SetExpan: Corpus-Based Set Expansion via Context Feature Selection and Rank Ensemble

Jiaming Shen*, Zeqiu Wu*, Dongming Lei, Jingbo Shang, Xiang Ren, Jiawei Han

Department of Computer Science, University of Illinois at Urbana-Champaign, USA
{js2, zeqiuwu1, dlei5, shang7, xren7, hanj}@illinois.edu

Abstract. *Corpus-based set expansion* (i.e., finding the "complete" set of entities belonging to the same semantic class, based on a given corpus and a tiny set of seeds) is a critical task in knowledge discovery. It may facilitate numerous downstream applications, such as information extraction, taxonomy induction, question answering, and web search.
To discover new entities in an expanded set, previous approaches either make *one-time entity ranking* based on distributional similarity, or resort to *iterative pattern-based bootstrapping*. The core challenge for these methods is how to deal with noisy context features derived from free-text corpora, which may lead to entity intrusion and semantic drifting. In this study, we propose a novel framework, *SetExpan*, which tackles this problem, with two techniques: (1) a context feature selection method that selects clean context features for calculating entity-entity distributional similarity, and (2) a ranking-based unsupervised ensemble method for expanding entity set based on denoised context features. Experiments on three datasets show that *SetExpan* is robust and outperforms previous state-of-the-art methods in terms of mean average precision.

Keywords: Set Expansion, Information Extraction, Bootstrapping, Unsupervised Ranking-Based Ensemble

1 Introduction

Set expansion refers to the problem of expanding a small set of seed entities into a complete set of entities that belong to the same semantic class [29]. For example, if a given seed set is {*Oregon, Texas, Iowa*}, set expansion should return a hopefully complete set of entities in the same semantic class, "*U.S. states*". Set expansion can benefit various downstream applications, such as knowledge extraction [8], taxonomy induction [27], and web search [2].

One line of work for solving this task includes *Google Set* [26], *SEAL* [29], and *Lyretail* [2]. In this approach, a query consisting of seed entities is submitted to a search engine to mine top-ranked webpages. While this approach can achieve relatively good quality, the required seed-oriented online data extraction is costly. Therefore, more studies [17][23][10][28][21] are proposed in a *corpus-based* setting where sets are expanded by offline processing based on a specific corpus.

* Equal Contribution

For *corpus-based* set expansion, there are two general approaches, *one-time entity ranking* and *iterative pattern-based bootstrapping*. Based on the assumption that similar entities appear in similar contexts, the first approach [17][23][10] makes a one-time ranking of candidate entities based on their distributional similarity with seed entities. A variety of "contexts" are used, including Web table, Wikipedia list, or just free-text patterns, and entity-entity distributional similarity is calculated based on *all* context features. However, blindly using *all* such features can introduce undesired entities into the expanded set because many context features are not representative for defining the target semantic class although they do have connections with some of the seed entities. For example, when expanding the seed set {*Oregon, Texas, Iowa*}, "*located in* _" can be a pattern feature (the entity is replaced with a placeholder) strongly connected to all the three seeds. However, it does not clearly convey the semantic meaning of *"U.S. states."* and can bring in entities like *USA* or *Ontario* when being used to calculate candidate entity's similarity with seeds. This is *entity intrusion* error. Another issue with this approach is that it is hard to obtain the full set at once without back and forth refinement. In some sense, iteratively bootstrapped set expansion is a more conservative way and leads to better precision.

The second approach, iterative pattern-based bootstrapping [22][8][9], starts from seed entities to extract quality patterns, based on a predefined pattern scoring mechanism, and it then applies extracted patterns to obtain even higher quality entities using another entity scoring method. This process iterates and the high-quality patterns from all previous iterations are accumulated into a pattern pool which will be used for the next round of entity extraction. This approach works only when patterns/entities extracted at each iteration are highly accurate, otherwise, it may cause severe *semantic shift* problem. Suppose in the previous example, *"located in _"* is taken as a good pattern from the seed set {*Oregon, Texas, Iowa*}, and this pattern brings in *USA* and *Ontario*. These undesired entities may bring in even lower quality patterns and iteratively cause the set shifting farther away. Thus, the pattern and entity scoring methods are crucial but sensitive in iterative bootstrapping methods. If they are not defined perfectly, the semantic shift can cause big problems. However, it is hard to have a perfect scoring mechanism due to the diversity and noisiness of unstructured text data.

This study proposes a new set expansion framework, SetExpan, which addresses both challenges posed above for corpus-based set expansion on free text. It carefully and conservatively extracts each candidate entity and iteratively improves the results. First, to overcome the entity intrusion problem, instead of using all context features, context features are carefully selected by calculating distributional similarity. Second, to overcome the semantic drift problem, different from other bootstrapped approaches, our high-quality feature pool will be reset at the beginning of each iteration. Finally, our carefully designed unsupervised ranking-based ensemble method is used at each iteration to further refine entities and make our system robust to noisy or wrongly extracted pattern features.

