Abstract—Set expansion aims to expand a small set of seed entities into a complete set of relevant entities. Most existing approaches assume the input seed set is unambiguous and completely ignore the multi-faceted semantics of seed entities. As a result, given the seed set ("Canon", "Sony", "Nikon"), previous methods return one mixed set of entities that are either Camera Brands or Japanese Companies. In this paper, we study the task of multi-faceted set expansion, which aims to capture all semantic facets in the seed set and return multiple sets of entities, one for each semantic facet. We propose an unsupervised framework, FUSE, which consists of three major components: (1) facet discovery module: identifies all semantic facets of each seed entity by extracting and clustering its skip-grams, and (2) facet fusion module: discovers shared semantic facets of the entire seed set by an optimization formulation, and (3) entity expansion module: expands each semantic facet by utilizing an iterative algorithm robust to skip-gram noise. Extensive experiments demonstrate that our algorithm, FUSE, can accurately identify multiple semantic facets of the seed set and generate quality entities for each facet.

Index Terms—Set Expansion, Multi-facetedness, Word Sense Disambiguation

I. INTRODUCTION

The task of set expansion is to expand a small set of seed entities into a more complete set of relevant entities. For example, to explore all Universities in the U.S., one can feed a seed set (e.g., {“Stanford”, “UCB”, “Harvard”}) to a set expansion system and then expect outputs such as “Princeton”, “MIT” and “UW”. Those expanded entities can benefit numerous entity-aware applications, including query suggestion [2], taxonomy construction [32], and information extraction [12], [23], [56], [38]. Besides, the set expansion algorithm itself becomes a basic building block of many NLP-based systems [15].

Previous studies on set expansion focus on returning one single set of most relevant entities. Methods have been developed to incrementally and iteratively add the entities of high confidence scores into the set. A variety of features are extracted, including word co-occurrence statistics [18], unary patterns [26], or coordinational patterns [23], from different data sources such as query log [31], web table [33], and raw text corpus [15], [26]. However, all these methods assume the given seed set is unambiguous and completely ignore the multi-faceted semantics of seed entities. As a result, given a seed set {“apollo”, “artemis”, “poseidon”} which has two semantic facets – Major Gods in Greek Mythology and NASA Missions, previous methods can only generate one mixed set of entities from these two facets, which inevitably hampers their applicabilities.

In this paper, we approach the set expansion task from a new angle. Our study focuses on multi-faceted set expansion which aims to identify semantic facets shared by all seed entities and return multiple expanded sets, one for each semantic facet. The key challenge lies in the discovery of shared semantic facets from a seed set. However, the only initial attempt towards multi-facetedness, EgoSet [22], not only requires user-created ontologies as external knowledge, but also has no guarantee that their generated semantic facets are relevant to all seed entities. As an illustrative example in Fig. 1, EgoSet generates more than five facets, but only two of them are relevant to both seeds.

To handle the key challenge of multi-faceted set expansion, we propose a novel framework, FUSE, as illustrated in Fig. 2. First, we discover all possible facets of each seed by clustering its skip-grams, and then utilize an iterative algorithm robust to skip-gram noise to expand each facet. Extensive experiments demonstrate that FUSE can accurately identify multiple semantic facets of the seed set and generate quality entities for each facet.
extracting and clustering its skip-grams. Second, we leverage an optimization formulation to discover the shared semantic facets across all seeds as coherent semantic facets. This helps eliminate those facets relevant only to a partial set of seeds. Third, based on the coherent skip-gram clusters, we design an iterative framework to further reduce skip-gram noise and provide quality entities for each facet.

It is considerably complicated to evaluate such multi-faceted set expansion task, mainly because we have no prior knowledge about the number of facets in a seed set. Therefore, we are likely to observe different number of facets between the generated result and the ground truth (e.g., the ground truth may have 3 facets, while the generated result has 4 facets). Previously proposed metric MMAP in [22] only measures how many entities and facets in the ground truth are covered by the generated result. However, it fails to measure how noisy those generated facets are and thus it biases toward methods that output as many facets as possible. To overcome the intrinsic limitation of MMAP, we propose a more comprehensive evaluation metric, BMAP, that can not only capture the purity of generated facets but also their coverage of ground truth facets.

Our contributions are highlighted as follows.

- We identify the key challenge of multi-faceted set expansion and develop an iterative framework, FUSE, to address it.
- We propose to determine semantic facets via clustering skip-gram contexts, and utilize an optimization formulation to discover coherent semantic facets.
- We propose a new evaluation metric for multi-faceted set expansion problem, which is shown to be a more comprehensive measure.
- Extensive experiments demonstrate that our proposed framework outperforms state-of-the-art set expansion algorithms significantly in both accuracy and efficiency.

II. PROBLEM FORMULATION

A facet refers to one semantic aspect or sense of seed words. For example, fruit and company are two facets of the word “apple”. Previous works study mostly single-faceted set expansion and ignore the seeds’ multi-facetedness nature. In this work, we explore a better coverage of all coherent semantic facets of a seed set and study corpus-based multi-faceted set expansion.

More formally, given a seed set query \( q = \{s_1, s_2, ..., s_m\} \) where \( s_i \) is a seed and a raw text corpus \( D \), our set expansion system is to find all lists of entities \( E = \{E^{(1)}, E^{(2)}, E^{(k)}, \ldots\} \), where \( E^{(i)} = \{x^{(i)}_1, x^{(i)}_2, ..., x^{(i)}_n\} \) is relevant to the \( i \)-th facet \( f_i \) of query \( q \), and \( x^{(i)}_j \) denotes an expanded entity.

III. MODEL

Our proposed FUSE framework consists of three main steps: 1) extracting and clustering skip-grams for each seed (c.f. Sec III-A); 2) discovering coherent semantic facets of a seed set (c.f. Sec III-B); and 3) iteratively expanding entity sets for each semantic facet (c.f. Sec III-C).

An overview of our approach is shown in Fig. 2 and the algorithm is shown in Algorithm 1.

