FaSets: Discovering Faceted Sets of Entities

Koninika Pal*
IIT Palakkad, India
kpal@iitpkd.ac.in

Hiba Arnaout*
TU Darmstadt, Germany
hiba.srnsout@tu-darmstadt.de

Simon Razniewski*
Bosch Center for AI, Germany
Simon.Razniewski@de.bosch.com

Gerhard Weikum
MPII Saarbruecken, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

Computing related entities for a given seed entity is an important task in exploratory search and comparative data analysis. Prior works, using the seed-based set expansion paradigm, have focused on the single aspect of identifying homogeneous sets with high pairwise relatedness. A few recent works discuss cluster-based approaches to tackle multi-faceted set expansion, however, they fail in harnessing the specificity of the clusters and generating an explanation for them. This paper poses the multi-faceted set expansion as an optimization problem, where the goal is to compute multiple groups of entities that convey different aspects in an explainable manner, with high similarity within each group and diversity across groups. To extend a seed entity, we collect a large pool of candidate entities and facets (e.g., categories) from Wikipedia and knowledge bases, and construct a candidate graph. We propose FaSets, an efficient algorithm for computing faceted groups of bounded size, based on random walks over the candidate graph. Our extensive evaluation shows the superiority of FaSets against prior baselines, with regard to ground-truth collected from crowdsourcing.

CCS CONCEPTS

• **Information systems** → **Information retrieval**.

KEYWORDS

web mining, set expansion, entity ranking.

ACM Reference Format:
Koninika Pal, Hiba Arnaout, Simon Razniewski, and Gerhard Weikum. 2024. FaSets: Discovering Faceted Sets of Entities. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3589335.3651924

1 INTRODUCTION

Motivation. Computing related entities for a given entity is a key task for search, recommendation and exploratory data analysis. For example, when users express interest in a celebrity, company or movie query and click, search engines and content platforms (e.g., Youtube) not just return information about the entity of interest, but also suggest exploring highly related entities. In set expansion, starting from one or several seed entities, the task is to compute highly related entities. This is enabled either by leveraging large Knowledge Graphs (KGs) in combination with machine learning over the underlying contents and user signals [23] or by explicitly gathering related entities with explainable labels from lists, tags and tables on the web (or entity mark-up and co-occurrences in unstructured content) [15, 20, 25, 26, 35, 36].

For example, starting with *Jeff Bezos* as a seed entity, an algorithm could yield the set {*Elon Musk, Sundar Pichai, Warren Buffet, George Lucas, MacKenzie Bezos, Tom Cruise, Amazon, Alibaba*}. Unfortunately, not only does this list conflate entities of different types, but it also does not give any clue about why and how these people are similar or related to Jeff Bezos. In fact, their relatedness stems from very different aspects.

This calls for an aspect-aware refined approach, with labels or other explanations for groups of related entities. In this paper, we introduce a new model and computational task of discovering *faceted entity sets*. Given an input entity and a large set of potential facets each in the form of a labeled entity set, the task is to compute a compact group of facets with a small set of salient entities such that (i) each group is highly related to the seed entity, (ii) the entities per group are highly related to each other, and (iii) the selected groups diversify the overall picture, by being pairwise dissimilar. For example, for Jeff Bezos, a faceted output of this kind could be a set of three facets, each with two representative entities:
- *Tech Company CEOs:* {*Elon Musk, Sundar Pichai*},
- *Billionaires:* {*Jack Ma, George Lucas*},
- *Newspaper Owners:* {*Warren Buffet, William Randolph Hearst*}.

If we want more facets, we could add *Amazon Employees*, *Princeton Alumni* and more. If we want more entities per facet, we could go deeper into the underlying lists, tables and tag-sets. The challenge is to discover the best output from thousands of candidates for the facets and even more candidates for the entities per facet.

Approach and Contribution. Pattern-based bootstrapping approaches [27, 35] expand seeds using refined text-based features collected in each iteration. However, they are prone to concept drifting due to the inclusion of semantically ambiguous expanded entities in the iterative procedures. On the other hand, cluster-based approaches [25, 26, 40] categorize the expanded set into multiple facets but fails to generate interpretable labels for the clusters. Additionally, these approaches are not able to harness refined subtopics within a cluster.

*The work was done while authors were affiliated with MPII Saarbruecken.

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05…$15.00
https://doi.org/10.1145/3589335.3651924

This paper presents FaSets, a methodology for discovering compact sets of faceted groups of entities, to provide a multi-perspective gist of related entities for a given seed entity. Our approach taps into interpretable features from KGs (like YAGO [30], Wikidata [34]) and from categories and infobox values in Wikipedia. The richness of Wikipedia often yields a huge number of candidate facets, often several thousands for a single seed entity. For example, infobox values for entity *Jeff Bezos* yield facets such as *known for Founding Amazon*, *occupation investor*, *occupation philanthropist*, and he appears in a total of 203 Wikipedia categories (incl. non-leaf categories till level three from leaf node) such as *American billionaires*, *Businesspeople from Houston*. Here, we face a combinatorial space of options for identifying the best output of a desired size, say three facets with five entities each. We show that the faceted-set expansion problem is NP-hard. For a tractable solution, we devise an iterative algorithm that operates over a judiciously constructed similarity graph of candidate entities by exploiting relevant facets, and they are further used to generate the explanation for the expansion.

The salient contributions of this work are as follows:

- We define a generalized problem of faceted set expansion,
- We develop an efficient and effective algorithm based on random walks over judiciously constructed candidate graphs,
- We report extensive experimental studies with data from three domains (people, companies, movies), showing that FaSets outperforms state-of-the-art baselines.
- The datasets and source code are available here.

While this paper focuses on the data-mining and knowledge-discovery problem itself, we foresee several use cases, such as recommender systems [10], KG curation [4], entity linking [28], where groups of faceted entities are beneficial.

2 PROBLEM STATEMENT

Faceted Set Expansion (*FSX*): Consider a universe of entities $E = \{e_1, e_2, \ldots, e_n\}$ which are distributed over a set of labeled facets $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ where $S_i \subset E$. Given a query $q \in E$, two parameters l and k representing the number of output groups and their size, the objective is to compute l sets of entities, called *faceted groups*, each of size k: $\mathcal{G} = \{G_1, G_2, \ldots G_l\}$, such that each $G_j \subset S_i$ for some input facet S_i from which G_j can inherit its label. And, these l faceted groups must satisfy the following three conditions:

1. The pairwise similarity between the query q and faceted groups G_i, i.e., the similarity summed up over all $e_i \in G_i$, is maximized.
2. The pairwise similarity between entities in each group G_i, i.e., the similarity summed up over all entity pairs $e_i, e_j \in G_i$, is maximized to reflect coherence inside the group.
3. The pairwise similarity across groups G_i, G_j, summed up over all entity pairs (e_i, e_j) with $e_i \in G_i$, $e_j \in G_j$ is minimized to preserve diversity among them.

