BAFREC: Balancing Frequency and Rarity for Entity Characterization in Linked Open Data

Hermann Kroll Denis Nagel Wolf-Tilo Balke
Institute for Information Systems, TU Braunschweig, Braunschweig, Germany
kroll@ifis.cs.tu-bs.de denis.nagel@tu-bs.de balke@ifis.cs.tu-bs.de

Abstract

Today's growth of linked open data (LOD) sources calls for summarization systems to help users to navigate through large volumes of data. A major task is *entity summarization*, where a most meaningful subset of all available information about entities has to be selected. In particular, the selected information has to *characterize* each entity with high precision. These summaries can then be used for a wide variety of applications such as initial presentation (so-called info boxes), entity disambiguation, or entity reconciliation. This paper introduces **BAFREC**, a novel entity summarization method balancing frequency and rarity metrics for all entity properties in a sophisticated manner. In contrast to simply choosing most popular or most frequent concepts, we design a new strategy: BAFREC first splits all facts about some entity into categories and then rates each category using a specifically tailored metric. For instance, some facts like type information are preferred with respect to their rarity, i.e. picking the most specialized concept, while others may be rated according to their general popularity. The evaluation against the ESBM benchmark shows that especially for computing short summaries, BAFREC outperforms commonly applied approaches.

1 Introduction

Today, a lot of popular knowledge is already contained in graph-structured databases or knowledge bases like *DBpedia*[1] or *LinkedMDB*[2] based on RDF representations. Simple statements generally represent real-world facts (e.g., the age of a person) and are stored as triples (subject, predicate, object). The subjects of such facts provide the core for entity summarization. Encoded by unique identifiers (URIs) they represent real-word objects (so-called entities) that are described by one or more facts. *Entity Summarization* is then the task of describing each single entity by selecting a limited subset of all available facts with the respective subject. Clearly, in such entity summaries there is a trade-off between the users' cognitive load imposed by larger summaries and the general separability of entities. Thus, the difficulty lies in selecting only the *most relevant facts*, i.e. a *strictly limited* amount of information that is best suited to describe the entity and at the same time to distinguish it from others. If we want to differentiate some entity from others in the sense of distinguishing various *classes of entities*, in the following we refer to the task as *Entity Characterization* to distinguish it from other summarization tasks.

Let us look at an example: in *DBpedia*, some entities comprise more than 100 or even 1000 facts. But for the entity *Barack Obama* the fact that he was the *44th president of the USA* is probably more important than for instance the name of his birthplace. Thus, to compute brief entity descriptions, first a ranking of all existing facts is necessary. Yet, the expected results (and thus the perceived quality of a summary) may differ. Gunaratna et al. argue that if multiple persons select *best* entity summaries, it is important to look at their respective consensus, since some persons may prefer facts about *Barack Obama's* life to facts about his career [GTS15].

In this work, we propose a greedy algorithm that is designed to split facts into disjoint sets and rank them by their relevant measures. We present recent approaches in the field of entity summarization in Sect.

[1] https://wiki.dbpedia.org, last access: 24.07.18
[2] http://www.linkedmdb.org, last access: 24.07.18

2. We present our concept in Sect. 3, including necessary definitions and operationalized metrics in Sect. 4. Next, the algorithm is described in Sect. 5. The results regarding the shared task are presented in Sect. 6. Lastly, we conclude our approach in Sect 7.

2 Related Work

Different strategies for Entity Summarization are already applied. *DIVERSUM* is based on choosing facts regarding their diversity [SPS10]. Diversity means that facts with labels not yet represented in the result set are preferred over already included ones.

FACES uses a different approach by categorizing the facts before ranking [GTS15]. For grouping the information into topics *FACES* relies on using WordNet [Mil95], representing a free lexical database which allows computing similarities between words. Facts inside the categories are then ranked by a combination of the uniqueness of predicates in the database and the popularity of the connected entities. This way the most important facts stemming from different categories (called facets) can sequentially be picked which also results in a diverse summarization. To obtain diversity semantic similarity measures between predicate names are necessary. Leacock et. al. discussed the idea to compute such a similarity using most informative classes or shortest paths between senses of two words [LC98].