---

1Implementations are available at: https://github.com/WanzhengZhu/SetExpansion-MultiFacet.
Algorithm 1: FUSE: Multi-faceted Set Expansion

Input: Corpus $D$; a user query $q$.
Output: a list of expanded entity lists $E$.

1. □ Skip-gram Clustering:
2. seedClusterDict = {};
3. for seed in $q$ do
4.     sgs ← extractSkipgrams(seed, $D$);
5.     sgClusters ← clustering(sgs);
6.     seedClusterDict[seed] ← sgClusters;
7. □ Clusters Fusion:
8.     refSeed ← $q$.pop();
9.     refC ← seedClusterDict[refSeed];
10. while seed is not empty do
11.         curSeed ← $q$.pop();
12.         curC ← seedClusterDict[curSeed];
13.         coherentC ← fuseClusters(refC, curC);
14.         refC ← coherentC;
15. □ Entity Expansion:
16.     $E$ ← entityExpansion(refC);
17. return $E$;

A. Skip-gram Features Extraction and Clustering

We preprocess the raw corpus as SetExpan [29] does, and extract skip-gram features of seed words as Egoset [22] does. Here skip-gram features are a sequence of words surrounding the seed word. Based on the distributional hypothesis [16], the semantics of the seed word is reflected by its neighboring skip-grams. We can derive different facets of a seed word by separating its skip-grams into different semantic clusters.

Embedding is commonly used in NLP applications to represent rich semantic information of words and phrases. We obtain the embedding for each skip-gram by simply averaging the embedding of its component words. The derivation of skip-gram embedding is another interesting research question, but it is not our focus in this work.

Now we cluster these skip-gram embeddings to discover different semantic facets of a seed word. Most clustering algorithms require the number of clusters as input, which deviates from our problem setting. Also, we note that the embedding usually lies in a high-dimension space (typically deviates from our problem setting. Also, we note that the algorithms require the number of clusters as input, which deviates from our problem setting. Therefore, these algorithms are hard coded cluster numbers as mentioned above. Specifically, we will construct a complete weighted graph where each node represents a skip-gram, and the edge weight between each pair of nodes indicates the cosine similarity of their corresponding skip-gram embeddings. As the edge weight measures the semantic relevance of two skip-grams, we expect those semantically coherent skip-grams are put into one graph cluster (i.e., a community in graph). Given a graph cluster $C$ with skip-gram clusters $C_i, C_j, \ldots$, we define its intra-cluster relevance $R(C)$ as the total relevance between word pairs in the same cluster as follow:

$$R(C) = \sum_{i \neq j} r_{i,j} \cdot 1_{(C_i = C_j)}$$

where $r_{i,j}$ is the relevance between skip-gram $i$ and $j$ (defined in Eq. [2] below), and $C_i, C_j$ are the clusters of node $i$ and $j$, respectively.

Translating this goal into graph model, we want to group nodes with strong semantic connections to each other. We do not use the edge weight as connection strength directly, since some nodes might have high edge weight with all nodes while other nodes tend to have low edge weights. We use a normalized edge weight as the relevance of the node pair $(i, j)$ by subtracting node-specific weight bias from the edge weight $e_{i,j}$. Mathematically, we define the node relevance as follows:

$$r_{i,j} = e_{i,j} - \frac{1}{W} \cdot \left( \sum_{k} e_{i,k} \right) \cdot \left( \sum_{k} e_{j,k} \right)$$

where $i, j$ are two nodes, $e_{i,j}$ is the edge weight, $\sum_{k} e_{i,k}$ is the total edge weights of node $i$, and $W$ is the total edge weight in the graph. Finally, by replacing $r_{i,j}$ in Eq. [1] with Eq. [2] we can obtain the intra-cluster relevance $R(C)$ of the given cluster $C$.

To find cluster $C$ that maximizes the intra-cluster relevance $R(C)$, we adopt the graph community detection algorithm Louvain [1]. Louvain starts by assigning a different community to each node. Then, it greedily aggregates communities to optimize the intra-community relevance until the relevance cannot be further improved, by when the “optimal” number of communities will be identified. Empirical results demonstrate that the community detection based skip-gram clustering is able to identify a reasonable number of facets (c.f. Section IV-D).

B. Discovering Coherent Semantic Facets of A Seed Set

After obtaining multiple skip-gram clusters for each seed, we then need to find the shared semantic facets among all seeds and generate the coherent skip-gram clusters. Take two seed words “apple” with facets fruit and company, and “orange” with facets fruit and color as an example, their common facet is fruit.

The key is to determine whether a facet of seed word $A$ matches any facet of word $B$. Suppose that $A$ has $r$ skip-gram clusters $S_A = \{S_{A}^{(1)}, \ldots, S_{A}^{(r)}\}$ where cluster $S_{A}^{(i)}$ contains a set of skip-grams relevant to the $i$-th facet of $A$. Similarly, $B$ has $t$ skip-gram clusters $S_B = \{S_{B}^{(1)}, \ldots, S_{B}^{(t)}\}$. If $A$ and $B$ share $k$ facets, and accordingly they have $k$ pairs of matching clusters $\{(S_{A}^{(i)}, S_{B}^{(j)}), \ldots, (S_{A}^{(i)}', S_{B}^{(j)'}), \ldots\}$. Therefore, these $k$ facets are jointly represented by these clusters: $S_{A,B} = \{S_{A}^{(i)} \cup S_{B}^{(j)} , \ldots, S_{A}^{(i)'} \cup S_{B}^{(j)'}\}$. 
We first measure the pairwise correlation of their skip-gram clusters (c.f. Sec III-B1), and then make a matching decision on a pair of clusters (c.f. Sec III-B2).

1) **Calculating correlation between two skip-gram clusters:** Suppose that facet A1 (one facet of word A) corresponds to a skip-gram cluster \( X = [x_1, \ldots, x_m] \) with \( m \) skip-gram vectors, where \( x_i \in \mathbb{R}^d \). Similarly, facet B1 (one facet of word B) corresponds to a cluster \( Y = [y_1, \ldots, y_n] \) with \( n \) skip-gram vectors, where \( y_j \in \mathbb{R}^d \). Two clusters \( X \) and \( Y \) are from different seed words, and we want to measure their correlation in order to decide whether they correspond to the same semantic facet.