Formally, we define FSX as the problem of finding the faceted groups $\mathcal{G}$ that maximize the following function $f(\mathcal{G})$:

$$\alpha \sum_{\substack{\forall i \\ y \in G_i}} rel(q, y) \; + \; \beta \sum_{\substack{\forall i \\ y,z \in G_i}} rel(y, z) - \gamma \sum_{\substack{\forall ij,\, i \neq j \\ m \in G_i, n \in G_j}} rel(m, n) \tag{1}$$

$$\begin{aligned} \text{subject to} \quad & \forall G_i, |G_i| = k, \text{(bounded size of each group)} \\ & |\mathcal{G}| = l, \quad \text{(bounded number of groups)} \\ \text{and} \quad & \exists S_p, G_i \subseteq S_p \text{ (selecting groups from input facets).} \end{aligned}$$

Where α, β, γ are tunable parameters and rel denotes the similarity between pairs of entities. Due to the combinatorial structure of the problem with size constraints on the output groups, computing exact solutions of it is intractable.

Theorem 2.1. *FSX is NP-hard.*

Proof. We give a polynomial-time reduction from the Weighted Max-Coverage (*WMC*) problem [11] to a special configuration of *FSX*.

WMC problem: Consider a universe of elements U, a weight function $w : U \rightarrow \mathbb{R}_0^+$, a positive integer l, and a family of subsets $\mathcal{X} = \{X_1, X_2, \ldots X_m\}$ where each $X_i \in 2^U$. The objective is to find $\mathcal{X}' \subseteq \mathcal{X}$, where $|\mathcal{X}'| \leq l$ and the total weight of the covered elements, $x \in X_i$ for some $X_i \in \mathcal{X}'$, is maximized.

For the reduction, we consider the special *FSX* case of $|G_i| = \max_i |X_i|$ with hyperparameters $\beta = \gamma = 0$ and $\alpha = 1$. Each instance $(U, \mathcal{X}, w)$ of WMC is mapped to an instance $(E, \mathcal{S}, rel)$ of *FSX* by setting $E = U$, $\mathcal{S} = \mathcal{X}$, and $w(e) = rel(e, q)$. By this construction, $\mathcal{X}' \subseteq \mathcal{X}$ maximizes $\sum_{x \in X_i \text{ with } X_i \in \mathcal{X}'} w(x)$ with $|\mathcal{X}'| \leq l$ if and only if $S' \subseteq \mathcal{S}$ maximizes $\sum_{e \in S_i \text{ with } S_i \in S'} rel(e, q)$ with $|S'| \leq l$. As the general *FSX* problem is at least as hard as the special configuration, we have shown that *FSX* is NP-hard. □

The above hardness is mitigated by the observation that, following [22], the optimization function in Equation 1 satisfies the submodularity property, which allows us to explore efficient heuristic-based methods for achieving high-quality approximate solutions.

3 THE FASETS METHOD

In this section, we propose our iterative set expansion method, FaSets, that finds compact faceted groups for a query entity. It leverages KBs and Wikipedia to collect input set of labeled facets from which candidate entities with their descriptive labels are judicially chosen for a query entity. A candidate graph is then created based on the similarity between candidate entities. FaSets takes a greedy approach to generate faceted groups one after the other by a random-walk-based iterative algorithm over the candidate graph.

3.1 Input Set of Labeled Facets.

FaSets provides an explanation (label) for each faceted group. For generating those explanations, we collect descriptive labels or categorical features for candidate entities. We use two sources to gather input set of labeled facets:

1. KBs like YAGO, Wikidata etc., provide billions of subject-predicate-object (SPO) triples. We group subjects that share predicate-object (PO) pairs to create faceted sets, where POs denote the label of facets. For example, we can create a facet,

{Jeff Bezos, Malcolm Forbes, ...} with the label *graduatedFrom Princeton_University* based on the facts from YAGO.
2. Wikipedia provides more than 1 Million categories which naturally group entities into facets, e.g., *American billionaires: {Zuckerberg, Jeff Bezos, Elon Musk, ...}*. Additionally, infobox properties are used to group entities under the property-and-value label (analogous to SPO triples from KBs).

We treat this large pool of input facets as a *bipartite graph*. Entities and labels of facets become two different types of nodes in the graph, and an entity-node is linked to a facet-node if the entity belongs to the facet. Presumably, these entities and facets are not equally salient or informative. For example, the facet *living people* provides very general information about an entity, whereas the facet *American Billionaires* gives specific and more descriptive information about an entity. Therefore, we assign a saliency score to each node of the bipartite graph. In this work, we focus on the visibility of facets or entities as a proxy of their salience. Yet this is merely a pragmatic choice, other proxies for salience could be plugged in, too.

Saliency Score of Entity Nodes ($score_e$). We consider the page-views of the Wikipedia page for an entity as its saliency score, reflecting its visibility on the web. Page-views are considered a standard popularity measure in web-based information systems. For each page, we extract total page-views over a period of one month.

Saliency Score of Facet Nodes ($score_f$). To capture the saliency of a facet, the same approach is not feasible, as Wikipedia category pages are rarely visited directly. Instead, we consider the number of existing multilingual Wikipedia editions for category pages as a measure of facet saliency. For example, the Wikipedia category page for *American Billionaires* exists in 37 languages, like French, Portuguese, Romanian, etc. We also collect the facets derived from KBs or infoboxes, which are not connected to clickable Wikipedia pages. Therefore, we cannot directly retrieve the saliency score for these facets. In that case, we use multilingual Wikipedia editions for the Wikipedia page of the object in POs as their saliency score. For example, a facet collected for *Jeff Bezos* from Yago is *graduatedFrom Princeton_University*, and the saliency score for it becomes 91 because the Wikipedia page for Princeton University exists in 91 languages.

Both saliency scores, are dampened by log values and normalized between 0 and 1.