Figure 1 shows the pipeline at each iteration. SetExpan iteratively expands an entity set through a context feature selection step and an entity selection step. At

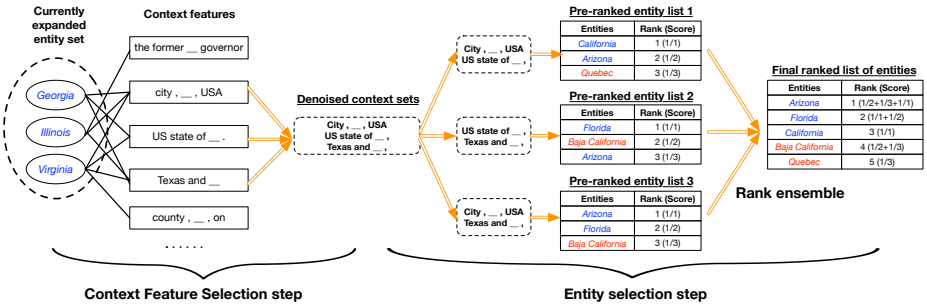


Fig. 1: An example showing two steps in one iteration of SetExpan.

the context feature selection, each context feature is scored based on its strength with currently expanded entities and top-ranked context features are selected. At the entity selection step, multiple subsets of the selected representative context features are sampled and each subset is used to obtain a ranked entity list. Finally, all the ranked lists are collected to compute the final ranking list of each candidate entity for expansion.

The major contributions of this paper are: (1) we propose an iterative set expansion framework with a novel context feature selection approach, to handle the issues of entity intrusion and semantic drift; (2) we develop an unsupervised ranking-based ensemble algorithm for entity selection to make our system robust and further reduce the impact of semantic drift. To evaluate the SetExpan method, we use three publicly available datasets and manually label expanded results of 65 queries over 13 semantic classes. Empirical results show that SetExpan outperforms the state-of-the-art baselines in terms of Mean Average Precision. Code[1] and datasets[2] described in this paper are publicly.

2 Related Work

The problem of completing an entity set given several seed entities has attracted extensive research efforts due to its practical importance. Google Sets [26] was among the earliest work dealing with this problem. It used proprietary algorithms and is no longer publicly accessible. Later, Wang and Cohen proposed *SEAL* system [29], which first submits a query consisting of all seed entities into a general search engine and then mines the top-ranked webpages. Recently, Chen et al. [2] improved this approach by leveraging a "page-specific" extractor built in a supervised manner and showed good performance on long-tail (i.e., rare) term expansion. All these methods need an external search engine and require seed-oriented data extraction. In comparison, our approach conducts corpus-based set expansion without resorting to online data extraction from specific webpages.

[1] https://github.com/mickeystroller/SetExpan
[2] https://tinyurl.com/SetExpan-data

To tackle the corpus-based set expansion problem, Ghahramani and Heller [6] used a Bayesian method to model the probability that a candidate entity belongs to some unknown cluster that contains the input seeds. Pantel et al. [17] developed a web-scale set expansion pipeline by exploiting distributional similarity on context words for each candidate entity. He et al. proposed the SEISA system [10] that used query logs along with web lists as external evidence besides free text, and designed an iterative similarity aggregation function for set expansion. Recently, Wang et al. [28] leveraged web tables and showed very competitive results when not only seed entities but also intended class name were given. While these semi-structured lists and tables are helpful, they are not always available for some specific domain corpus such as PubMed articles or DBLP papers. Perhaps the most relevant work to ours is by Rong [21]. In that paper, the authors used the skip-gram feature combined with additional user-generated ontologies (i.e., Wikipedia list) for set expansion. However, they targeted the multifaceted expansion and exploited all skip-gram features for calculating the similarity because two entities. In our work, we keep the core idea of distributional similarity but calculate such similarity using only carefully selected *denoised* context features.

In a broader sense, our work is also related to information extraction and named entity recognition. Without given enough training data, bootstrapped entity extraction system [5][7][8] is the most popular and effective choice. At each bootstrap iteration, the system will first create patterns around entities; score patterns based on their ability to extract more positive entities and less negative entities (if provided), and use top-ranked patterns to extract more candidate entities. Multiple pattern scoring and entity scoring functions are proposed. For example, Riloff et al. [20] scored each pattern by calculating the ratio of positive entities among all entities extracted by it, and scored each candidate entity by the number and quality of its matched patterns. Gupta et al. [7] scored patterns using the ratio of scaled frequencies of positive entities among all entities extracted by it. All these methods are heuristic and sensitive to different model parameters.

More generally, our work is also related to class label acquisition [24][30] which aims to propagate class labels to data instances based on labeled training examples, and entity clustering [1][12] where the goal is to find clusters of entities. However, the class label acquisition methods require a much larger number of training examples than the typical size of user input seed set, and the entity clustering algorithms can only find semantically related entities instead of entities strictly in the same semantic class.

3 Our Methodology: The SetExpan Framework

This section introduces first the context features and data model used by SetExpan in Sect. 3.1 and then our context-dependent similarity measure in Sect. 3.2. It then discusses how to select context features in Sect. 3.3 and presents our novel unsupervised ranking-based ensemble method for entity selection in Sect. 3.4.

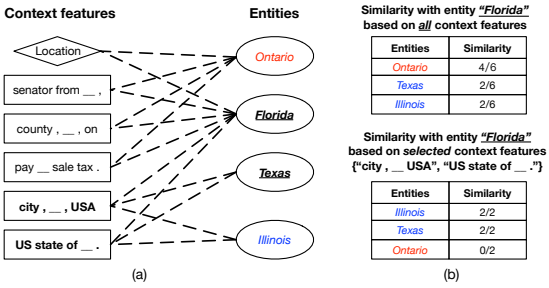


Fig. 2: (a) A simplified bipartite graph data model. (b) Similarity with seed entity conditioned on two different sets of context features.

3.1 Data Model and Context Features

We explore two types of context features obtained from the plain text: (1) skip-grams [21] and (2) coarse-grained types [8]. Data is modeled as a bipartite graph (Figure 2(a)), with candidate entities on one side and their context features on the other. Each type of context features are described as follows.