To measure their correlation, we find the semantic sense which \( X \) and \( Y \) have in common. Inspired by the idea of compositional semantics \([10, 28]\), we set the sense vector to the linear combination of skip-gram vectors.

Suppose that the sense vector \( u \) from cluster \( X \) and the sense vector \( v \) from \( Y \) are the sense shared by the two clusters. Therefore, the common sense vectors should be highly correlated, i.e., we want to find \( u \) and \( v \) so that their correlation \( u^T v \) is maximized. We formulate the following optimization problem \([3]\):

\[
\max_{u, v} \frac{u^T v}{||u|| \cdot ||v||} \quad \text{s.t.} \quad u = X a, \quad v = Y b,
\]

where \( a \in \mathbb{R}^m \) and \( b \in \mathbb{R}^n \) are coefficient vectors.

Solving the problem \([3]\) by CCA \([8]\), we can find their common sense vectors \( u^* \) and \( v^* \). The semantic correlation \( \text{corr}(X, Y) \) between cluster \( X \) and \( Y \) is defined as the correlation between these two sense vectors:

\[
\text{corr}(X, Y) = u^*^T v^* \quad (4)
\]

2) **Matching facets of all seeds:** After quantifying correlation for two skip-gram clusters, we cast it as a binary decision whether the cluster \( X \) of facet \( A_1 \) matches semantically with any facet of word \( B \).

We note that it is not a good way to decide the matching clusters by setting a hard correlation threshold, since the numerical correlation range is word-specific.

It is easy to see that if a facet of seed \( A \) (e.g., \( A_1 \)) is of the same semantic class with a facet of seed \( B \) (e.g., \( B_2 \)), then \( \text{corr}(A_1, B_2) \) is higher than the correlation between \( A_1 \) and any other facets of seed \( B \). Otherwise, the correlation of \( A_1 \) and all facets of seed \( B \) should be equally small.

Based on the intuition above, we define a relevance score below:

\[
\text{rele}(A_1, B) = D_{KL}(\text{Corr}(A_1, B), U) \quad (5)
\]

where \( U \) is uniform distribution, \( \text{Corr}(A_1, B) = \text{softmax}((\text{corr}(A_1, B_1), \ldots, \text{corr}(A_1, B_m))) \), and \( D_{KL} \) is KL-divergence \([13]\).

We then make the matching decision based on the relevance score \( \text{rele}(A_1, B) \). Once the matching decision is satisfied, we find the best matching facet \( B* \) in word \( B \) and generate the coherent skip-gram cluster \( A_1 \cup B* \).

**Remarks:** If there are more than two seed words, we first discover coherent skip-gram clusters of two seeds and then use their coherent skip-gram clusters to match with the third seed and so on.

C. **Iterative Entity Expansion**

The skip-gram clusters for different facets are used to expand the seed set by adding their neighboring words as relevant candidates. This is again based on the distributional hypothesis that words co-occurring with similar skip-grams are likely to be semantically similar.

It is unavoidable to include noises in the generated skip-gram clusters due to various factors such as noisy skip-gram representations and the clustering process. We note that noisy skip-grams can seriously degrade the quality of generated words in the set expansion process. In this part, we will illustrate how we can reduce the skip-gram noise with an iterative refining process.

For each skip-gram denoted as \( sg \), we learn its ‘TF-IDF’ weight \([22, 26]\) \( h_{c, sg} \) associated with a word candidate \( c \),

\[
h_{c, sg} = \log(1 + N_{c, sg}) \log \left( \frac{|W|}{\sum_{c'} N_{c', sg}} \right)
\]

where \( N_{c, sg} \) is the co-occurrences of word \( c \) and skip-gram \( sg \) in the corpus, and \( |W| \) is the total number of candidate entities.

Therefore, given a set of skip-grams, the importance weight \( w_c \) of a word candidate \( c \) is:

\[
w_c = \sum_{sg'} h_{c, sg'} \cdot w_{sg'}
\]

where \( w_{sg'} \) is the skip-gram weight defined as:

\[
w_{sg} = \sum_{c'} h_{c', sg} \cdot w_c
\]

Two equations above, in fact, represent an iterative framework to update word weights and skip-gram weights respectively. Quality skip-grams will rank quality words high and in turn, quality words will make quality skip-grams weigh more. For the first iteration, all skip-gram weights \( w_{sg} \) are set to 1. Then we proceed with word weights update and skip-gram weights update, and then iterate. Empirically, we find that it converges very fast and we use 3 iterations in our experiment.

IV. **Experiments**

Our model targets the corpus-based entity set expansion problem, and thus we evaluate its performance on a local corpus.

**Dataset:** We evaluate our approach, FUSE, on the public benchmark dataset used in \([22]\). This dataset contains 56
A. Evaluation Metric

It is considerably complicated to properly evaluate multi-faceted set expansion task due to different number of facets between the generated result and the ground truth. Previous work [22] adopted the following mean of mean average precision (MMAP) measure:

$$MMAP@l = \frac{1}{M_q} \sum_{m=1}^{M_q} AP_l(B_{q,m^*}, G_{qm}),$$

where $M_q$ is the number of facets for query $q$ in the ground truth; $G_{qm}$ is the ground truth set of $m$-th facet for $q$; $B_{q,m^*}$ is the output facet that best matches $G_{qm}$, and $AP_l(c, r)$ represents the average precision of top $l$ entities in a ranked list $c$ given an unordered ground truth set $r$. This metric measures the coverage of ground truth sets by the generated sets.

However, it does not penalize additional noisy facets in generated sets and thus it is biased towards the model that generates more facets. For example, a model generating 15 facets with 3 relevant facets achieves higher MMAP than another model generating 3 facets with 2 relevant facets. One can “cheat” the performance by generating as many facets as possible.