3.2 Relatedness between Entities

The proposed iterative approach operates on the candidate graph created based on similarity between entities. From the representation of the input facets as a bipartite graph mentioned earlier, we can express an entity by its facet memberships from the bipartite graph. For example, *Jeff Bezos* as {*graduatedFrom Princeton, type_Billionaires, type_ Businessmen, ...*}, considering there exists an edge between Jeff Bezos and those facets. Moreover, capturing the saliency of facets notes, we consider a better representation of entities by their notability-weighted group memberships, e.g., *Jeff Bezos: { graduatedFrom Princeton (0.63), type_Billionaires (0.65), type_ Businessmen (0.39), ...}*; and we use weighted-Jaccard similarity to define the relatedness between entities using their weighted-group memberships. However, this distributional similarity only captures saliency of facet noes and does not consider the salience of entities. As a result, Stephen Hawking is closer to a number of less notable "long-tail" physicists than to prominent ones such as Einstein or Feynman. Hence, we incorporate entity-proximity based on Wikipedia pageviews with the distributional similarity to define the relatedness of entities.

Definition 3.1. Relatedness Score (rel): Given two entities e_x and e_y, the relatedness score is calculated by the weighted average of the weighted-Jaccard similarity between their weighted-group memberships and the proximity based on their page-views. For entity e_x, the group-membership is represented by a vector of size m, denoted as $\hat{e}_x = [v_1, \ldots, v_m]$ where $v_i = score_f(S_i)$, if $e_x \in S_i$ otherwise $v_i = 0$. Then, $rel(e_x, e_y)$ is defined as:

$$w_1 \cdot \frac{\sum_i min(\hat{e}_{x_i}, \hat{e}_{y_i})}{\sum_i max(\hat{e}_{x_i}, \hat{e}_{y_i})} + w_2 \cdot (1 - |score_e(e_x) - score_e(e_y)|) \quad (2)$$

The parameters w_1 and w_2 control two components of the relatedness measure. We create the candidate graph for a query based on this similarity measure between candidate entities.

3.3 Iterative Algorithm to Find Faceted Groups

FSX considers a large pool of labeled facets as input. However, the query entity is typically related only to a tractable subset of facets in the collection. Hence, FaSets works on a subset of input facets w.r.t. the input query, and efficiently computes the desired number of faceted groups. It operates in two stages: 1) constructing a candidate graph for the input query and 2) computing the faceted output groups on the candidate graph.

Construction of Candidate Graph. We build a candidate graph for the query by selecting potential entities and facets that can form the faceted groups and provide their explainable labels. For this purpose, we explore the bipartite graph, starting from the query node and alternating between entity nodes and facet nodes using breadth-first search until we gather θ candidate entities. We include all facets that have an edge with these θ candidate entities in the bipartite graph as candidate facets for generating explanation labels for output faceted groups. Using these candidate facets we create the initial candidate graph G^1_{sim} for the candidate entities using the relatedness score, defined in definition 3.1.

Discovering the Faceted Groups. We propose a random-walk-based iterative approach on the candidate graph to find the output faceted groups and their descriptive label from candidate facets, presented in Algorithm 1. This algorithm runs multiple times. In each run, from the candidate pool, the proposed method finds the best group that is similar to the input query but different from the output groups from the earlier runs.

It starts with the input query q and generates the faceted group G_1 with the label from the facet S^* from the initial candidate graph G^1_{sim}. Based on the generated faceted group G_i in the i^{th} round, we update the candidate graph to G^{i+1}_{sim}, and repeat the proposed iterative algorithm on the updated graph to find the faceted group G_{i+1} in the next round.

Algorithm 1: Finding i^{th} faceted group G_i

Input : Query q, transition matrix M^i from candidate graph G^i_{sim} with candidate entities $E_c \subset E$, candidate facets $S_c \subset \mathcal{S}, k, \alpha, \beta, \gamma$
Output : i^{th} faceted group G_i
Initialize::
1 $t = 0, V_q, V_t$ =top-k entities e based on $rel(e, q)$, $V_p = \cup_{j<i} G_j$
2 **while** *True* **do**
3 compute V^i_{t+1}
4 $G^{t+1}_i \leftarrow$ top-k $e \in E_c$ based on $Q(e, q, G^{t+1}_i)$ from V^i_{t+1}
5 **foreach** $S_j \in S_c$ **do**
6 $Score(S_j) = 1/\log_2 |S_j| + |G^{t+1}_i \cap S_j|$
7 $S^* \leftarrow argmax_j \{Score(S_j)\}$
8 $G'_i \leftarrow$ top-k $e \in S^*$ based on $Q(e, q, G^{t+1}_i)$ from V^i_{t+1}
9 **if** $G^{t+1}_i \neq G'_i$ **then**
10 $e_a \leftarrow argmin_i \{Q(e, q, G^t_i) | e \in G^t_i\}$
11 $e_b \leftarrow argmax_i \{Q(e, q, G^{t+1}_i) | e \in G'_i \setminus G^t_i\}$
12 $G^{t+1}_i \leftarrow \{G^t_i \setminus e_a \cup e_b\}$
13 **else**
14 Break *while* loop
15 **return** G'_i, S^*

In the proposed iterative approach of FaSets, we consider the candidate graph for i^{th} round G^i_{sim} as the transition probability matrix M^i where the edge weight between entities reflects the probability to jump from one entity to another, and the walk starts with the query node. However, unlike random walk, the propagation through the graph is influenced by only top-k prominent entities from the previous iteration and the entities in the faceted groups from earlier rounds. Let us consider V_q, V_t, V_p are three vectors, representing respectively the query entity, top-k prominent entities G^t_i based on the entity score from t^{th} iteration, and entities that form faceted groups in preceding runs ($e \in G_x, x < i$). Then the score of all entities in the candidate graph for $(t+1)^{th}$ iteration is calculated as follows:

$$V^i_{t+1} = \alpha M^i V_q + \beta M^i V^i_t - \gamma M^i V_p \tag{3}$$

From the Equation 3, we find that the score an entity e for $(t+1)^{th}$ iteration $Q(e, q, G^{t+1}_i)$ is a combination of three components:

- the similarity score to q: $\alpha * rel(e, q)$;
- the coherence score to top-k selected entities G^t_i from t^{th} iteration: $\beta * \sum_{e_j \in G^t_i} rel(e, e_j)/|G^t_i|$;
- a penalty based on the similarity of e to previously found faceted groups: $\gamma * \sum_{e_j \in \cup_{j<i} G_j} 1 - rel(e, e_j)/|\cup_{j<i} G_j|$.