Skip-gram: Given a target entity e_i in a sentence, one of its skip-gram is "w_{-1} _ w_1" where w_{-1} and w_1 are two context words and e_i is replaced with a placeholder. For example, one skip-gram of entity *"Illinois"* in sentence *"We need to pay Illinois sales tax."* is "pay _ sales". As suggested in [21], we extract up to six skip-grams of different lengths for one target entity e_i in each sentence. One advantage of using skip-grams is that it imposes strong positional constraints.

Coarse-grained type: Besides the unstructured skip-gram features, we use coarse-grained type to filter those obviously-wrong entities. For examples, when we expand the "U.S. states", we will not consider any entity that is typed "Person". After this process, we can obtain a cleaner subset of candidate entities. This mechanism is also adopted in [8].

After obtaining the "nodes" in bipartite graph data model, we need to model the edges in the graph. In this paper, we assign the weight between each pair of entity e and context feature c using the *TF-IDF transformation* [21], which is calculated as follows:

$$f_{e,c} = \log(1 + X_{e,c}) \left[\log |E| - \log \left(\sum_{e'} X_{e',c} \right) \right], \tag{1}$$

where $X_{e,c}$ is the raw co-occurrence count between entity e and context feature c, $|E|$ is the total number of candidate entities. We refer to such scaling as the *TF-IDF transformation* since it resembles the *tf-idf* scoring in information retrieval if we treat each entity e as a "document" and each of its context feature c as a "term". Empirically, we find such weight scaling performs outperforms some other alternatives such as point-wise mutual information (PMI) [10], truncated PMI [15], and BM25 scoring [19].

3.2 Context-dependent Similarity

With the bipartite graph data model constructed, the task of expanding an entity set at each iteration can be viewed as finding a set of entities that are most "similar" to the currently expanded set. In this study, we use the weighted Jaccard similarity measure. Specifically, given a set of context features F, we calculate the *context-dependent* similarity as follows:

$$Sim(e_1, e_2|F) = \frac{\sum_{c \in F} \min(f_{e_1,c}, f_{e_2,c})}{\sum_{c \in F} \max(f_{e_1,c}, f_{e_2,c})}. \quad (2)$$

Notice that if we change context feature set F, the similarity between entity pair is likely to change, as demonstrated in the following example.

Example 1 *Figure 2(a) shows a simplified bipartite graph data model where all edge weights are equal to 1 (and thus omitted from the graph for clarity). The entity "Florida" connects with all 6 different context features, while the entity "Ontario" is associated with top 4 context features including 1 type feature and 3 skip-gram features. If we add all the 6 possible context features into the context feature set F, the similarity between "Florida" and "Ontario" is $\frac{1+1+1+1}{1+1+1+1+1+1} = \frac{4}{6}$. On the other hand, if we put only two context features "city , _, USA", "US state of _ ." into F, the similarity between same pair of entities will change to $\frac{1+1}{1+1} = \frac{2}{2}$. Therefore, we refer such similarity as context-dependent similarity.*

Finally, we want to emphasize that our proposed method is general in the sense that other common similarity metrics such as cosine similarity can also be used. In practice, we find the performance of a set expansion method depends not really on the exact choice of base similarity metrics, but more on which contexts are selected for calculating *context-dependent* similarity. Similar results were also reported in a previous study [10].

3.3 Context Feature Selection

As shown in Example 1, the similarity between two entities really depends on the selected feature set F. The motivation of context feature selection is to find a feature subset F^* of fixed size Q that best "profiles" the target semantic class. In other words, we want to select a feature set F^* based on which entities within target class are most "similar" to each other. Given such F^*, the entity-entity similarity conditioned on it can best reflect their distributional similarity with regard to the target class. In some sense, such F^* best profiles the target semantic class. Unfortunately, to find such F^* of fixed size Q, we need to solve the following optimization problem which turns out to be NP-Hard, as shown in [3].

$$F^* = \arg\max_{|F|=Q} \sum_{i=1}^{|X|} \sum_{j>i}^{|X|} Sim(e_i, e_j|F), \quad (3)$$

where X is the set of currently expanded entities. Initially, we treat the user input seed set S as X. As iterations proceed, more entities will be added into X.

Given the NP-hardness of finding the optimal context feature set, we resort to a heuristic method that first scores each context feature based on its accumulated strength with entities in X and then selects top Q features with maximum scores. This process is illustrated in the following example:

Example 2 *For demonstration purpose, we again assume all edge weights in Figure 2(a) are equal to 1 and let the currently expanded entity set X be {"Florida", "Texas"}. Suppose we want to select two "denoised" context features, we will first score each context feature based on its associated entities in X. The top 4 contexts will obtain a score 1 since they match only one entity in X with strength 1, and the 2 contexts below will get a score 2 because they match both entities in X. Then, we rank context features based on their scores and select 2 contexts with highest scores: "city , _, USA", "US state of _ ." into F.*

Finally, we want to emphasize two major differences of our context feature selection method from other heuristic "pattern selection" methods. First, most pattern selection methods require either users to explicitly provide the "negative" examples for the target semantic class [11][8][22], or implicitly expand multiple mutually exclusive classes in which instances in one class serve as negative examples for all the other classes [4][15]. Our method requires only a small number of "positive" examples. In most cases, it is hard for humans to find good discriminative negative examples for one class, or to provide both mutually exclusive and somehow related comparative classes. Second, the bootstrapping method will add its selected "quality patterns" during each iteration into a quality pattern pool, while our method will select high quality context features at each iteration from scratch. If one noisy pattern is selected and added into the pool, it will continue to introduce more irrelevant entities at all the following iterations. Our method can avoid such noise accumulation.

