in each facet are ranked using the ranking algorithm. (4) The top ranked features from each facet are picked according to Definition 6 to form the faceted entity summary of length k. In this implementation, we avoid picking features that have the same property name from each facet when $k > |F(e)|$.

Evaluation

We evaluate the FACES approach in two orthogonal ways: (1) Evaluate the quality of the summaries in comparison to the state-of-the-art systems using a gold standard. (2) Evaluate summary quality by user preference. We evaluate FACES against RELIN that outperformed earlier entity summarization tools and SUMMARUM which is DBpedia specific. We did not choose domain specific summarization tools like (Thalhammer, Knuth, and Sack 2012) as they are not applicable in general and require additional domain specific usage data. We do not consider the graphical entity summarization (Sydow, Pikuła, and Schenkel 2013) approach as it is different from an entity summarization captured by RELIN and FACES.

We selected the DBpedia dataset for our evaluation as it was the benchmark dataset selected in (Cheng, Tran, and Qu 2011) and contains entities that belong to different domains. We created a gold standard for the evaluation due to unavailability of the evaluation data of RELIN (as confirmed by the authors of RELIN). We randomly selected 50 entities[3] from DBpedia (English version 3.9) that have at least 17 distinct properties per entity. The average number of distinct features per entity is 44. Further, RELIN's results and user agreement for ideal summaries are not the same as those reported in (Cheng, Tran, and Qu 2011) because of the differences in the test set. We filtered out schema information and dataset dependent details such as *dcterms:subject*, *rdf:type*, *owl:sameAs*, *wordnet_type* and *Wikipedia related links* to ease manual evaluation and further they do not express facts about the entities. We extracted object-type properties for this dataset as the original set of triples is too large for manual evaluators, they are cleaner than data-type properties, and they contain more interesting information. Further, use of object-type properties is consistent with SUMMARUM, and hence enables a meaningful comparison. We asked 15 human judges with background in Semantic Web to select 5 and 10 feature length summaries for each of the entities. These are referred from now on as the *ideal summaries*. They were not given specific information about any system and asked to select a summary that can better represent the entity (facilitate quick identification). We provided them the Wikipedia page link of each entity in case they needed additional information about an unfamiliar entity. Each entity has at least 7 ideal summaries from 7 different judges and this comprises the gold standard for evaluation. All experiments were performed using a Core i7 3.4 GHz Desktop machine with 12 GB of RAM. We replicated DBpedia dataset locally and used caches for RELIN as mentioned in (Cheng, Tran, and Qu 2011). More details about the approach and gold standard dataset can be found at our web page[4].

Evaluation with the gold standard

Our objective is to show that faceted entity summaries produce results that are closer to human judgment. We configured FACES that produces the best possible results and RELIN according to its recorded optimal configuration. We empirically determined to cut FACES cluster hierarchies at level 3 which gave good results. We also set the cut-off threshold of Cobweb to 5, which gave the optimal results. For RELIN, we set the jump probability and number of iterations to 0.85 and 10, respectively. Authors of RELIN provided the source code of RELIN in absence of the evaluation data to replicate the environment. We replaced Google search service with Sindice API[5] as Google search API is no longer free of charge. Sindice indexes the LOD data and is adequate for this purpose. Further, we collected Google API and Sindice API search hits for a small random sample (5 entities) and applied RELIN to both API search hits. The results confirmed that the difference is negligible.

$$Agreement = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} |Summ_i^I(e) \cap Summ_j^I(e)| \quad (6)$$

$$Quality(Summ(e)) = \frac{1}{n} \sum_{i=1}^{n} |Summ(e) \ \cap \ Summ_i^I(e)| \quad (7)$$

We use the evaluation metrics used in (Cheng, Tran, and Qu 2011). When there are n ideal summaries denoted by $Summ_i^I(e)$ for $i = 1, .., n$ and an automatically generated summary denoted by $Summ(e)$ for entity e, the agreement on ideal summaries is measured by Equation 6 and the quality of the automatically generated summary is measured by Equation 7. In other words, quality of an entity summary is its average overlap with the ideal summaries for the entity. Our evaluation results and statistics are presented in Table 1. We modified the original RELIN algorithm to discard duplicate properties and named it as RELINM in Table 1. The modification did not yield much improvement in the results

[3]Selected entities are from the domains of politician, actor, scientist, song, film, country, city, river, company, game, etc

[4]http://wiki.knoesis.org/index.php/FACES

[5]http://sindice.com/developers/searchapiv3

System	Evaluation 1 - Gold standard					Evaluation 2 - User preference	
	k = 5			k = 10		User preference	
	Quality	FACES % ↑	Time/Entity	Quality	FACES % ↑	User study 1	User study 2
FACES	**1.4314**	NA	**0.76 sec.**	**4.3350**	NA	**84%**	**54%**
RELIN	0.4981	187 %	10.96 sec.	2.5188	72 %	NA	NA
RELINM	0.6008	138 %	11.08 sec.	3.0906	40 %	16%	16%
SUMMARUM	1.2249	17 %	NA	3.4207	27 %	NA	30%
Agreement	1.9168			4.6415			