Clearly, the way we normalize the entity score for each iteration also deviates from the traditional random walk process. We select top-k entities G^{t+1}_i based on V_{t+1} to continue the walk. Intuitively, it helps us to propagate the score only through the confident nodes.Additionally, to ensure the third constraint in FSX (Equation 1), which is preserving the structure of input facets in the output faceted groups, FaSets performs an additional step before continuing with the next iteration. It calculates a representative score for each candidate facet based on the entities in G^{t+1}_i and the size of the facet, presented in Line 6 in Algorithm 1. Finally, it chooses the best candidate facet according to this representative score and extracts top-k entities G'_i according to the entity score from V^i_{t+1}. The iteration stops if the top-k entities G'_i from the best candidate facets S^* remain the same as G^{t+1}_i found by Equation 3 at $(t+1)^{th}$ iteration, and the algorithm returns G'_i as the output group for this round. Otherwise, we modify the V_t vector by replacing the entry corresponding to the least-scored entity of G^t_i from t^{th} iteration with the highest-scored entity from G'_i from $(t+1)^{th}$ iteration based on V^i_{t+1}, and continue to the successive iteration.

After generating the faceted group G_i in i^{th} run, we update the candidate graph G^i_{sim} by penalizing the relatedness-score of entities from G_i to other candidates, in order to enforce the diversity among the output faceted groups. To do so, We find the candidate facets S_i that include *all* entities in the faceted group G_i, and remove the edges between them from the bipartite graph. Consequently, the weighted-group membership vectors $\hat{e}_x$ for the entities $e_x \in G_i$ are updated by assigning $score_f(S_i) = 0$. As a result, the relatedness score between e_x and other candidate entities changes, and the candidate graph is modified accordingly to G^{i+1}_{sim} for the computation of faceted group in the next run.

To generate l faceted groups, the iterative algorithm of FaSets runs l consecutive rounds. In each of these rounds, the algorithm iterates over the candidate graph until top-k entities based on the score computed by Equation 3 converge. Hence, to prove the convergence of FaSets, w.l.o.g., it is sufficient to show that the iterative process converges for finding each output group. For this purpose, the algorithm needs to ensure that the top-k entities based on their scores remain at the same after a finite number of iterations.

Theorem 3.2. *The iterative algorithm in* FaSets *converges.*

Proof. Consider the i^{th} round of iterative algorithm, where $E_p = \bigcup_{j<i} G_j$ represents the entities from the faceted groups generated by previous rounds. The score of an entity e at i^{th} round for query q is denoted by $Q(e, q, G_i)$.

$$Q(e, q, G_i) = \alpha \cdot rel(e, q) + \frac{\beta}{k} \cdot \sum_{x \in G_i} rel(e, x) - \frac{\gamma}{|E_p|} \cdot \sum_{x \in E_p} rel(e, x)$$

To prove the convergence of the iterative algorithm 1, we need to show that the aggregated score of the i^th output group G_i, represented as $Q(G_i, q) = \sum_{e \in G_i} Q(e, q, G_i)$, increases monotonically in each iteration.

Let us consider, at t^{th} iteration the output group $G^t_i = \{p_1, p_2, \dots p_k\}$ and at $t + 1^{th}$ iteration $G^{t+1}_i = \{p'_1, p'_2, \dots p'_k\}$. Our iterative algorithm replaces one element at each subsequent iteration. Here, without loss of generality, we can say $p_i = p'_i, \forall i \neq k$. Therefore,

$$Q(G^{t+1}_i, q) - Q(G^t_i, q) = \alpha \cdot (rel(p'_k, q) - rel(p_k, q)) + \frac{\beta}{k} \cdot \Big(\sum_{x \in G^{t+1}_i \setminus p'_k} rel(p'_k, x) - \sum_{x \in G^t_i \setminus p_k} rel(p_k, x) \Big)$$

$$- \frac{\gamma}{|E_p|} \cdot \Big(\sum_{x \in E_p} rel(p'_k, x) - \sum_{x \in E_p} rel(p_k, x) \Big)$$

$$\geq \quad \alpha \cdot (rel(p'_k, q) - rel(p_k, q)) + \frac{\beta}{k} \cdot \Big(\sum_{x \in G_i^t} rel(p'_k, x) - \sum_{x \in G_i^t} rel(p_k, x) \Big) - \frac{\gamma}{|E_p|} \cdot \Big(\sum_{x \in E_p} rel(p'_k, x) - \sum_{x \in E_p} rel(p_k, x) \Big)$$

$$\text{as } [rel(p_k, p_k) \geq rel(p'_k, p_k)]$$

$$\Rightarrow \quad Q(G_i^{t+1}, q) - Q(G_i^t, q) \geq Q(p'_k, q, G_i^t) - Q(p_k, q, G_i^t)$$

According to the replacement strategy of the proposed algorithm, the lowest scored entity $p_k \in G_i^t$ will be replaced by highest scored entity $p'_k \in U \backslash G_i^t$ when the top-k elements in G_i^{t+1} differs. Hence,

$$Q(p'_k, q, G_i^t) > Q(p_k, q, G_i^t) \Longrightarrow Q(G_i^{t+1}, q) > Q(G_i^t, q)$$

Since, $G_i \subseteq E \backslash E_p$, the possible combinations of k entities that can form G_i is $\binom{|E \backslash E_p|}{k}$. As the aggregated score for G_i increases monotonically with iterations, the total number of replacement of entity is bounded by $\binom{|E \backslash E_p|}{k}$. Hence, the algorithm converges. □

4 EVALUATION

4.1 Datasets and Setup

We collected a real-world dataset from three domains:

- People: 50k popular persons, associated with 65k facets.
- Movies: 50k popular movies, associated with 54k facets.
- Companies: 5k companies, associated with 33k facets.

All the entities in our datasets are collected from Wikipedia and also exist in the YAGO, which enable us to collect the saliency score of facets and entities from Wikipedia. For these entities, we collected 73k facets from Wikipedia categories and 81k facets from YAGO facts. In our experiments, we discard the facets with less than five entities, and also prune 20 overly generic facets, like wordnet_Physical_entity, owl_things.

We compiled benchmark queries with 40 popular people (e.g., Stephen Hawking, John Lennon, etc.), 40 popular movies (e.g., Toy Story, The Matrix, etc.), and 20 companies (e.g., Nokia, IBM, etc.).

We set the threshold θ to 2000 for the construction of candidate graph based on the observation that, the average number of candidate entities, collected from traversing three hops in the bipartite graph for randomly selected queries, is approx 2000.

We conducted experiments on a Linux server with Intel Xeon(R) CPU (32 cores@3.20GHz) and 500 GB RAM. We choose the parameters $\alpha = \beta = 0.3, \gamma = 0.4$ based on a small training dataset, and consider $w_1 = w_2 = 0.5$ for the relatedness-score.

4.2 Baselines

We compare FaSets against the following baselines, and all of them operate on the same candidate graph as FaSets.

- *SEISA [14]:* This method expands seed entities by preserving the similarity and coherence property. As FaSets outputs multiple faceted groups, we extend SEISA by running it l (#groups) times while updating the candidate graph analogously to FaSets after each iteration, to enforce diversity.