3.4 Entity Selection via Rank Ensemble

Intuitively, the entity selection problem can be viewed as finding those entities that are most similar to the currently expanded set X conditioned on the selected context feature set F. To achieve this, we can rank each candidate entity based on its score in eq. (4) and then add top-ranked ones into the expanded set:

$$score(e|X,F) = \frac{1}{|X|}\sum_{e' \in X} Sim(e,e'|F). \tag{4}$$

However, due to the ambiguity of natural language in free-text corpora, the selected context feature set F may still be noisy in the sense that an irrelevant entity is ranked higher than a relevant one. To further reduce such errors, we propose a novel ranking-based ensemble method for entity selection.

The key insight of our method is that an inferior entity will not appear frequently in multiple pre-ranked entity lists at top positions. Given a selected context set F, we first use sampling without replacement method to generate T subsets of context features $F_t, t = 1, 2, \ldots, T$. Each subset is of size $\alpha|F|$ where α

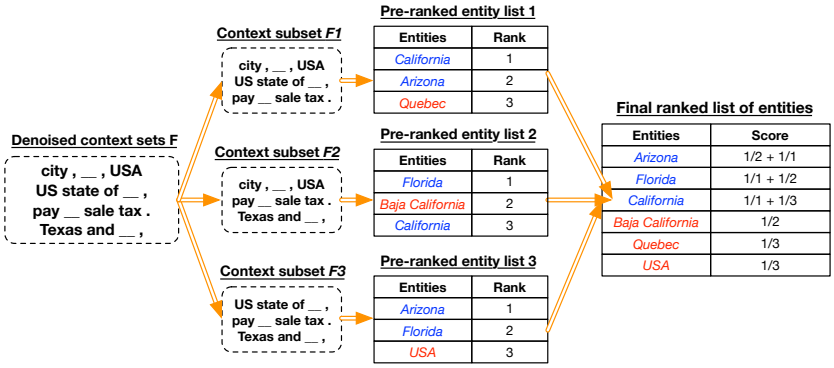


Fig. 3: A toy example to show entity selection via rank ensemble.

is a model parameter within range $[0, 1]$. For each F_t, we can obtain a pre-ranked list of candidate entities L_t based on $score(e|X, F_t)$ defined in eq. (4). We use r_t^i to denote the rank of entity e_i in list L_t. If entity e_i does not appear in L_t, we let $r_t^i = \infty$. Finally, we collect T pre-ranked lists and score each entity based on its mean reciprocal rank (mrr). All entities with average rank above r, namely $mrr(e) \leq T/r$, will be added into entity set X.

$$mrr(e_i) = \sum_{t=1}^{T} \frac{1}{r_t^i}, \qquad r_t^i = \sum_{e_j \in E} I\left(score(e_i|X, F_t) \leq score(e_j|X, F_t)\right), \tag{5}$$

where $I(\cdot)$ is the indicator function. Naturally, a relevant entity will rank at top position in multiple pre-ranked lists and thus accumulate a high mrr score, while an irrelevant entity will not consistently appear in multiple lists at high position which leads to low mrr score. Finally, we use the following example to demonstrate the whole process of entity selection.

Example 3 *In Figure 3, we want to expand the "US states" semantic class given a selected context feature set F with 4 features. We first sample a subset of 3 context features $F_1 = \{$"city , __ , USA", "US state of __ ,", "pay __ sales tax ."$\}$, and then use F_1 to obtain a pre-ranked entity list $L_1 = \langle$"California", "Arizona", "Quebec"$\rangle$. By repeating this process three times, we get 3 pre-ranked lists and ensemble them into a final ranked list in which entity "Arizona" is scored 1.5 because it is ranked in the 2nd position in L_1 and 1st position in L_3. Finally, we add those entities with mrr score larger than 1, meaning this entity is ranked at 3rd position on average, into the expanded set X. In this simple example, the model parameters $T = 3, \alpha = \frac{|F_1|}{|F|} = 0.75$, and $r = 3$.*

Put all together. Algorithm 1 summarizes the whole SetExpan process. The candidate entity set E and bipartite graph data model G are pre-calculated and stored. A user needs only to specify the seed set S and the expected size of output set K. There is a total of 4 model parameters: the number of top quality context

Algorithm 1 SetExpan

Input: Candidate entity set E, initial seed set S, entity-context graph G, expected size of output set K, model parameters $\{Q, T, \alpha, r\}$.
Output: The expanded set X.
$X = S$.
while $|X| \leq K$ **do**
  Set $F = \emptyset$ // Select denoised contexts from scratch
  Score context features based on X and add top Q denoised contexts into F.
  // Entity-selection via rank ensemble
  for $t = 1, 2, \ldots, T$ **do**
    Uniformly sample αQ contexts and construct feature subset F_t.
    Score entities based on Eq. (4) given F_t and obtain the pre-ranked list L_t.
    Update the mrr score of each entity based on Eq. (5).
  end for
  $X = X \cup \{e | mrr(e) \geq \frac{T}{r}\}$ // Add entities into expanded set X .
end while
Return X.

features selected in each iteration Q, the number of pre-ranked entity lists T, the relative size of feature subset $0 < \alpha < 1$, and final mrr threshold r. The tuning and sensitivity of these parameters will be discussed in the experiment section.

4 Experiments

4.1 Experimental Setup

Datasets preparation. *SetExpan* is a *corpus-based* entity set expansion system and thus we use three corpora to evaluate its performance. Table 1 lists 3 datasets we used in experiments. (1) **APR** is constructed by crawling all 2015 news articles from AP and Reuters. (2) **Wiki** is a subset of English Wikipedia used in [13]. (3) **PubMed-CVD** is a collection of research paper abstracts about cardiovascular disease retrieved from PubMed.