Table 1: Evaluation of the summary quality and FACES %↑ = 100 * (FACES result - Other system result) / (Other system result) for k=5 and k=10, respectively, and average time taken per entity for k=5 for Evaluation 1. Evaluation 2 measures user preference % for each system. (NA stands for Not Applicable)

and this was to test whether re-ranking mechanisms can improve RELIN results. We ran each algorithm for all the 50 entities 5 times and recorded the average time taken per entity in seconds. According to the results, FACES achieved 138% and 40% increase in quality against RELINM and 187% and 72% increase in quality against RELIN, and 17% and 27% increase in quality against SUMMARUM for k=5 and k=10, respectively. Even though FACES achieves superior results compared to RELIN, it does not compromise its efficiency. FACES is much more efficient (14 times faster) than RELIN as shown in Table 1 in summary computation (i.e., running time). Time for SUMMARUM is not applicable (NA) as we get results from a web service but can be assumed to be similar to RELIN. FACES can be further improved to compute summaries on the fly.

We conducted the *paired t-test* to confirm the significance of the FACES's mean summary quality improvements over others. For k = 5 and k = 10, P values for FACES against RELINM are 2.04E-10 and 1.92E-14 and for FACES against SUMMARUM are 0.029 and 3.71E-7. When P values are less than 0.05, the results are statistically significant.

Evaluation of user preference

We carried out two blind user evaluations (users didn't know which system produced which summaries) to see how users rated summaries created by FACES, RELINM, and SUMMARUM. We randomly selected 10 instances from our evaluation sample and created k = 5 length summaries and had 69 users participate in total. The results are shown in Table 1. For the first user evaluation, we showed them summaries of FACES and RELINM side by side and asked them to select the best summary that helped them to identify the entity. The users average preference was 84% for FACES and 16% for RELINM. This shows that unique properties alone are not desirable to humans. In the second user evaluation, we showed users all three system summaries and their preferences were 54%, 16%, and 30% for FACES, RELINM, and SUMMARUM, respectively. This shows that users sometimes prefer popularity as in SUMMARUM but not in all the cases. RELINM got almost the same percentage in both experiments reflecting that users preferred unique features for some entities. Moreover, results suggest that users like a balanced (unique and popular) and diversified approach like in FACES and confirm our claim that the diversity makes summaries more human friendly.

Discussion

In summary, our evaluation shows that FACES performs better in all cases. It was able to achieve 138% and 40% increase over RELINM and 17% and 27% increase over SUMMARUM in summary quality for k=5 and k=10, respectively. The results of the paired t-test confirms that FACES results are not random and consistently outperforms the other two systems. FACES achieves this by its non-trivial facet identification process and persistence in selecting a diverse summary based on the facets. Note that agreements between ideal summaries are not very high in the sample dataset due to the large number of distinct features present in each entity (see Table 1). The second evaluation shows that the faceted summaries are desirable to human users.

FACES behaves similar to an ordinary summarization algorithm such as RELIN and SUMMARUM when there are few facets available. It behaves as if it is ranking a flat list of features. A key reason why RELIN (including RELINM), which is based on PageRank that exploits both informativeness and relatedness measures, underperforms is that the summary can include redundant and correlated features. This also affects SUMMARUM. The redundancy comes at the expense of reduced coverage. We address this limitation by emphasizing diversity to suppress redundant features[6] and improve coverage by identifying facets to pick representative features. Figure 3 shows entity summary examples generated by the three approaches.

We observed that sometimes FACES clusters features differently from what we expect. For example, *spouse* property of Barack Obama is clustered into a facet that contains *vicePresident* property. This happens because sometimes typing information of values is too specific[7] and affects the clustering process. Moreover, FACES performs better in almost all cases according to the second evaluation except for some specific entities. E.g., *Usain Bolt* is an Olympic athlete with many records. Some users preferred facts about his 100 meter records over his 200 meter records, while both information were present in a facet. FACES can generate either of the two facts in a summary (based on a subjective ranking within a facet) and the tie can be broken. We can further investigate how to effectively combine diversity, uniqueness, and popularity of features as the user preference varies. Fur-

[6]May not be syntactically the same but conceptually similar

[7]*Michelle Obama*'s typing information is similar to a politician

FACES	RELINM	SUMMARUM
<birthPlace, Warsaw>	<isPrimaryTopicOf, Marie_Curie>	<birthPlace, Poland>
<workInstitutions, University of Paris>	<wasDerivedFrom, oldid=547107936>	<birthPlace, Warsaw>
<field, Physics>	<knownFor, Polonium>	<birthPlace, Russian_Empire>
<spouse, Pierre Curie>	<almaMater, ESPCI>	<field, Physics>
<deathPlace, Passy, Haute-Savoie>	<deathPlace, Passy, Haute-Savoie>	<field, Chemistry>

Figure 3: Entity summaries for the entity Marie Curie by each system. k = 5 and the size of feature set is 39.

thermore, clustering phase of FACES can be tuned for fine grained grouping by modifying the enrichment process. E.g., adding hyponyms to the word set makes features containing *birthPlace* and *stateOfOrigin* properties fall into different facets.