Table 1: Gold-standard groups for Nelson Mandela.

Socialists	President of South Africa	Revolutionists
Karl Marx	Thabo Mbeki	Che Guevara
Noam Chomsky	P. W. Botha	Vladimir Lenin
Leon Trotsky	Jacob Zuma	Malcolm X
George Galloway	F. W. de Klerk	Mahatma Gandhi
George Orwell	Kgalema Motlanthe	Leon Trotsky

- *Random Walk with Restart (RW):* This method approximates stationary visiting probabilities, and returns the group and entities with the highest ranks. To output multiple groups, the method is run repeatedly (l times), with removal of previously returned entity nodes after each iteration, to enforce diversity. Each round performs 300 iterations with error tolerance 0.001 and restart probability 0.15.
- *EgoSet [25]:* This is a graph-based set expansion method that considers a seed can belong to multiple classes. In our context, instead of using skip-grams from text, we use group memberships of entities as features to construct the graph for the query and cluster it. We collected tables and lists from Wikipedia pages where the title contains the word 'List'. The previously generated clusters are then refined by the membership overlap in these tables and lists and the similarity based on wikipedia2vec embeddings [38] of clustered entities, to produce disjoint output clusters.
- *FUSE [40]:* This is a corpus-based multi-faceted set expansion approach that uses embedding features of coherent contexts from skip-grams to expand the seed entities using masked-language-model(MLM) from BERT. In our context, we use group memberships of entities as context.

4.3 Ground Truth

We compare FaSets against the baselines primarily using the pooling technique: i) obtain top-ranked results from all methods under comparison to form a result pool, ii) use crowdsourcing (AMT) to assess all results in the pool, iii) compute quality measures for all methods based on this ground truth.

In addition, we obtained a-priori gold-standard groups for all benchmark queries, also by crowdsourcing (AMT), but independently of the results computed by FaSets or the baselines.

First, we generate 10 diverse and informative facets for each query. To this end, we combine top-20 candidate facets based on four simple scoring functions: 1) the size of a facet 2) $max(score_e(e))$ for the entities from a facet, 3) $avg(score_e(e))$ for the entities from a facet, and 4) $score_f$ for a facet. The combined list of facets is shown to five annotators who are asked to select five diverse and informative facets. The choices of the five annotators are aggregated, to select the top-10 frequently chosen facets as gold-standard groups. For these gold facets, we obtained a perfect Fleiss-kappa agreement of value 1. In total, 104 different annotators performed this task.

After gathering the labels for gold-standard groups, we generate the group of entities for the collected facets for each query. For each facet, we identified the 20 closest entities to the query based on four simple scoring functions: 1) number multilingual Wikipedia editions that feature the entity, 2) length of the English Wikipedia article for the entity, and 3) number of SPO triples for the entity

in YAGO and 4) number of pageviews of the Wikipedia article. We merged these lists and showed top-30 entities to five annotators. They were asked to choose 5 similar entities to the query from the collected list. Finally, the most frequently selected entities are taken for the gold standard. In total, 337 annotators performed this task, and we obtained a moderate inter-annotator agreement with a Fleiss-kappa value of 0.51. Table 1 presents an example of three gold-standard groups with top-5 entities for *Nelson Mandela*.

4.4 Evaluation Metric

Suppose an algorithm generates output $\mathcal{G}$ with l groups with k entities each, and we have ground truth $\mathcal{GT}$: m groups with n entities each. We define the quality of the algorithm against the ground truth, $Quality@l.k$, as follows.

Conceptually, we consider all mappings between the output groups $\mathcal{G}$ and the ground-truth groups $\mathcal{GT}$. For each mapping $\mathcal{G} \rightarrow \mathcal{GT}$, we calculate the *Quality* by averaging the precision for each group in $\mathcal{G}$ against its mapped group in $\mathcal{GT}$:

$$Quality@l.k = \frac{\sum_{g_i \rightarrow gt_j} \frac{|g_i \cap gt_j|}{k}}{l} \quad i \in l, j \in m$$

From all possible mappings, we select the one where the Quality metric is maximal, and we present $Quality@l.k$ for this mapping.

We also use the B-cubed measure [3], a standard metric for evaluating co-reference resolution by clustering. B-cubed computes the the weighted average of the per-entity precision scores over all output groups, with precision defined as the fraction of correct elements in an output group containing the entity.

4.5 Intrinsic Evaluation of FaSets

4.5.1 Evaluation of faceted groups using pooling. We gather output groups from all methods and ask crowdsourcing workers for assessments. As there is no restriction on the number of entities in the output clusters generated by EgoSet, we select the top-k entities, based on the entity-saliency scores, as cluster representatives for a fair comparison. As FUSE uses Masked-language-model to generate expanded entities for the coherent cluster representatives, it often includes out-of-domain entities in the output clusters, e.g., the genre Animie becomes an extended entity for the movie *Rango*. Therefore, we filter out such out-of-domain entities and consider top-k entities from the given query-domain as an output cluster. For the crowdsourcing task, we show the output groups generated by FaSets and all baselines for a given query, and ask three annotators to rate each result by one of the three labels {*bad, moderate, good*} reflecting the coherence within each group and diversity across groups. We map the three labels to {0, 0.5, 1}, and calculate the average score for each method. Table 2 shows the results for the benchmark queries from each domain, for different numbers of groups l. FaSets outperforms all baselines by a large margin, consistently across domains. FUSE uses the affinity propagation algorithm to cluster the context features and automatically finds the number of clusters using the exemplars from the input data. According to our datasets, FUSE has not generated five output groups, and therefore, evaluation for $l = 5$ is kept empty for FUSE.

4.5.2 Evaluation of faceted groups using gold-standard groups. We also report the quality of output groups from different methods w.r.t. gold-standard groups using two different evaluation metrics in Table 3 by varying the number of output groups l and entities per group k. For EgoSet and FUSE we consider the top-k entities based on the entity-saliency scores from a cluster as the output group. As mentioned earlier, for FUSE, we consider top-k expanded entities from the query domain (people/movies/companies) as an output cluster.

Table 2: Evaluation of faceted groups using pooling.