For APR and PubMed-CVD datasets, we adopt a data-driven phrase mining tool [14] to obtain entity mentions and type them using ClusType [18]. Each entity mention is mapped heuristically to an entity based on its lemmatized surface name. We then extract variable-length skip-grams for all entity mentions as features for their corresponding entities, and construct the bipartite graph data model as introduced in the previous section. For Wiki dataset, the entities have already been extracted and typed using distant supervision. For the type information in each dataset, there are 16 coarse-grained types in APR and 4 coarse-grained types in PubMed-CVD. For Wiki, since it originally has about 50 fine-grained types, which may reveal too much information, we manually mapped them to 11 more coarse-grained types.

Query construction. A query is a set of seed entities of the same semantic class in a dataset, serving as the input for each system to expand the set. The process of query generation is as follows. For each dataset, we first extract 2000 most frequent entities in it and construct an entity list. Then, we ask three volunteers to manually scan the entity lists and propose a few semantic classes for each list. The proposed class should be interesting, relatively unambiguous and has a reasonable coverage in its corresponding corpus. These semantic classes

Table 1: Datasets statistics and Query descriptions

Dataset	FileSize	#Sentences	#Entities	#Test queries
APR	775MB	1.01M	122K	40
Wiki	1.02GB	1.50M	710K	20
PubMed-CVD	9.3GB	23M	179K	5

cover a wide variety of topics, including locations, companies as well as political parties, and have different degrees of difficulty for set expansion. After finalizing the semantic classes for each dataset, the students randomly select entities of each semantic class from the frequent entity list to form 5 queries of size 3. To select the queries for PubMed-CVD, we seek help from two additional students with biomedical expertise, following the same previous approach. Due to the large size of PubMed-CVD dataset and runtime limitation, we only select 1 semantic class (hormones) with 5 queries.

With all queries selected, we have humans to label all the classes and instances returned by each of the following 7 compared methods. For APR and Wiki datasets, the inter-rater agreements (kappa-value) over three students are 0.7608 and 0.7746, respectively. For PubMed-CVD dataset, the kappa-value is 0.9236. All entities with conflicting label results are further resolved after discussions among all human labelers. Thus, we have our ground truth datasets.

Compared methods. Since the focus on this work is the corpus-based set expansion, we do not compare with other methods that require online data extractions. Also, to further analyze the effectiveness of each module in SetExpan framework. We implement 3 variations of our framework.

- word2vec [16]: We use the "skip-gram" model in word2vec to learn the embedding vector for each entity, and then return k nearest neighbors around seed entities as the expanded set.
- PTE [25]: We first construct a heterogeneous information network including entity, skip-gram features, and type features. PTE model is then applied to learn the entity embedding which is used to determine the k nearest neighbors around seed entities.
- SEISA [10]: 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 our skip-gram and coarse-grained context features.
- EgoSet [21]: A multifaceted set expansion system based on skip-gram features, word2vec embeddings and WikiList. The original system is proposed to expand a seed set to multiple entity sets, considering the ambiguities in seed set. To achieve this, we use a community detection method to separate the extracted entities into several communities. However, in order to better compare with EgoSet, we carefully select queries that have little ambiguity or at least the seed set in the query is dominating in one semantic class. Thus, we discard the community detection part in EgoSet and treat all extracted entities as in one semantic class.

Table 2: Overall end-to-end performance evaluation on 3 datasets over all queries.

Methods	APR			Wiki			PubMed-CVD		
	MAP@10	MAP@20	MAP@50	MAP@10	MAP@20	MAP@50	MAP@10	MAP@20	MAP@50
EgoSet	0.3949	0.3942	0.3706	0.5899	0.5754	0.5622	0.0511	0.0410	0.0441
SEISA	0.7423	0.6090	0.3892	0.7643	0.6606	0.4998	-	-	-
word2vec	0.6054	0.5385	0.4180	0.7193	0.6289	0.4510	0.8427	0.7701	0.6895
PTE	0.3144	0.2777	0.1996	0.6817	0.5596	0.3839	0.9071	0.7654	0.5641
SetExpan^{-cs}	0.8240	0.7997	0.7674	0.9540	0.8955	0.7439	**1.000**	**1.000**	0.5991
SetExpan^{-re}	0.8509	0.7792	0.7681	0.9392	0.8680	0.7291	**1.000**	0.9605	0.7371
SetExpan^{full}	**0.8967**	**0.8621**	**0.7885**	**0.9571**	**0.9010**	**0.7457**	**1.000**	**1.000**	**0.7454**

– SetExpan^{-cs}: Disable the context feature selection module in SetExpan, and use all context features to calculate distributional similarity.
– SetExpan^{-re}: Disable the rank ensemble module in SetExpan. Instead, we use all selected context feature to rank candidate entities at one time and add top-ranked ones into the expanded set.
– SetExpan^{full}: The full version of our proposed method, with both context feature selection and rank ensemble components enabled.