RELIN and *LinkSUM* both extend the concept of popularity by using Google's PageRank Algorithm [CTQ11, TLR16, PBMW99], to analyze the structure of the linked data. For fact selection, *RELIN* then applies a novel relatedness measure using search engine results to minimize the redundancy in their result set. *LinkSUM* on the other hand, prefers facts that are strongly connected to the entity by combining PageRank with a BackLink approach.

Homoceanu et. al. investigated how to derive typical attributes for entities based on the concept of family resemblance within certain classes [HB15]. Experiments on Web data show that indeed companies e.g., in the field of ICT can clearly be distinguished by typically mentioned attributes from companies e.g., in the financial sector. Hence, the idea to find both, specialized and unique information for diversification is first introduced here. **BAFREC** further refines this strategy.

3 Basic Algorithm Design

BAFREC (BAlancing Frequency and Rarity for Entity Characterization) combines the best ideas of the techniques reviewed above. Yet, it also introduces central concepts to further improve the quality of today's Entity Summarization. A key observation is that whenever we need to choose between facts, the utility of a specific kind of fact is strongly coupled to the interestingness of its respective value. For instance, take our sample entity *Barack Obama*. There may be typical kinds of facts, such as birthdate, country, or type, available for ranking. On one hand, these facts obviously provide very different types of information: while a birthdate is quite a special information, type information seems to be of a more general kind and thus may be more desirable to characterize an entity. On the other hand, a statement that *Barack Obama* is of the type *person* may be far less useful than that he is of type *politician* or even *US president*. The core idea is thus to split facts into two major categories: meta-information and data-information. In turn, this reflects on the individual usefulness of frequency and rarity of facts.

Meta-information describes structural knowledge about an entity usually with respect to its placement within some suitable ontology. As argued above, facts with greater ontological depth are often more useful. This is because choosing the fact that *Barack Obama* is an *US president*, also includes the information that he is a *politician*, *person*, a *living thing*, and so on. Thus, whenever for some entity multiple predicates with the same label exist, BAFREC optimizes the uniqueness by assessing each fact's depth within some ontology available for that label. Should there exist different meta information like e.g., type and class, BAFREC always prefers using more frequent concepts in the entire database to rare ones. Additionally BAFREC is able to handle missing ontologies: instead of using depth inside an ontology, the rarity of the concept can be used. We assume, that a concept like *president* appears more rare than a *person*.

Data-information are facts describing real-world properties of an entity like its name, birthdate, address, or similar information. To rank the usefulness of these facts, measures like popularity and/or frequency are essential. For instance, for summarizing facts about a movie it is important to choose facts about famous directors and actors. Therefore, we use a simple popularity metric, which assesses the frequency of some object belonging to a fact. Additionally, we implemented diversity inspired by *DIVERSUM's* concept of diversity, but used WordNet to determine concept similarity. To foster high diversity, we chose an iterative approach: while adding facts to the result set, we decrease scores of facts having predicate names identical or similar to the predicate names already in the result set.

4 Operationalizing The Metrics

This chapter describes the necessary metrics and used model of our Entity Summarization BAFREC in detail. At first, we define a graph database as an edge-labeled directed graph $G = (V, \Sigma, E)$ with a finite set of nodes V, a finite (label) alphabet Σ and a directed labeled edge relation $E \subseteq V \times \Sigma \times V$. We call an edge a fact f consisting of (s, p, o) with subject $s \in V$, predicate $p \in \Sigma$ and object $o \in V$.

First, to categorize the facts into meta- and data-information, we introduce a $isMeta_{OD}$ predicate. Fact f is meta-information if the predicate p is contained in an ontology. To prohibit checking all ontology predicates, we analyze the domain of this predicate. A domain forms the beginning of a predicate like *http://www.dbpedia.org*. Therefore, we define a set of typical ontology predicate domains OD. With that, we can check if a predicate name starts with a typical ontology domain (*startswith*). Formally, our categorization predicate $isMeta_{OD}$ regarding ontology predicate domain OD is defined as:

$$isMeta_{OD}(f) :\Leftrightarrow \exists_{n \in OD} StartsWith(p, n) \quad (1)$$