To overcome the intrinsic limitation of MMAP, we, inspired by [4], [7], propose a new metric, Best-Matching Average Precision (BMAP) to capture both the purity of generated facets and their coverage of ground truth facets. Our metric is defined as follows:

$$BMAP@l = HMean(MMAP@l, PMAP@l),$$

$$PMAP@l = \frac{1}{F_q} \sum_{f=1}^{F_q} AP_l(B_{qf}; G_{ql^*}),$$

where $F_q$ is the number of facets in generated output; $B_{qf}$ is the $f$-th output ranked list for query $q$; $G_{ql^*}$ is the ground truth facet that best matches $B_{qf}$. Here $HMean(a, b) = \frac{2ab}{a+b}$ is the harmonic mean of $a$ and $b$.

Our proposed BMAP metric not only evaluates how well generated facets match the ground truth by $MMAP@l$ but also penalizes low-quality facets by $PMAP@l$. Intuitively, $MMAP@l$ measures “recall” to capture how many ground truth results has been discovered, while $PMAP@l$ measures “precision” to capture the fraction of good facets in the generated output. Accordingly, $BMAP@l$ measures “F1 score” to leverage “precision” and “recall”. Results are reported by averaging all 150 queries.

B. Methods

The following approaches are evaluated:

- **word2vec**: We use the “skip-gram” model in word2vec to learn the embedding vector for each entity, and then return $k$ nearest neighbors of the seed words.
- **SEISA** [9]: An entity set expansion algorithm based on iterative similarity aggregation. It uses the occurrence of entities in web list and query log as entity features. In our experiments, we replace the web list and query log with skip-gram features.
- **SetExpan** [26]: A corpus-based set expansion that selects quality context features for entity-entity similarity calculation and expand the entity sets using rank ensemble.
- **EgoSet** [22]: The only existing work for multi-faceted set expansion. It expands word entities from skip-gram features, and then clusters the expanded entities into multiple sets.
- **Sensegram** [19]: We learn different embeddings for each word’s different senses and return $k$ nearest neighbors for each embedding.
- **FUSE-k-means**: A variant of FUSE which replaces Louvain with $k$-means clustering algorithm for skip-gram clustering.

### TABLE I: End-to-end evaluation using MMAP (“recall”), PMAP (“precision”) and BMAP (“F1 score”).

| Method                      | Single-Faceted | Multi-Faceted | Ablations |
|-----------------------------|----------------|--------------|-----------|
|                            | $l=5$          | $l=10$       | $l=20$    |
|                            | $l=5$          | $l=10$       | $l=20$    |
|                            | $l=5$          | $l=10$       | $l=20$    |
| word2vec                   | 0.323          | 0.283        | 0.252     |
|                            | 0.552          | 0.499        | 0.448     |
|                            | 0.390          | 0.352        | 0.316     |
| SEISA                      | 0.345          | 0.301        | 0.268     |
|                            | 0.550          | 0.503        | 0.455     |
|                            | 0.408          | 0.368        | 0.331     |
| SetExpan                   | 0.373          | 0.337        | 0.304     |
|                            | 0.605          | 0.563        | 0.512     |
|                            | 0.448          | 0.413        | 0.374     |
| Sensegram                  | 0.312          | 0.301        | 0.275     |
|                            | 0.479          | 0.443        | 0.398     |
|                            | 0.359          | 0.343        | 0.314     |
| EgoSet                     | 0.446          | 0.390        | 0.325     |
|                            | 0.306          | 0.261        | 0.206     |
|                            | 0.335          | 0.292        | 0.236     |
| FUSE                        | 0.449          | 0.398        | 0.361     |
|                            | 0.643          | 0.570        | 0.517     |
|                            | 0.513          | 0.450        | 0.413     |
| Ablations                  |               |              |           |
| FUSE-k-means ($k=2$)       | 0.419          | 0.365        | 0.328     |
|                            | 0.607          | 0.546        | 0.506     |
|                            | 0.477          | 0.423        | 0.388     |
| FUSE-k-means ($k=3$)       | 0.444          | 0.387        | 0.350     |
|                            | 0.620          | 0.550        | 0.496     |
|                            | 0.500          | 0.442        | 0.400     |
| FUSE-NoIter                | 0.437          | 0.371        | 0.333     |
|                            | 0.601          | 0.533        | 0.478     |
|                            | 0.490          | 0.425        | 0.382     |
- **FUSE-NoIter**: A variant of FUSE without the iterative skip-gram selection module.
- **FUSE**: The full version of our proposed framework with both Louvain approach and iterative skip-gram selection modules enabled.

### C. End-to-End Evaluation

We compare the end-to-end performance of FUSE against all baselines using MMAP (“recall”), PMAP (“precision”) and BMAP (“F1 score”). As shown in Table I, FUSE achieves the highest scores in all cases and outperforms all other baselines with obvious margins in BMAP.

It is worth mentioning that EgoSet achieves decent results in MMAP. However, it generates too many noisy facets, which deteriorate PMAP and the overall performance BMAP. We will further discuss this phenomenon in Sec. [V-D].

It is also interesting to note that single-faceted baselines (i.e., SetExpan) have much stronger PMAP performance than multi-faceted baselines. This is because by generating a single cluster of the most confident expansion results, they usually match with one ground truth cluster very well and thus achieve high PMAP (“precision”) value.

In the ablation analysis, it is worth noting that FUSE, even without pre-determined number of clusters, performs better than FUSE-k-means. We think it is because the noise of forcing skip-grams into a fixed number of clusters will propagate to the skip-gram cluster fusing step, and thus lead to bad performance. Meanwhile, the comparison between FUSE and FUSE-NoIter demonstrates that our iterative framework is able to reduce the skip-gram noise and generate more coherent entities.