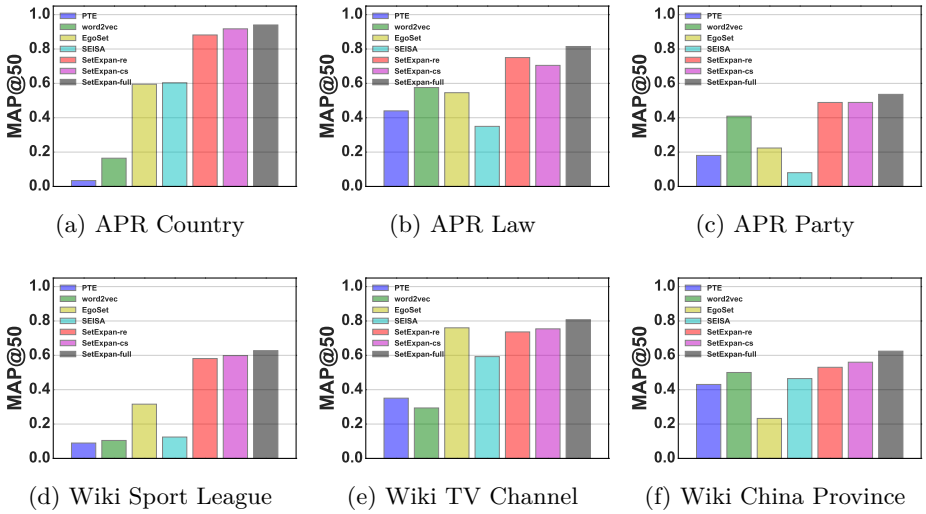
For fair comparison, we try different combinations of parameters and report the best performance for each baseline method.

Evaluation Metrics. For each test case, the input is a query, which is a set of 3 seed entities of the same semantic class. The output will be a ranked list of entities. For each query, we use the conventional average precision $\text{AP}_k(c, r)$ at k (k = 10, 20, 50) for evaluation, given a ranked list of entities c and an unordered ground-truth set r. For all queries under a semantic class, we calculate the mean average precision (MAP) at k as $\frac{1}{N}\sum_i AP_k(c_i, r)$, where N is the number of queries. To evaluate the performance of each approach on a specific dataset, we calculate the mean-MAP (MMAP) at k over all queried semantic classes as $\text{MMAP}_k = \frac{1}{T}\sum_{t=1}^{T}[(\frac{1}{N_t})\sum_i AP_k(c_{ti}, r_t)]$, where T is the number of semantic classes, N_t is the number of queries of t-th semantic class, c_{ti} is the extracted entity list for i-th query for t-th semantic class, and r_t is the ground truth set for t-th semantic class.

4.2 Experimental Results

Comparison with four baseline methods. Table 2 shows the MMAP scores of all methods on 3 datasets[3]. We can see the MMAP scores of SetExpan outperforms all four baselines a lot. We further look at their performances on each concept class, as shown in Figure 4. We can see that the performance of these baseline methods varies a lot on different semantic classes, while our SetExpan can consistently beat them. One reason is that none of these methods applies context feature selection or rank ensemble, and a single set of unpruned features can lead to various levels of noise in the results. Another reason is the lack of an iterative mechanism in some of those approaches. For example, even if EgoSet includes

[3] Results of SEISA on PubMed-CVD are omitted due to the scalability issue.



(a) APR Country (b) APR Law (c) APR Party

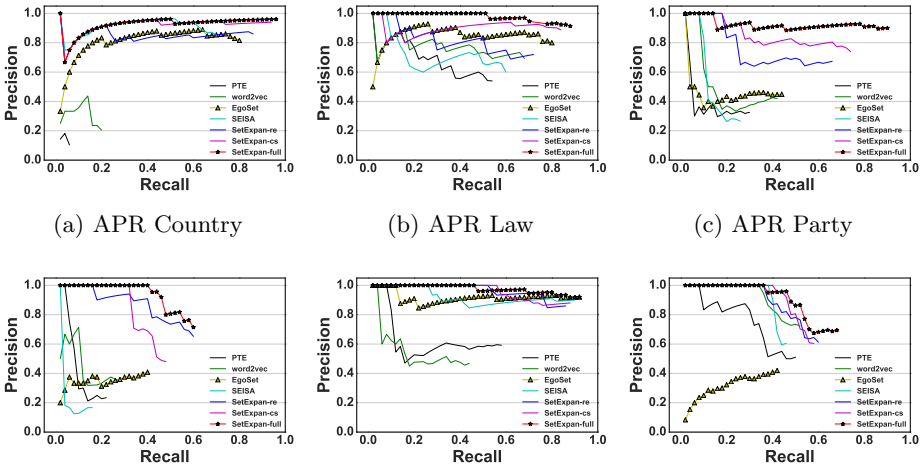
(d) Wiki Sport League (e) Wiki TV Channel (f) Wiki China Province

Fig. 4: Evaluation results for each semantic class.

the results from word2vec to help it boost the performance, it still achieves low MAP scores in some semantic classes. Finding the nearest neighbors in only one iteration can be a key reason. And although SEISA is applying the iterative technique, instead of adding a small number of new entities in each iteration, it expands a full set in each iteration based on the coherence score of each candidate entity with the previously expanded set. It pre-calculates the size of the expanded set with the assumption that the feature similarities follow a certain distribution, which does not always hold to all datasets or semantic classes. Thus, if the size is far different from the actual size or is too big to extract a confident set at once, each iteration will introduce a lot of noise and cause semantic drift.

Comparison with SetExpan^{-re} and SetExpan^{-cs}. At the dataset level, the MMAP scores of SetExpan^{full} outperforms its two variation approaches. In the semantic class level, we can see that SetExpan^{-re} and SetExpan^{-cs} sometimes have their MAP much lower than SetExpan^{full} while sometimes they almost achieve the same performance with SetExpan^{full}. This means they fail to stably extract entities with good quality. The main reason is still that a single set of features or ensembles over unpruned features can lead to various levels of noise in the results. Only under the circumstances that the single set of features or the unpruned features happen to be nicely selected without too much noise, which tends to happen when the query is relatively "easy", these variation approaches can achieve good results.