Next, in ontologies nodes are inserted into a hierarchy of subclasses and superclasses. In a database structured this way, it is possible to iteratively travel from a node over its superclasses until reaching the highest superclass. We call this the root element of the node. We define the depth of a node inside the ontology, as the shortest path between the root element and the given node. E.g., the shortest path between *Thing* and a node is computed using the ontology for *DBpedia*. The shortest path is computed by using the Dijkstra algorithm. In short, a depth of a node inside an ontology O with root element r is defined as:

$$depth_O(l) = ShortestPathDijsktra(v, r) \quad (2)$$

Sometimes no ontology information is available. In this case, we use the concept of rarity instead. If no ontology is available for a dataset like for *LinkedMDB*, we assume, that if a concept (node) is rare inside the database, it is treated like having a larger depth inside an ontology, because larger depth means a more specialized concept, e.g., there exist more persons than presidents. Therefore, we regard the indegree of a node $v \in V$ as:

$$Inc_G(v) = |\{(s, p, o) \in E \mid o = v\}| \quad (3)$$

Formally, we define the rarity of a node as:

$$rarity_G(v) = \frac{1}{Inc_G(v)} \quad (4)$$

For ranking different predicate regarding meta-information, we want to prefer frequent concepts inside the database. We formally define the frequency of a predicate p as:

$$freq_G(p) = |\{(s, p, o) \in E\}| \quad (5)$$

For selecting data-information, we use the concept of popularity for ranking these information. PageRank offers the idea of computing popularity by the number of incoming links and propagating this information to all neighbours of a node. Instead of using computational heavy iterations, we use the popularity metric introduced in [GTS15]. The idea is to rank a fact by the number of objects incoming edges obtaining popularity. Formally, we define the popularity of a fact $f = (s, p, o)$ as:

$$pop_G(f) = \log\left(Inc_G(o)\right) \quad (6)$$

Finally, we want to arrange predicates into groups to increase the diversity of our result. To compare two predicate names regarding their topic, we need a semantic similarity measure between words. Therefore we use WordNet, because there exists multiple similarity measures based on WordNet: *ws4j*[3] offers an implementation for such a similarity between words given as sim_{ws4j}. Additionally, we need to split a predicate name like *starringActor* into *starring* and *actor*, because *starringActor* cannot be found inside the WordNet dictionary. To transform the combined predicate names, we split them at capital letters and use lower case for all words. We call this tokenizing a predicate name. We define a predicate name p as a concatenation of single words $w_1 \dots w_n$. We define the length of predicate p as the number of its tokens, e.g., $|p| = n$ if tokenization yields $w_1 \dots w_n$. To compare two different words, we first tokenize into single words. If both names consist of one word, we apply the given WordNet similarity directly. In general, we define the similarity as follows:

$$sim_{diversity}(p_1, p_2) = \frac{\sum_{w_i \in p_1} \sum_{w_j \in p_2} sim_{ws4j}(w_i, w_j)}{|p_1| \cdot |p_2|} \quad (7)$$

Additionally, we need to check whether a similar predicate name is already used in the summary. Therefore, we compute the arithmetic mean of the diversity similarities ($sim_{diversity}$) between the predicate name and all preselected predicate names.

5 Algorithm

In the following, we introduce our greedy algorithm for Entity Summarization called BAFREC. Our pro-

[3] https://code.google.com/archive/p/ws4j/, last access: 26.07.18

cedure expects the entity to be categorized as an input as well as all facts to be ranked. The algorithm generates a ranked list of the facts as an output. Every fact is included inside the ranking, so that an arbitrary number k of facts can be selected subsequently. As a parameter, BAFREC needs a ratio between meta- and data-information introduced later.

First, BAFREC categorizes the given facts of an entity into meta- and data-information ($isMeta_{OD}$). BAFREC uses the *DBpedia* ontology[4]. Next, both categories are ranked: for meta-information, the depth inside an ontology ($depth_O$) is used, if available. Sometimes, if like for *LinkedMDB* no ontology is available, BAFREC instead uses the rarity metric ($rarity_G$) scoring the facts. BAFREC prefers ontologies automatically, because the shortest distance starts at 1, while rarity is normalized between 0 and 1. Next, the meta-information is grouped by the predicate names. Then, the groups are ranked with the frequency metric ($freq_G$). For each group the best scored fact is selected. This is repeated, until all meta-information is included inside the result. Optimizing the performance, the metrics are evaluated once and stored in an index.