Domain	l	k	FaSets	SEISA	RW	EgoSet	FUSE
People	3	5	**0.73**	0.49	0.25	0.46	0.25
	5	5	**0.70**	0.64	0.53	0.55	-
Movies	3	5	**0.74**	0.69	0.67	0.60	0.35
	5	5	**0.63**	0.46	0.62	0.47	-
Companies	3	5	**0.65**	0.53	0.49	0.49	0.55
	5	5	**0.58**	0.48	0.45	0.50	-

From Table 3, we can observe that FaSets outperforms the baselines with a large margin. The random walk performs poorly, which aligns with the evaluation using pooling method mentioned in Table 2. Even though FUSE and EgoSet use a similar clustering-based approach, EgoSet performs better than FUSE as the extended entities are refined using Wikipedia lists. As mentioned before, FUSE could not generate five output groups with our dataset, and therefore, evaluation for $l = 5$ is kept empty for FUSE. We also find that the results for People are the weakest for FaSets. This can be attributed to the much larger number of input facets for people, covering highly diverse sub-types such as politicians, athletes, musicians, scientists, etc. The Character of different metrics is also reflected in the results. Quality@l.k metric uses the best alignment between faceted groups and the ground truth. As a result, the quality value decreases with the increasing number of output groups. On the other hand, the performance differs as the number of entities per group changes for the B-cubed metric as B-cubed reflects entity-level precision.

4.5.3 Evaluation of explanation labels for faceted groups. Additionally, we evaluate the labels of faceted groups in Table 4. As none of the baselines provide labels for the output groups, we post-process their output groups and generate labels from the most specific facets where all the entities of an output group appear. FUSE expands the entities using MLM from BERT. As a result, many of them are out of our dataset. Hence, we are unable to produce the labels for FUSE, and we omit the evaluation of FUSE for label matching. The metric used for this evaluation is the number of exactly matching labels produced by a method for a query against the set of labels for gold-standard groups for the query. Table 4 shows that FaSets clearly wins over the baselines. It inclines to produce more specific labels to general ones as the number of output groups increases. Due to this characteristic, the overlap with the ground truth facets decreases as the number of groups increases. This pattern deviates for the people domain because the gold-standard labels are often general in this domain.

4.6 Extrinsic Evaluation

We design an *intruder task* to evaluate the coherence of our faceted groups against baselines. We collected 40 groups people, 30 groups

Table 3: Evaluation of l faceted groups with k entities using gold-standard groups.

			Quality@l.k					B-cubed				
Domain	l	k	FaSets	SEISA	RW	EgoSet	FUSE	FaSets	SEISA	RW	EgoSet	FUSE
People	3	5	**0.25**	0.11	0.01	0.20	0.06	**0.18**	0.05	0.001	0.02	0.01
	5	5	**0.16**	0.07	0.01	0.15	-	**0.18**	0.06	0.002	0.02	-
	3	10	**0.19**	0.08	0.01	0.12	0.05	**0.06**	0.01	0.001	0.01	0.01
	5	10	**0.13**	0.05	0.01	0.08	-	**0.06**	0.01	0.001	0.01	-
Movies	3	5	**0.42**	0.28	0.07	0.01	0.01	**0.44**	0.20	0.01	0.05	0.003
	5	5	**0.31**	0.18	0.04	0.01	-	**0.44**	0.22	0.01 4	0.05	-
	3	10	**0.29**	0.20	0.06	0.05	0.01	**0.20**	0.08	0.007	0.04	0.002
	5	10	**0.22**	0.13	0.04	0.01	-	**0.21**	0.06	0.006	0.04	-
Companies	3	5	**0.63**	0.55	0.07	0.17	0.1	**0.93**	0.69	0.01	0.07	0.11
	5	5	**0.49**	0.40	0.04	0.10	-	**0.90**	0.79	0.01	0.05	-
	3	10	**0.50**	0.40	0.06	0.11	0.07	**0.52**	0.33	0.005	0.05	0.04
	5	10	**0.39**	0.30	0.04	0.06	-	**0.53**	0.38	0.006	0.03	-

Table 4: Comparison of group labels w.r.t. gold-standard.

Domain	l	FaSets	SEISA	RW	EgoSet
People	3	0.51	0.41	0.36	**0.59**
	5	**0.78**	0.65	0.60	0.37
Movies	3	**0.69**	0.50	0.36	0.26
	5	**0.43**	0.29	0.21	0.15
Companies	3	**0.86**	0.60	0.03	0.05
	5	**0.55**	0.33	0.02	0.03

Table 5: Evaluation based on the intruder task.

Domain	FaSets	SEISA	RW	EgoSet	FUSE
Movies	**0.4**	0.27	0.14	0.30	0.30
People	**0.73**	0.66	0.3	0.62	0.60
Companies	0.63	0.6	0.21	**0.70**	0.60
Overall	**0.59**	0.51	0.22	0.54	0.53

of movies, and 30 groups of companies for FaSets and the baselines. These groups contain four entities from an output group generated by a method and one random entity from the same domain that acts as an intruder. Specifically, we consider 50 queries and two output groups per query in the evaluation process. For each group, We ask three annotators to pick the entity that should not belong to the given group. Table 5 reports the correct intruder detection rate based on majority voting in percentage. Overall, the annotators detect the correct intruder 59% of the time for FaSets. For this task, an annotator has to be familiar with cast members for many movies from different genres. Due to this characteristic, we reach a lower intruder detection rate in the movies domain than others. We also observe that, all the methods, FaSets, SEISA, EgoSet, and FUSE that include a coherence component in the approach, perform significantly better than other as expected. A total of 201 annotators evaluated this task, and we obtain a fair inter-annotator agreement with the Fleiss' Kappa value of 0.31.

The intruder task alone cannot evaluate the diversity among the faceted groups for a query, which is one of the components of FSX problem. Therefore, to assess the quality of diverse groups generated by the algorithm, we sample random 30 queries and present generated labels for five faceted groups per query by FaSets against the post-processed group labels for baselines. We assigned each task to three annotators and asked them to choose the method that presents more informative and diverse group labels for the query. FaSets were preferred in 77% of the cases over those of the baseline methods. A total of 55 annotators evaluated this task, and we achieve a moderate inter-annotator agreement with the Fleiss' Kappa value of 0.47.

4.7 Qualitative Discussion with Examples

Table 6 presents an anecdotal example of three faceted groups with five entities for *Max Planck*. We can see that FaSets discovers three diverse groups with informative labels, whereas other baselines fail. SEISA provides coherent faceted groups but does not guarantee the diversity among the groups, even though we use the similar modification in the graph to enforce diversity in the iterative approach analogous to FaSets. Additionally, FaSets uses saliency score for entity and facet nodes in the graph, which affects positively in finding similarity between entities. As a result, FaSets found more coherent faceted group than SEISA, e.g., *German Physicists vs. Physicists*. Random Walk suffers badly from concept drifting, and consequently, output groups can only be described under very general concept, such as *Person*. EgoSet chooses cluster representatives based on saliency. As a result, the output groups have many popular entities but lose specificity and are vulnerable to concept drift. EgoSet uses Wikipedia tables to enforce coherence, but the diverse output groups connect to general lists from the domain. As the categories for a seed entity can be easily expressed under a generalized context, FUSE fails to generate multiple refined coherent semantic clusters. As a result, it suffers from semantic drift while expanding seeds.