Effects of Context Feature Selection. We already see that adding the context feature selection component helps improve the performance. What's also noticeable is that the addition of context selection process becomes more obvious as the size of the corpus increases. The difference between MMAP scores of



(a) APR Country (b) APR Law (c) APR Party

(d) Wiki Sport League (e) Wiki TV Channel (f) Wiki China Province

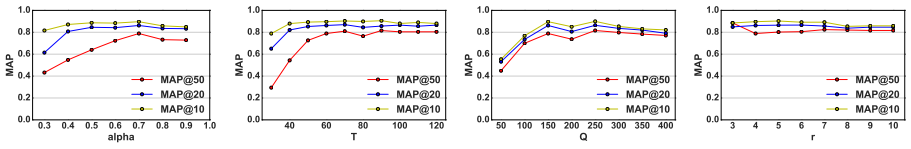
Fig. 5: Evaluation results for each concept class on individual query

SetExpan^{-cs} and SetExpan^{full} is much larger in PubMed-CVD compared with APR and Wiki datasets. This is because that as the corpus size increases, we will have more noisy features and more candidate entities while the good features to define the target entity set may be limited. Thus, without context selection, noise can damage the performance much more. The evidence can also be found from the performance of EgoSet across the three datasets. It can achieve reasonably good results in APR and Wiki, however, it performs much worse in PubMed-CVD.

Effect of Rank Ensemble. From the above experiments, the effect of rank ensemble has variance across the different semantic classes, however, it seems to be more stable across datasets, compared with the effect of context selection. This is because we apply the default set of parameter values in each test case above. In the parameter analysis part, we will show that the number of ensemble batches and the percentage of features to be randomly sampled can affect the contribution of rank ensemble to the set expansion performance.

Parameter Analysis. There are totally 4 parameters in SetExpan – Q (the number of selected context features), α (the percentage of features to be sampled), T (the number of ensemble batches), and r (the threshold of a candidate entity's average rank). We study the influence of each parameter by fixing all other parameters to default values, and present one graph showing the MMAP scores of SetExpan on APR dataset versus the changes of that parameter.

- α: From the graph, the performance increases sharply as α increases until it reaches about 0.6. Then, it starts to stay stable and decreases after 0.7.
- Q: In the range of 50 - 150, the performance increases sharply as Q increases, which means the majority of top 150 context features can provide rich information to identify entities belonging to the target semantic class. The



(a) α (b) T (c) Q (d) r

Fig. 6: Parameter Sensitivity on two datesets

available information gets more and more saturated after Q reaches 150 and start to introduce noises and hamper the performance after around 300.
- r: Our experiments show that the performance is not very sensitive to the threshold of a candidate entity's average rank.
- T: The performance keeps increasing as we increase the ensemble batches, due to the robustness to noise of ensembling. The performance becomes more stable after 60 batches.

Case Studies. Figure 7 presents three case studies for SetExpan. We show one query for each dataset. In each case, we show top 3 ranked entities and top/bottom 3 skip-gram features after context feature selection for the first 3 iterations as well as the coarse-grained type. In all cases, our algorithm successfully extracts correct entities in each iteration, and the top-ranked skip-grams are representative in defining the target semantic class. On the other hand, we notice that most of the bottom 3 skip-grams selected are very general or not representative at all. These context features could potentially introduce noisy entities and thus the rank ensemble can play a rival role in improving the results.

5 Conclusion and Future Work

In this paper, we study the problem of corpus-based set expansion. First, we propose an iterative set expansion framework with a context feature selection method, to deal with the problem of entity intrusion and semantic drift. Second, we develop a novel unsupervised ranking-based ensemble algorithm for entity selection, to further reduce context noise in free-text corpora. Experimental results on three publicly available datasets corroborate the effectiveness and robustness of our proposed SetExpan.

The proposed framework is general and can incorporate other context features besides skip-grams, such as Part-Of-Speech tags or syntactic head tokens. Besides, it would be interesting to study more rank ensemble methods for aggregating multiple pre-ranked lists. In addition, our current framework treats each feature independently, it would be interesting to study how the interaction of context features can influence the expansion result. We leave it for future work.

Acknowledgments. Research was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), National

Dataset	Query	Top ranked entities in first 3 iterations	Top/Bottom skip-gram features selected in first 3 iterations	Coarse-grained type
APR	**{Patriot Act, Obamacare, Clery Act}**	**Iteration 1**: USA Patriot Act, USA Freedom Act, Voting Rights Act, ... **Iteration 2**: Stock Act, Religious Freedom Restoration Act, Foreign Intelligence Surveillance Act, ... **Iteration 3**: Americans with Disabilities Act, Healthy Families Act, Goonda Act, ...	**Iteration 1:** Top 3: "the __ provisions", "provisions of the __", "defund __ ." Bottom 3: "2010 __ ,", "also known as __ .", "under the __ , and" **Iteration 2:** Top 3: "under the __ to", "provisions of the __ ,", "the __ into law." Bottom 3: "the __ - which has", "the _The House", "the _ , first" **Iteration 3:** Top 3: "under the __ to", "Under the __ ,", "the __ into law" Bottom 3: "of the __ passed", "the __ , the most", "replacing __ ."	**Event**
Wiki	**{ESPN, ESPN2, Spike TV}**	**Iteration 1:** ABC, CBS, NBC, ... **Iteration 2:** BBC, ITV, Channel 4, ... **Iteration 3:** TBS, ITV1, BBC Two, ...	**Iteration 1:** Top 3: "telecast on __ .", "televised on __ .", "televised by __ ." Bottom 3: "on __ , to the", ", and perhaps __ .", "from an __ website" **Iteration 2:** Top 3: "the __ sitcom", "the __ television network", "ABC , __ ," Bottom 3: "on __ on September", "broadcast on __ on", "the __ soap opera The" **Iteration 3:** Top 3: "the __ sitcom", "the __ soap", "the __ soap opera", ... Bottom 3: "aired on __ between", "of the __ show", "on the __ crime"	**Organization**
PubMed -CVD	**{FSH, TSH, MSH}**	**Iteration 1:** LH, GH, ACTH, ... **Iteration 2:** LHRH, AMH, GHRH, ... **Iteration 3:** Renin, GnRH-I, AVP, ...	**Iteration 1:** Top 3: "stimulating hormone (__)", "hormone (__) ,", "hormone (__) and", ... Bottom 3: "g/L , __ =", ", __ and prolactin", "hormone (__) -" **Iteration 2:** Top 3: "hormone (__) ,", "hormone (__) and", "hormone (__) .", ... Bottom 3: ", __ , estradiol ,", ", __ , and PRL", "hormone (__) -" **Iteration 3:** Top 3: "hormone (__) ,", "hormone (__) and", "hormone (__) .", ... Bottom 3: "(__) and insulin-like", ", TSH , __ ,", "levels of __ , FSH"	**Proteins and Genes (PRGE)**