### D. Number of Facets Identified

|       | $l_1$ distance | $l_2$ distance |
|-------|---------------|---------------|
| EgoSet| 783           | 78.02         |
| FUSE  | 159           | 26.55         |

We explore the number of facets identified by different multi-faceted set expansion methods. Specifically, we adopt $l_1$ and $l_2$ distances.

\[
l_1 \text{ distance } = \sum_{q \in Q} |\text{GT}_q - \text{Gen}_q| \]

\[
l_2 \text{ distance } = \sqrt{\sum_{q \in Q} (\text{GT}_q - \text{Gen}_q)^2} \]

Here $Q$ is all queries, $\text{GT}_q$ and $\text{Gen}_q$ are the number of facets that ground truth has and the number of facets that the corresponding model identifies for query $q$, respectively.

We set the number of clusters $k$ equals to 2 and 3, which are the mode and the mean of the number of clusters of the ground truth respectively.
listing top skip-grams of each facet. It is worth noting that even the ground truth may not present a full coverage of semantic facets of given seeds. For example, as shown in Case 4, the ground truth only includes semantic facet Animals. Our system also finds another reasonable semantic facet Tributaries. The query "{Chongqing}" shown in Case 5 is another example, where the ground truth again fails to capture the semantic facet of War-related Major Cities. FUSE is shown to mine reasonable semantic facets, which may not even be captured by the ground truth.

G. Case Studies: Single-Faceted Setting

From previous results, we have demonstrated that FUSE has favorable performance against state-of-the-art systems in expanding multiple semantic facets of a seed set. In this subsection, we inspect the performance of FUSE when the ground truth has only one single facet.

In single-faceted set expansion task, there is exactly one semantic facet derived from a seed set. Most existing systems [3], [9], [15], [18], [22], [25], [26], [31], [33]–[35] adopt a two-step approach to the set expansion task:

1) Extracting representative contextual features (e.g., skip-grams) from the seed set;
2) Ranking the keyword candidates based on the contextual features.

In most real-life cases, it is very common that some words in the seed set might be ambiguous. Such ambiguity of even a single seed will introduce entities related to noisy facets, and thus hurt the quality of the expanded set. For example, the seed set \{"apple", "google"\} has only one semantic facet corresponding to Technology Companies, however, the seed "apple" is an ambiguous word and also covers a noisy facet, i.e., Fruits. Because of the seed word "apple" and its ambiguity, existing approaches (e.g., EgoSet [22], SEISA [9], SetExpan [26], etc.) come up with many noisy contextual features. Therefore, these systems generate keywords related to the facet of Fruits (e.g., "pear", "banana"), influenced by the ambiguity of the seed word "apple", despite the fact that the seed set \{"apple", "google"\} has only one semantic facet. In contrast, FUSE is robust to such lexical ambiguity, since we find the shared coherent semantic facet across all seeds and do the entity expansion using only relevant contextual features. From this example, one can clearly see that for single-faceted set expansion, it is also critical to resolve the lexical ambiguity and identify the common facet among seeds.

To gain deeper insights into different set expansion systems in the single-faceted setting, we present case studies with two seed sets \{"apple", "google"\} and \{"beaver", "fox"\} in Table V. We highlight those noisy entities resulting from seed ambiguity in bold, bright red, and other noisy entities in dark red.

In Table V(a), no existing set expansion algorithms, to the best of our knowledge, can return one clean set of that single coherent semantic facet of Technology Companies. Instead, outputs of all existing works involved noisy entities related to the Fruits facet. In Table V(b), existing set expansion algorithms perform even worse since both seeds have its own ambiguity (e.g., "beaver" can be Animals or Counties in Penn-

TABLE III: Case study on comparison between FUSE and EgoSet.

| Query          | FUSE Identified Facets and Their Associated Top Skip-grams                                                                 | EgoSet Identified Facets and Their Associated Top Skip-grams                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| berkeley       | Facet 1 (Top universities in the US): columbia_university, harvard, new_york, yale_university, princeeton, harvard_university, ucla, stanford_university, columbia, boston_college, nyu, cornell_university, indiana_university, johns_hopkins_university, georgetown, georgetown_university, … | Facet 1: los_angeles, san_diego, san_francisco, santa_barbara, santa_cruz, riverside, sacrament, …, Top Skip-grams: (‘in __, ca.’), (‘to __, california’). |
|                | **Top Skip-grams:** (‘at __, yale’), (‘graduate school at __’), (‘former __ professor’), (‘__, harvard university’), (‘__, campus.’) | Facet 2: stanford, harvard_university, yale_university, columbia, cornell_university, mit, ucsd, …, **Top Skip-grams:** (‘the __ campus.’), (‘at __’s school of’). |
|                | Facet 2 (Cities in California, US): los_angeles, san_diego, santa_barbara, san_francisco, irvine, santa_cruz, riverside, palo_alto_long_beach, oakland, pasadena, san_jose, fresno, … | Facet 3: chicago, san_leandro, huntington_beach, los_angeles_county, oxnard, van_mys, …, **Top Skip-grams:** (‘in __, los’), (‘in __, los angeles’). |
|                | **Top Skip-grams:** (‘of california, __’), (‘area of __, california’), (‘in __, california is’), (‘california, __ and the’), (‘found in __, california’). | Facet 4: culver_city, santa_ana, costa_mesa, alameda, redondo_beach, san_rafael, fullerton, redlands, …, **Top Skip-grams:** (‘city of __, california’), (‘in __, california where’). |
|                | Facet 5: california_state_university, marine_corps_recruit_depot, howard_university, california.polytechnic_state_university, uc_santa_barbara, scripps, … | Facet 5: california_state_university, marine_corps_recruit_depot, howard_university, california.polytechnic_state_university, uc_santa_barbara, scripps, …, **Top Skip-grams:** (‘at __, san’), (‘of california, __’). |
|                | **Top Skip-grams:** (‘city of __, california’), (‘in __, california where’). | Facet 6: ojai, yuba_city, whittier_college, pasadena_city_college, mather_air_force_base, Moffett_field, Beale_AFB, March_field, Beale_air_force_base, Fort_ord, …, **Top Skip-grams:** (‘at __, california.’), (‘at __, california in’). |