We also observe that FaSets generates the label for the first faceted group from Wikipedia categories in 85% of the cases in people and companies domain, whereas 75% of the labels for the first faceted group comes from infobox properties in movies. This reflects the characteristic of the dataset. For movies, Wikipedia categories do not cover the cast-oriented small groups that are captured by infobox properties. Overall, for the top-5 groups, 61% of labels are generated from Wikipedia categories.

5 RELATED WORK

Web-based set expansion. Using web-based search engines, Google Set [31], SEAL [36], LYRETAIL [7] access a large set of corpora

Table 6: Anecdotal example for three faceted groups with five entities for Max Planck.

Methods	Faceted Groups	Explanation
FaSets	Max von Laue, Philipp Lenard, Arnold Sommerfeld, Max Born, Werner Heisenberg	German physicists
	C. V. Raman, Abdus Salam, Erwin Schrödinger, Niels Bohr, Paul Dirac	Nobel laureates in physics
	Max Born, Werner Heisenberg, Wolfgang Pauli, Niels Bohr, Arnold Sommerfeld	Quantum physicists
SEISA	Max Born, Niels Bohr, Abdus Salam , Max von Laue, Arnold Sommerfeld	Physicists
	Philipp Lenard, Carl Bosch, Robert Bunsen, Hans Geiger, Rudolf Clausius	Ethnic German people
	Arnold Sommerfeld, Max von Laue, Max Born, Abdus Salam, Niels Bohr	Physicists
RW	Lew Ayres, Harald Quandt, Philip Pullman, Clarence Darrow, James Tobin	Person
	Max Riemelt, Joseph Stiglitz, Traudl Junge, Zeena Schreck, Li Si	Person
	Ethan Allen, Sinclair Lewis, Franklin Graham, Rachel Corrie, Leonard Adleman	Person
EgoSet	Martin Luther, Charles Darwin, Aristotle, Galileo Galilei, Isaac Newton, Leonardo da Vinci	Scientists
	Richard Feynman, Isaac Newton, Leonardo da Vinci, Karl Marx, Albert Einstein	Intellectuals
	Carl Sagan, Kurt Vonnegut, Michael Crichton, H. P. Lovecraft, Stan Lee	Writers
FUSE	William Armstrong, Oscar Magoni, Douglas Hodkin, Arthur McMaster, Piereson	-
	Dean Mcgraw, John Cook, William Armstrong, Oscar Magoni, Milkovisch	-
	Micheal S. Berman, Gerald Malloy, Christina Hale, Neal Zimmers, Eldon Nygaard	-

which are exploited to extend the seeds. By selectively marking entities in Web pages, SEAL [36] and iSEAL [37] build a directed graph and use the random-walk-based method to rank the entities. LYRETAIL [7] extends long-tail queries from a single page by a supervised page-specific extractor. The main drawback of these methods is the dependency on online web applications, which can lead to noisy data collection and increases query time as well.

Corpus-based set expansion. Most of the recent set-expansion systems use offline resources of specific type (such as text [6, 15], Web tables [35]) or heterogeneous corpora (text and Wikipedia tables [25]). FaSets exploits Wikipedia categories and KB facts, but it is equally applies to other inputs.

Addressing the efficiency aspect of seed expansion on a large domain-specific corpus, one-time ranking methods are explored in [12, 39]. Ghahramani and Heller [12] propose a probabilistic ranking model based on Bayesian inference that reflects the relevance of a candidate entity to a cluster, containing the seeds. CaSE [39] presents a ranking method by combining lexical features from skip-grams and distributional representation from learned embedding of candidate entities. Many systems [2, 27, 35] use iterative pattern-based bootstrapping where seeds are extended based on refined context features collected in each iteration. This iterative approach is prone to concept drifting due to ambiguous input seeds or intrusion of noisy patterns or entities. To tackle the concept drifting, SEISA [14] adopts an additional component of coherence with the extended group in the formalization of the set expansion problem. We further generalize the problem of multi-faceted set expansion in this work and consider an extended version of SEISA as one of the baselines. Wang et al. [35] identify relevant concepts to the seed entities using web tables and preserve the coherence of extended seeds by restricting them to the identified concepts. SetExpan [27] deals with semantic drift by refining skip-gram features in each iteration and select extended entities via rank-ensemble. Generating auxiliary sets of entities during expansion and using them as negative concepts, Set-coExpan [15] restricts concept drifting. Similarly, by manually introducing negative examples, a boundary for the target semantic class is set in [17, 29]. To handle the concept drifting with minimum supervision, few works [6, 13, 21] use word-embeddings in defining similarity between entities. ProbExpan [20] uses contrastive learning to find a better representation of entities belonging to a semantically similar class and tackle semantic drift.

Multi-faceted set expansion. The literature mentioned above consider the seeds belong to a single target concept, and therefore, they explore different methods to capture accurate patterns for the target concept. To deal with noisy or multi-faceted seeds, many works cluster candidate entities to discover different concepts [5, 18, 19, 25]. Rong et al. [25] propose a framework called EgoSet that uses variable-length skip-grams to build an ego net for the seed and cluster them into multiple communities. Those communities are then further refined based on Wikipedia-lists-memberships and word-embedding of candidates. A similar approach considered in FuSE [40]. First, skip-grams of seed entities are clustered based on their embedding space to find coherent contextual features. Later, the representations of resulted coherent clusters of skip-grams are used to extend the seed entities based on the masked-language-model of BERT. A few recent works also use cluster-based approaches for topic discovery and expansion [16, 24] using embedding space of a concept along with its representative terms. Our formulation of *FSX* is closest to SEISA, but we did not consider a single target class. In that sense, our problem description is similar to EgoSet and FUSE. As we enforce diversity among extended groups, *FSX* also relates to the diversification of results in web search engines [1, 9, 41] or recommender systems [8, 32, 33].

6 CONCLUSION

This work proposes FaSets, an iterative set expansion method to discover a compact set of *explainable* faceted groups related to a given entity, starting from a pool of thousands of candidate sets. FaSets is a potential asset for interactive data exploration, guiding advanced users to better understand online contents with noisy category and tagging systems. It is applicable, for example, to hashtags as facets of social media posts, to large product catalogs, and potentially even structured but highly heterogenous "data lakes" with extensive coverage of entities and rich categorical attributes.

REFERENCES

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *WSDM*.