Fig. 7: Three case studies on each dataset.

Science Foundation IIS-1320617, IIS 16-18481, and NSF IIS 17-04532, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov).

References

1. R. Balasubramanyan, B. B. Dalvi, and W. W. Cohen. From topic models to semi-supervised learning: Biasing mixed-membership models to exploit topic-indicative features in entity clustering. In *ECML/PKDD*, 2013.
2. Z. Chen, M. Cafarella, and H. Jagadish. Long-tail vocabulary dictionary extraction from the web. In *WSDM*, pages 625–634. ACM, 2016.
3. F. Chierichetti, R. Kumar, S. Pandey, and S. Vassilvitskii. Finding the jaccard median. In *SODA*, 2010.
4. J. R. Curran, T. Murphy, and B. Scholz. Minimising semantic drift with mutual exclusion bootstrapping. 2007.
5. O. Etzioni, M. J. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165:91–134, 2005.
6. Z. Ghahramani and K. A. Heller. Bayesian sets. In *NIPS*, 2005.
7. S. Gupta, D. L. MacLean, J. Heer, and C. D. Manning. Research and applications: Induced lexico-syntactic patterns improve information extraction from online medical forums. *JAMIA*, 21:902–909, 2014.
8. S. Gupta and C. D. Manning. Improved pattern learning for bootstrapped entity extraction. In *CoNLL*, pages 98–108, 2014.
9. S. Gupta and C. D. Manning. Distributed representations of words to guide bootstrapped entity classifiers. In *HLT-NAACL*, 2015.

10. Y. He and D. Xin. Seisa: set expansion by iterative similarity aggregation. In *WWW*, 2011.
11. P. Jindal and D. Roth. Learning from negative examples in set-expansion. In *2011 IEEE 11th International Conference on Data Mining*, 2011.
12. D. Lin and X. Wu. Phrase clustering for discriminative learning. In *ACL/IJCNLP*, 2009.
13. X. Ling and D. S. Weld. Fine-grained entity recognition. In *AAAI*, 2012.
14. J. Liu, J. Shang, C. Wang, X. Ren, and J. Han. Mining quality phrases from massive text corpora. In *SIGMOD*, pages 1729–1744. ACM, 2015.
15. T. McIntosh and J. R. Curran. Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. 2008.
16. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.
17. P. Pantel, E. Crestan, A. Borkovsky, A.-M. Popescu, and V. Vyas. Web-scale distributional similarity and entity set expansion. In *EMNLP*, 2009.
18. X. Ren, A. El-Kishky, C. Wang, F. Tao, C. R. Voss, and J. Han. Clustype: Effective entity recognition and typing by relation phrase-based clustering. In *WWW*, pages 995–1004. ACM, 2015.
19. X. Ren, Y. Lv, K. Wang, and J. Han. Comparative document analysis for large text corpora. *CoRR*, abs/1510.07197, 2017.
20. E. Riloff. Automatically generating extraction patterns from untagged text. In *AAAI/IAAI, Vol. 2*, 1996.
21. X. Rong, Z. Chen, Q. Mei, and E. Adar. Egoset: Exploiting word ego-networks and user-generated ontology for multifaceted set expansion. In *WSDM*, pages 645–654. ACM, 2016.
22. B. Shi, Z. Zhang, L. Sun, and X. Han. A probabilistic co-bootstrapping method for entity set expansion. In *COLING*, 2014.
23. S. Shi, H. Zhang, X. Yuan, and J.-R. Wen. Corpus-based semantic class mining: Distributional vs. pattern-based approaches. In *COLING*, 2010.
24. P. P. Talukdar, J. Reisinger, M. Pasca, D. Ravichandran, R. Bhagat, and F. Pereira. Weakly-supervised acquisition of labeled class instances using graph random walks. In *EMNLP*, 2008.
25. J. Tang, M. Qu, and Q. Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*, pages 1165–1174. ACM, 2015.
26. S. Tong and J. Dean. System and methods for automatically creating lists, 2008. US Patent 7,350,187.
27. P. Velardi, S. Faralli, and R. Navigli. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3):665–707, 2013.
28. C. Wang, K. Chakrabarti, Y. He, K. Ganjam, Z. Chen, and P. A. Bernstein. Concept expansion using web tables. In *WWW*, 2015.
29. R. C. Wang and W. W. Cohen. Language-independent set expansion of named entities using the web. In *ICDM*, 2007.
30. Y.-Y. Wang, R. Hoffmann, X. Li, and J. Szymanski. Semi-supervised learning of semantic classes for query understanding: from the web and for the web. In *CIKM*, 2009.