7Beaver River, Elk River and Bear River are tributaries of Pennsylvania, Mississippi River and the Great Salt Lake, respectively.
8Chongqing was the second capital of Chinese nationalist party during the war.
TABLE IV: Case studies on top skip-grams for each semantic facet. The concept name of each facet is in bold.

| ID | Query | FUSE Identified Facets: Associated Top Skip-grams | Ground Truth Facets: Example Entities |
|----|-------|--------------------------------------------------|---------------------------------------|
| 1  | (hydrogen, uranium) | Chemical Elements: (helium, __), (of __ and nitrogen,) Energy Sources: (for __ energy), (in __ energy), (the __ fuel cell) | Chemical Elements: Helium, Carbon, Nitrogen, Oxygen Energy Sources: Solar, Coal, Oil, Natural Gas |
| 2  | {apollo, artemis, poseidon} | Major gods: (the god __ and), (zeus __ ), (athena __ ) NASA missions: (nasa's __ ) | Major Greek gods: Aphrodite, Ares, Athena, Zeus NASA missions: Juno, Voyager, InSight, NuSTAR |
| 3  | {Beijing} | Chinese Major Cities: (in __, china), (of __, shanghai) Capitals/International Major Cities: (paris and __), ( __ capital) International Metropolitan/Art Cities: (theater in __), (of music in __), (with the __ symphony orchestra) Olympic Games Host Cities: (olympic games in __), (at the __ olympic) | Chinese Major Cities: Beijing, Shanghai, Wuhan, Harbin Province-level divisions of China: Beijing, Jiangsu, Zhejiang Capital cities in the world: Paris, Tokyo, Jakarta, Berlin Olympic Games Host Cities: Paris, Tokyo, Munich, Calvary |
| 4  | {beaver, elk, bear} | Animals: (tailed deer __), (wolf __ and) Tributaries: (in __ river.), (along the __ river.) | Animals: alligator, bear, deer, pig |
| 5  | {Chongqing} | Chinese Major Cities: (of __, china), (based in __, china) War related Major Cities: (__ broadcasting), (congress of __), (party in __ led by) | Chinese Major Cities: Beijing, Shanghai, Wuhan, Harbin Province-level divisions of China: Beijing, Jiangsu, Zhejiang, Guangdong |
| 6  | {fox} | Animals: (__, wolf __), (__ and), (badger), (deer __, and) Popular TV networks: (in the __ sitcom), (on the __ tv), (the __ drama series) English Surnames: (officer of __), (by david __) | Animals: alligator, bear, deer, pig Popular TV networks: NBC, CBS, CNN Common English surnames: Smith, Jones, Williams Snake species: Blind snake, Sea snake, Python |

**sylvenia, US, and “fox” can be Animals, Popular TV Network, or Common English Surnames.** Thus, too many irrelevant semantic facets from ambiguous seeds lead to unsatisfactory results.

As has shown, FUSE performs favorably against previous approaches in single-faceted set expansion too, in that it is robust to semantic ambiguity by extracting the coherent semantic facet and skip-gram set shared by all seed words.

**V. RELATED WORK**

Early work on entity set expansion, including Google Sets [31], SEAL [33], and Lyretrail [3] submits a query consisting of seed entities to a general-domain search engine (e.g., Google) and then mines the returned, top-ranked web pages. These approaches depend on the external search engine and require costly online data extraction.

Later studies, therefore, shift to the corpus-based set expansion setting, where sets are expanded within a given domain-specific corpus. For example, [18] compute the semantic similarity between two entities based on their local contexts and treat the nearest neighbors around the seed entities as the expanded set. [9] further extend this idea by proposing an iterative similarity aggregation function to calculate entity-entity similarity using query logs and web lists besides free text. More recently, [26], [27] propose to compute semantic similarity using only selected high-quality context features, and [14], [15] develop SetExpander system to leverage multi-context term embedding for entity set expansion. All the above attempts, however, assume the input seed entities belong to one unique, clear semantic class, and thus largely suffer from the multi-faceted nature of these seeds – they could represent multiple semantic meanings.

To resolve the ambiguity of seeds, [33] propose to utilize the target semantic facet name and then retrieve its most relevant web tables. However, it requires the exact name of the target semantic facet and outputs the one semantic facet of entities only. This does not accomplish multi-faceted set expansion in nature.

The only initial attempt towards multi-faceted set expansion is EgoSet [22], to the best of our knowledge. EgoSet first extracts quality skip-gram features to construct an ego-network, and then clusters all candidates in the ego-network into different facets, and lastly combines them with external knowledge (e.g., user-generated ontologies) to generate final output. Although FUSE may appear to be similar to EgoSet at first glance, we show key differences and significance over the following aspects:

- **Key challenges of multi-faceted set expansion:** We identify the key challenge of multi-faceted set expansion to be discovery of shared semantic facets from a seed set. While in EgoSet, noisy facets that are relevant only to partial seeds are also generated. As a result, it fails to solve multi-seed single-faceted cases (e.g., {“apple”, “google”}) and multi-seed multi-faceted cases (e.g., Fig. [1]).
### TABLE V: Case studies on single-faceted set expansion.