[2] Enrique Alfonseca, Marius Pasca, and Enrique Robledo-Arnuncio. 2010. Acquisition of instance attributes via labeled and related instances. In *SIGIR*.

[3] Amit Bagga and Breck Baldwin. 1998. Algorithms for Scoring Coreference Chains. In *LREC*.

[4] Vevake Balaraman, Simon Razniewski, and Werner Nutt. 2018. Recoin: Relative Completeness in Wikidata. In *Wiki Workshop*.

[5] Ramnath Balasubramanyan, Bhavana Bharat Dalvi, and William W. Cohen. 2013. From Topic Models to Semi-supervised Learning: Biasing Mixed-Membership Models to Exploit Topic-Indicative Features in Entity Clustering. In *ECML*.

[6] Matthew Berger, Ajay Nagesh, Joshua A. Levine, Mihai Surdeanu, and Helen Zhang. 2018. Visual Supervision in Bootstrapped Information Extraction. In *EMNLP*.

[7] Zhe Chen, Michael J. Cafarella, and H. V. Jagadish. 2016. Long-tail Vocabulary Dictionary Extraction from the Web. In *WSDM*.

[8] Peizhe Cheng, Shuaiqiang Wang, Jun Ma, Jiankai Sun, and Hui Xiong. 2017. Learning to Recommend Accurate and Diverse Items. In *WWW*.

[9] Marina Drosou and Evaggelia Pitoura. 2010. Search result diversification. *SIGMOD Record* (2010).

[10] Xiaobin Fu, Jay Budzik, and Kristian J Hammond. 2000. Mining navigation history for recommendation. In *IUI*.

[11] M. R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman.

[12] Zoubin Ghahramani and Katherine A. Heller. 2005. Bayesian Sets. In *NIPS*.

[13] Sonal Gupta and Christopher D. Manning. 2015. Distributed Representations of Words to Guide Bootstrapped Entity Classifiers. In *NAACL HLT*.

[14] Yeye He and Dong Xin. 2011. SEISA: set expansion by iterative similarity aggregation. In *WWW*.

[15] Jiaxin Huang, Yiqing Xie, Yu Meng, Jiaming Shen, Yunyi Zhang, and Jiawei Han. 2020. Guiding Corpus-based Set Expansion by Auxiliary Sets Generation and Co-Expansion. In *WWW*.

[16] Jiaxin Huang, Yiqing Xie, Yu Meng, Yunyi Zhang, and Jiawei Han. 2020. CoRel: Seed-Guided Topical Taxonomy Construction by Concept Learning and Relation Transferring. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 1928–1936.

[17] Prateek Jindal and Dan Roth. 2011. Learning from Negative Examples in Set-Expansion. In *ICDM*.

[18] Weize Kong and James Allan. 2013. Extracting query facets from search results. In *SIGIR*.

[19] Chengkai Li, Ning Yan, Senjuti Basu Roy, Lekhendro Lisham, and Gautam Das. 2010. Facetedpedia: dynamic generation of query-dependent faceted interfaces for wikipedia. In *WWW*.

[20] Yinghui Li, Yangning Li, Yuxin He, Tianyu Yu, Ying Shen, and Hai-Tao Zheng. 2022. Contrastive Learning with Hard Negative Entities for Entity Set Expansion. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*. 1077–1086.

[21] Jonathan Mamou, Oren Pereg, Moshe Wasserblat, Ido Dagan, Yoav Goldberg, Alon Eirew, Yael Green, Shira Guskin, Peter Izsak, and Daniel Korat. 2018. SetExpander: End-to-end Term Set Expansion Based on Multi-Context Term Embeddings. In *COLING*, Dongyan Zhao (Ed.).

[22] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions-I. *Mathematical Programming* (1978).

[23] Natalya Fridman Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM* (2019).

[24] Masayo Ota, Heiko Mueller, Juliana Freire, and Divesh Srivastava. 2020. Data-Driven Domain Discovery for Structured Datasets. *Proc. VLDB Endow.* 13, 7 (2020), 953–965.

[25] Xin Rong, Zhe Chen, Qiaozhu Mei, and Eytan Adar. 2016. EgoSet: Exploiting Word Ego-networks and User-generated Ontology for Multifaceted Set Expansion. In *WSDM*.

[26] Jiaming Shen, Wenda Qiu, Jingbo Shang, Michelle Vanni, Xiang Ren, and Jiawei Han. 2020. SynSetExpan: An Iterative Framework for Joint Entity Set Expansion and Synonym Discovery. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. 8292–8307.

[27] Jiaming Shen, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, and Jiawei Han. 2017. SetExpan: Corpus-Based Set Expansion via Context Feature Selection and Rank Ensemble. In *ECML*.

[28] Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Trans. Knowl. Data Eng.* (2015).

[29] Bei Shi, Zhenzhong Zhang, Le Sun, and Xianpei Han. 2014. A Probabilistic Co-Bootstrapping Method for Entity Set Expansion. In *COLING*.

[30] F. Suchanek, G Kasneci, , and G. Weikum. 2007. Yago: A Core of Semantic Knowledge. In *WWW*.

[31] Simon Tong and Jeff Dean. 2008. Systen and Methods for Automatically Creating Lists. In *US Patent 7350187*.

[32] Saúl Vargas, Linas Baltrunas, Alexandros Karatzoglou, and Pablo Castells. 2014. Coverage, redundancy and size-awareness in genre diversity for recommender systems. In *RecSys*.

[33] Saul Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *RecSys*.

[34] D. Vrandečić and M. Krötzsch. 2014. Wikidata: A Free Collaborative Knowledge base. *CACM* (2014).

[35] Chi Wang, Kaushik Chakrabarti, Yeye He, Kris Ganjam, Zhimin Chen, and Philip A. Bernstein. 2015. Concept Expansion Using Web Tables. In *WWW*.

[36] Richard C. Wang and William W. Cohen. 2007. Language-Independent Set Expansion of Named Entities Using the Web. In *ICDM*.

[37] Richard C. Wang and William W. Cohen. 2008. Iterative Set Expansion of Named Entities Using the Web. In *ICDM*.

[38] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia. In *EMNLP*.

[39] Puxuan Yu, Zhiqi Huang, Razieh Rahimi, and James Allan. 2019. Corpus-based Set Expansion with Lexical Features and Distributed Representations. In *ACM SIGIR*.

[40] Wanzheng Zhu, Hongyu Gong, Jiaming Shen, Chao Zhang, Jingbo Shang, Suma Bhat, and Jiawei Han. 2020. FUSE: Multi-faceted Set Expansion by Coherent Clustering of Skip-Grams. In *ECML PKDD*.

[41] Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. 2014. Learning for search result diversification. In *SIGIR*.