Conclusion

We have investigated how to create entity summaries that are *concise* and *comprehensive* for the purpose of quick identification of an entity. We adapted a well known incremental hierarchical conceptual clustering algorithm for entities (in RDF format) to identify facets (that addressed diversity) and developed intra-cluster ranking algorithm for features (that addressed uniqueness and popularity) to create *faceted entity summaries*. We showed that faceted entity summaries created by combining diversity, uniqueness, and popularity are better representatives for an entity and closer to ideal ones. Our approach shows superior results (improvement in summary quality in the range 17% - 187%) and does not require pre-computation of values like the existing systems. In future, we plan to investigate facet ranking, which facets to select and how many features to pick from each facet in creating "personalized" and "balanced" summaries.

Acknowledgments

This work was supported by the National Science Foundation under award 1143717 III: EAGER Expressive Scalable Querying over Linked Open Data.

References

Anyanwu, K.; Maduko, A.; and Sheth, A. 2005. Semrank: ranking complex relationship search results on the semantic web. In *Proceedings of the 14th international conference on World Wide Web*, 117–127. ACM.

Auer, S.; Lehmann, J.; Ngomo, A.-C. N.; and Zaveri, A. 2013. Introduction to linked data and its lifecycle on the web. In *Reasoning Web. Semantic Technologies for Intelligent Data Access*. Springer. 1–90.

Cheng, G.; Tran, T.; and Qu, Y. 2011. Relin: relatedness and informativeness-based centrality for entity summarization. In *The Semantic Web–ISWC 2011*. Springer. 114–129.

Ding, L.; Pan, R.; Finin, T.; Joshi, A.; Peng, Y.; and Kolari, P. 2005. Finding and ranking knowledge on the semantic web. In *The Semantic Web–ISWC 2005*. Springer. 156–170.

Fisher, D. H. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine learning* 2(2):139–172.

Franz, T.; Schultz, A.; Sizov, S.; and Staab, S. 2009. Triplerank: Ranking semantic web data by tensor decomposition. In *The Semantic Web-ISWC 2009*. Springer. 213–228.

Gennari, J. H.; Langley, P.; and Fisher, D. 1989. Models of incremental concept formation. *Artificial intelligence* 40(1):11–61.

Gunaratna, K.; Thirunarayan, K.; Jain, P.; Sheth, A.; and Wijeratne, S. 2013. A statistical and schema independent approach to identify equivalent properties on linked data. In *Proceedings of the 9th International Conference on Semantic Systems*, 33–40. ACM.

Mani, I. 2001. *Automatic summarization*, volume 3. John Benjamins Publishing.

Nenkova, A., and McKeown, K. 2012. A survey of text summarization techniques. In *Mining Text Data*. Springer. 43–76.

Sahoo, N.; Callan, J.; Krishnan, R.; Duncan, G.; and Padman, R. 2006. Incremental hierarchical clustering of text documents. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, 357–366. ACM.

Sydow, M.; Pikuła, M.; and Schenkel, R. 2013. The notion of diversity in graphical entity summarisation on semantic knowledge graphs. *Journal of Intelligent Information Systems* 41(2):109–149.

Thalhammer, A., and Rettinger, A. 2014. Browsing dbpedia entities with summaries. In *The Extended Semantic Web Conference 2014. Poster & Demo.*

Thalhammer, A.; Knuth, M.; and Sack, H. 2012. Evaluating entity summarization using a game-based ground truth. In *The Semantic Web–ISWC 2012*. Springer. 350–361.

Tonon, A.; Catasta, M.; Demartini, G.; Cudré-Mauroux, P.; and Aberer, K. 2013. Trank: Ranking entity types using the web of data. In *The Semantic Web–ISWC 2013*. Springer. 640–656.

Wang, D., and Li, T. 2010. Document update summarization using incremental hierarchical clustering. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, 279–288. ACM.

Xu, D.; Cheng, G.; and Qu, Y. 2014. Facilitating human intervention in coreference resolution with comparative entity summaries. In *The Semantic Web: Trends and Challenges*. Springer. 535–549.

Zhao, L., and Ichise, R. 2012. Graph-based ontology analysis in the linked open data. In *Proceedings of the 8th International Conference on Semantic Systems*, 56–63. ACM.