(a) Case study 1: Expanded entities of query {“apple”, “google”}

| Approach | Expanded Entities |
|----------|-------------------|
| **Ground Truth** | **Technology Companies**: samsung electronics, foxconn, hp, ibm, amazon, microsoft, sony, panasonic, dell, intel, toshiba, … |
| FUSE | microsoft, sony, ibm, motorola, hewlett_packard, yahoo, dell, general_electric, intel, ati_technologies, altavista, lycos, citrix_systems, … |
| SetExpan | microsoft, yahoo, ibm, **pear**, novell, **strawberry**, aol, sony, **company**, netscape, **cherry**, **banana**, mozilla, abc, apple_inc, **black**, **grape**, motorola, intel, adobe_systems, american, … |
| EgoSet | **Facet 1**: microsoft, ibm, sony, hewlett_packard, intel, motorola, sun Microsystems, oracle, atari, aol, macromedia, adobe_systems, lenovo, hp, compaq, emc, …  
**Facet 2**: facebook, myspace, ebay, and, twitter, yahoo, gmail, icq, hotmail, netflix, iphone, verizon_wireless, …  
**Facet 3**: sprint_nextel, symantec, samsung, internet, telecom_italia, southwesterenBell, adobe_reader, javafx, google_talk, …  
**Facet 4**: linux, amiga, windows, vms, openvms, posix, windows_base, gnu, unix, symbian, microcontroller, android, …  
**Facet 5**: milk, pineapple, breadfruit, dairy_product, grapefruit, software, cream_soda, risc, commodore_64, palm, lemon, fruit, …  
**Facet 6**: spss, linux_kernel, electronic_arts, msn_messenger, file_share, freeware, middleware, napster, …  
**Facet 7**: red_hat, samsung_electronics, postgresql, db2, microsoft_sql_server, firebird, mastercard, sap_ag, cognos, … |
| word2vec | microsoft, novell, iPod, aol, **pineapple**, netscape, mozilla, apple_inc, pda, adobe_systems, microsoft_office, excel, iphone, ibm, search_engine, corel, app_store, messaging, ipod, ebay, … |
| SEISA | microsoft, ibm, sony, **company**, abc, black, american, intel, australia, hewlett_packard, dell, **banana**, motorola, nbc, oracle, hp, red, … |

(b) Case study 2: Expanded entities of query {“beaver”, “fox”}

| Approach | Expanded Entities |
|----------|-------------------|
| **Ground Truth** | **Animals**: alligator, crocodile, alpaca, ant, antelope, ape, armadillo, ass, donkey, burro, baboon, badger, bat, bear, beaver, bee, beetle, … |
| FUSE | deer, raccoon, bear, wolf, rabbit, skunk, coyote, bobcat, tiger, cat, squirrel, dog, **black_tail_deer**, mountain_lion, bird, sheep, brown_bears, leopard, owl, opossum, squirrels, pine_marten, rat, skunks, hyena, … |
| SetExpan | raccoon, badger, **green**, coyote, **abc**, otter, bear, **lion**, nbc, moose, **black**, **fetal_cat**, deer, **indian**, **african_leopard**, lovejoy, amazon_river_dolphin, rock, … |
| EgoSet | **Facet 1**: deer, moose, skunk, elk, coyote, marten, raccoon, mink, squirrel, white_tail_deer, mountain_lion, bird, sheep, brown_bears, leopard, owl, opossum, squirrels, pine_marten, rat, skunks, hyena, …  
**Facet 2**: duck, eagle, otter, lost, flint, swift, cherokee, dog, turtle, **blair**, pond, perch, **myrtle**, birch, borden, sallisaw, butte, cat, …  
**Facet 3**: black, green, bear, **falls**, white, salmon, mercer, concord, big_sandy, pear, harnett, coeur_d_alene, yukon, bremer, …  
**Facet 4**: prairie, **sioux**, **osage**, huron, owl, wild_horse, driftwood, loon, **chevy_chase**, sequoyah, bledsoe, **hancock**, covington, …  
**Facet 5**: henson, marsh, gopher, **newberry**, stork, creighton, colby, rockford, drake, barr, douglass, …  
**Facet 6**: drysdale, alewife, **bat_masterson**, clearwater, burnsie, cranberry, portage, rubicon, …  
**Facet 7**: sioux_county, huntsville, circleville, hampstead_county, logan_county, pike_county, hot_springs, hershey, …  
**Facet 8**: butler, hamilton, lewis, rooney, montgomery, trout, bridget, **nbc_tv**, dunbar, holmes, mcdonald, fraser, cbs_television, … |
| word2vec | badger, otter, mouse, bear, coyote, lovejoy, prairie, cheyenne, daniel_boone, duck, kit_carson, deadwood, saginaw, trout, wabash, … |
| SEISA | **green**, abc, black, nbc, **indian**, rock, american, **bbc**, british, silver, **cbs**, red, white, river, deer, south, hbo, elk, washington, bear, … |

- **External knowledge**: EgoSet requires user-created ontology (e.g., Wikipedia) as external knowledge. While these semi-structured web tables and ontologies are helpful for disambiguation, they are not always available for domain-specific corpus. FUSE relies on free text only and thus, can be applied in a more general setting.
- **Clustering over skip-grams**: EgoSet adopts clustering over expanded entities, while FUSE adopts clustering over skip-grams. Clustering over entities usually leads to mediocre results in non-parametric settings, since any expanded entity can be ambiguous. However, skip-grams, consisting of multiple words, are usually of more clear semantics and much easier to be clustered compared to entities themselves (demonstrated in Sec. [V-D]. In additional, EgoSet adopts hard clustering on entities, which ignores the nature that the same entity may fall into different facets (e.g., “Paris” should appear in both sets of National Capitals and Olympic Games Host Cities in Fig. [1], while the design of FUSE naturally allows the same entities to appear in multiple outputted facets.

More generally, our work is also related to word sense disambiguation [11]. [19]–[21]. [30] and computational effi-
We identify the key challenge of the problem – multi-faceted set expansion and have proposed a novel and effective approach, FUSE, to address it. By extracting and clustering skip-grams for each seed, identifying coherent semantic facets of all seeds and iteratively expanding entity sets for each semantic facet, FUSE is capable of identifying semantically reasonable facets, generating quality entity set for each facet, and therefore outperforms previous state-of-the-art approaches significantly.

The proposed framework FUSE is general in that it is able to achieve quality set expansion in both multi-faceted and single-faceted settings. In particular, it, for the first time, is able to solve the case where different seeds have different multi-facet edness. In addition, FUSE can incorporate external knowledge (e.g., Wikipedia) for general-domain set expansion too. For future work, we plan to explore more on skip-gram representations and quality entity expansion from a set of skip-grams.