Next, BAFREC rates the data-information using the introduced popularity metric (pop_G). First, the most popular fact is selected. Now the following is repeated, until all data-information is included inside the result: each not selected fact is scored by multiplying the inverse diversity similarity with its popularity score. The inverse diversity similarity is computed using the introduced metric ($1 - sim_{diversity}$) between the given not selected fact and the set of selected facts. Therefore the arithmetic mean of the diversity similarity ($sim_{diversity}$) is used. Then, the highest ranked fact is included in the result.

Sometimes, all remaining, not selected facts are scored with the score zero because either the popularity is zero (only one incoming edge for an entity), or the inverse diversity similarity is zero (predicate names are too similar). Keep in mind that at this point all other facts are ranked and already included inside the result set. For the remaining facts BAFREC uses the frequency metric multiplied with the rarity metric for fact ranking. This is done, to distinguish the facts deterministically instead of choosing a random order.

Finally, we derived two individually ranked sets, one for meta- and one for data-information. To aggregate both ranked sets into a single ranking we tested different strategies. In the end, we decided for a simple integration scheme: a weighted round-robin strategy. That means starting with a meta-information a certain number of data-information is added to the integrated list, then the next- ranked meta-information is added and so on. The ratio can be flexibly adjusted, but throughout our experiments, a ratio of 1:3 showed best results while selecting meta-information first. A Java implementation of BAFREC including all generated results for the given benchmark are available as open source project at GitHub[5].

[4]https://wiki.dbpedia.org/services-resources/ontology, last access: 25.07.18

Table 1: Comparison of F-measure (ESBM)

k	Database	**BAFREC**	FACES-E	DIVERSUM	CD
5	DBpedia	**34.9%**	28.5%	26.0%	29.9%
	LinkedMDB	**33.3%**	25.2%	22.2%	21.5%
	Combined	**34.4%**	27.6%	24.9%	26.7%
10	DBpedia	50.5%	52.7%	52.2%	**53.1%**
	LinkedMDB	33.3%	34.8%	**36.5%**	32.6%
	Combined	45.6%	47.6%	**47.7%**	46.7%

6 Experiments

All experiments in this paper are based on the ESBM Benchmark[6] as ground truth. To perform the experiments we used an Intel Core-i7 4870HQ@2,5Ghz with 16GB RAM and a Virtuoso graph database[7] containing dumps of *DBpedia* and *LinkedMDB* to compute the necessary measures required by our algorithm. We chose a ratio of 1:3 between meta- and data-information. All predicates stemming from the domains *http://www.w3.org/2000/01/rdf-schema#* and *http://www.w3.org/1999/02/22-rdf-syntax-ns#* are considered as meta-information. The absolute run-time of our algorithm on both datasets is about 75 seconds (including all necessary database queries) or 5 seconds (excluding database query times) and thus quite practical. We report results for all measures given by the benchmark tool, i.e. F-measure and mean average precision (MAP) at k=5 and k=10 for both datasets individually and combined. The results regarding F-measures are shown in table 1 and MAPs are shown in table 2. While the benchmark website analyzed six entity summarization tools for the shared task, i.e. *FACES-E*, *DIVERSUM*, *CD*, *RELIN*, *FACES* and *LinkSum*, for brevity in both tables we only report results for the first three approaches. This is because BAFREC consistently outperforms *RELIN*, *FACES* and *LinkSum* in all test cases.