REFERENCES
[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment, 2008(10):P10008, 2008.
[2] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li. Context-aware query suggestion by mining click-through and session data. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 875–883. ACM, 2008.
[3] Z. Chen, M. Cafarella, and H. Jagadish. Long-tail vocabulary dictionary extraction from the web. In WSDM, 2016.
[4] N. Chinchor. Muc-4 evaluation metrics. In Proceedings of the 4th conference on Message understanding, pages 22–29. Association for Computational Linguistics, 1992.
[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence, 24(5):603–619, 2002.
[6] X. Dai, H. Yin, and N. K. Jha. Grow and prune compact, fast, and accurate lstms. arXiv preprint arXiv:1805.11797, 2018.
[7] M. K. Goldberg, M. Hayvanovych, and M. Magdon-Ismail. Measuring similarity between sets of overlapping clusters. In Social Computing (SocialCom), 2010 IEEE Second International Conference on, pages 303–308. IEEE, 2010.
[8] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. Neural computation, 16(12):2639–2664, 2004.
[9] Y. He and D. Xin. Seisa: set expansion by iterative similarity aggregation. In WWW, 2011.
[10] K. M. Hermann and P. Blunsom. Multilingual models for compositional distributed semantics. arXiv preprint arXiv:1404.4643, 2014.
[11] I. Jaccobi, M. T. Plchov, and R. Navigli. Embeddings for word sense disambiguation: An evaluation study. In Proceedings of the 4th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 897–907, 2016.
[12] A. Jain and M. Pennacchiotti. Open entity extraction from web search query logs. In Proceedings of the 23rd International Conference on Computational Linguistics, pages 510–518. Association for Computational Linguistics, 2010.
[13] S. Kullback and R. A. Leibler. On information and sufficiency. The annals of mathematical statistics, 22(1):79–86, 1951.
[14] J. Manou, O. Pererg, M. Wasserblat, I. Dagan, Y. Goldberg, A. Eirew, Y. Green, S. Guskin, P. Izsak, and D. Korat. Setexpander: End-to-end term set expansion based on multi-context term embeddings. In COLING, 2018.
[15] J. Manou, O. Pereg, M. Wasserblat, A. Eirew, Y. Green, S. Guskin, P. Izsak, and D. Korat. Term set expansion based nlp architect by intel ai labs. In EMNLP, 2018.
[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119, 2013.
[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In NIPS, 2013.
[18] P. Pantel, E. Crestan, A. Borkovsky, A.-M. Popescu, and V. Vyas. Web-scale distributional similarity and entity set expansion. In EMNLP, 2009.
[19] M. Pelevina, N. Arefev, C. Biemann, and A. Panchenko. Making sense of word embeddings. In RepNL@ACL, 2016.
[20] A. Raganato, C. D. Bovi, and R. Navigli. Neural sequence learning models for word sense disambiguation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1156–1167, 2017.
[21] A. Raganato, J. Camacho-Collados, and R. Navigli. Word sense disambiguation: A unified evaluation framework and empirical comparison. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, volume 1, pages 99–110. Association for Computational Linguistics, 2017.
[22] X. Rong, Z. Chen, Q. Mei, and E. Adar. Egoseet: Exploiting word ego-networks and user-generated ontology for multifaceted set expansion. In WSDM, 2016.
[23] A. Sarker and G. Gonzalez. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. Journal of biomedical informatics, 53:196–207, 2015.
[24] L. Sarments, V. Jikouim, M. de Rijke, and E. C. Oliveira. “more like these”: growing entity classes from seeds. In CIKM, 2007.
[25] L. Sarments, V. Jikouim, M. De Rijke, and E. Oliveira. More like these: growing entity classes from seeds. In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pages 959–962. ACM, 2007.
[26] J. Shen, Z. Wu, D. Lei, J. Shang, X. Ren, and J. Han. Setexpansion: Corpus-based set expansion via context feature selection and rank ensemble. In ECML/PKDD, 2017.
[27] J. Shen, Z. Wu, D. Lei, C. Zhang, X. Ren, M. T. Vanni, B. M. Sadler, and J. Han. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In KDD, 2018.
[28] R. Socher, A. Karpathy, V. Q. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. Transactions of the Association of Computational Linguistics, 2(1):207–218, 2014.
[29] M. Steinbach, L. Ertöz, and V. Kumar. The challenges of clustering high dimensional data. In New directions in statistical physics, pages 273–309. Springer, 2004.
[30] K. Taghipour and H. T. Ng. Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 314–323, 2015.
[31] S. Tong and J. Dean. System and methods for automatically creating lists, 2008. US Patent 7,350,187.
[32] P. Velardi, S. Faralli, and R. Navigli. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. Computational Linguistics, 39(3):665–707, 2013.
[33] C. Wang, K. Chakrabarti, Y. He, K. Ganjam, Z. Chen, and P. A. Bernstein. Concept expansion using web tables. In WWW, 2015.
[34] R. C. Wang and W. W. Cohen. Language-independent set expansion of named entities using the web. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 314–323, 2015.
[35] S. Tong and J. Dean. System and methods for automatically creating lists, 2008. US Patent 7,350,187.
[36] P. Velardi, S. Faralli, and R. Navigli. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. Computational Linguistics, 39(3):665–707, 2013.
[37] C. Wang, K. Chakrabarti, Y. He, K. Ganjam, Z. Chen, and P. A. Bernstein. Concept expansion using web tables. In WWW, 2015.
[38] R. C. Wang and W. W. Cohen. Language-independent set expansion of named entities using the web. In ICDM, 2007.
[39] R. C. Wang and W. W. Cohen. Iterative set expansion of named entities using the web. In Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on, pages 1091–1096. IEEE, 2008.
[40] G. Weikum and M. Theobald. From information to knowledge: harvesting entities and relationships from web sources. In ICDM, 2008.
[41] J. Zhao, K. Liu, G. Zhou, Z. Qi, Y. Liu, and X. Han. Knowledge extraction from wikis/bbs/blogs/news web sites., 2014.