Below, we compare BAFREC with the benchmark results of the algorithm, which yields the best scores regarding a database. As shown in table 1 for producing short summaries (k = 5) to characterize entities, BAFREC achieves best results outperforming all six approaches. With a plus of 5% for *DBpedia* and 8.1% for *LinkedMDB* BAFREC outperforms the given

[5]https://github.com/HermannKroll/EntityCharacterization
[6]http://ws.nju.edu.cn/summarization/esbm/, l. a.: 30.07.18
[7]https://virtuoso.openlinksw.com, last access: 25.07.18

Table 2: Comparison of MAP (ESBM)

k	Database	**BAFREC**	FACES-E	DIVERSUM
5	DBpedia	**40.9%**	35.4%	31.6%
	LinkedMDB	**34.2%**	25.8%	26.9%
	Combined	**39.0%**	32.6%	30.2%
10	DBpedia	**56.1%**	52.9%	51.1%
	LinkedMDB	35.5%	36.1%	**38.8%**
	Combined	**50.2%**	48.1%	47.6%

benchmark results for *CD* and *FACES-E* which yield the second highest values. Regarding MAP BAFREC again obtains a plus of 5.5% for *DBpedia* compared with *FACES-E* and a plus of 7.3% for *LinkedMDB* compared with *DIVERSUM*. Consequently, BAFREC achieves the best summaries at k = 5 with about 34.4% F-measure and 39.0% MAP.

The results at k=10 show that even though our approach is specifically suited for small entity characterizations BAFREC still obtains quite good results. Regarding F-measure BAFREC lags behind about 2.6% for *DBpedia* (*CD*) and about 3.2% for *LinkedMDB* (*DIVERSUM*). Comparing MAP BAFREC outperforms *FACES-E* with a plus of 3.2% for *DBpedia* while lagging about 3.3% behind *DIVERSUM* regarding *LinkedMDB* . Consequently, there seems to be no universal strategy which works best for both datasets. For *DBpedia CD* and *BAFREC* obtain the best results, whereas *DIVERSUM* achieves the highest scores regarding *LinkedMDB*.

Summarized, BAFREC outperforms every other approach analyzed in the ESBM benchmark at producing short summaries (k = 5). As summary sizes grow, there is no strategy which works best for both datasets. Meanwhile, BAFREC obtains comparable results at larger summary sizes (k = 10).

7 Conclusions

We introduced BAFREC, a novel strategy for Entity Summarization by balancing frequency and rarity metrics. In brief, it builds on the concept of splitting facts into meta- and data-information, i.e. treating structural information and real-world properties of entities individually. Furthermore, we have analyzed different metrics for each information category and developed an efficient greedy algorithm to support fact diversity. The experiments against the ESBM benchmark show that BAFREC is especially useful for entity characterization, i.e. short and concise summaries used for entity classification. Indeed, BAFREC consistently outperforms all commonly applied techniques for summaries of length 5. With increasing summary sizes, experiments begin to show a different picture: it seems hard to define an overall best strategy. Although BAFREC still obtains best results regarding MAP, each algorithm has its own strength and there is no clear winner outperforming all the others. A reason for this could lie in a weak inter-rater-agreement when building the benchmark. We believe, that diversity between ratings is evoked by human variety. With increasing summary sizes, opinions between the evaluators seem to differ more and thus, different strategies may show their individual strengths.

References

[CTQ11] Gong Cheng, Thanh Tran, and Yuzhong Qu. Relin: Relatedness and informativeness-based centrality for entity summarization. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Noy, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2011*, pages 114–129, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[GTS15] Kalpa Gunaratna, Krishnaprasad Thirunarayan, and Amit P Sheth. Faces: Diversity-aware entity summarization using incremental hierarchical conceptual clustering. In *AAAI*, pages 116–122, 2015.

[HB15] Silviu Homoceanu and Wolf-Tilo Balke. A chip off the old block-extracting typical attributes for entities based on family resemblance. In *International Conference on Database Systems for Advanced Applications*, pages 493–509. Springer, 2015.

[LC98] Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

[Mil95] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[SPS10] M. Sydow, M. Pikula, and R. Schenkel. Diversum: Towards diversified summarisation of entities in knowledge graphs. In *2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010)*, pages 221–226, March 2010.

[TLR16] Andreas Thalhammer, Nelia Lasierra, and Achim Rettinger. Linksum: Using link analysis to summarize entity data. In Alessandro Bozzon, Philippe Cudre-Maroux, and Cesare Pautasso, editors, *Web Engineering*, pages 244–261, Cham, 2016. Springer International Publishing.