Corpus-based Set Expansion
with Lexical Features and Distributed Representations

Puxuan Yu, Zhiqi Huang, Razieh Rahimi, and James Allan
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
{pxyu,zhiqihuang,rahimi,allan}@cs.umass.edu

ABSTRACT
Corpus-based set expansion refers to mining “sibling” entities of some given seed entities from a corpus. Previous works are limited to using either textual context matching or semantic matching to fulfill this task. Neither matching method takes full advantage of the rich information in free text. We present CaSE, an efficient unsupervised corpus-based set expansion framework that leverages textual features as well as distributed representations of entities for the set expansion task. Experiments show that CaSE outperforms state-of-the-art set expansion algorithms in terms of expansion accuracy.

ACM Reference Format:

1 INTRODUCTION
Corpus-based set expansion – i.e., finding in a given corpus the complete set of entities that belong to the same semantic class of a few seed entities – is a critical task in information retrieval and knowledge discovery. For example, given the input seed set {Massachusetts, Virginia, Washington}, a set expansion method is expected to output all other states in the United States. Set expansion is broadly useful for a number of downstream applications, such as question answering [14, 23], taxonomy construction [19], relation extraction [9], and query suggestion [1].

Most corpus-based approaches [5, 12, 15–18] are based on the assumption of distributional similarity [6], which, in the context of set expansion, can be understood on two levels: (1) contexts are in textual form so that expanded sets can be explained by reversing the process; and, (2) contexts are features of a latent model (e.g., Word2Vec [13] and BERT [4]) to generate distributed representations of entities. Each dimension of an embedding vector represents an unknown latent concept. Either perspective can be adopted to fulfill the task, though they both have limits. The former transforms the task of finding sibling entities to finding optimal textual patterns. For an entity to be considered a candidate, it has to meet the “hard match” condition: sharing at least one textual pattern with at least one seed. Thus, many target entities end up with low relevance scores especially on smaller corpora. On the other side, distributed representations of entities do not require exact matching of textual patterns because they are calculated according to terms within a certain window, regardless of term arrangement. Therefore, not only sibling entities, but also other semantically related entities, such as twin or parent entities, are included in the final result.

Different from prior methods which explored either side of the distributional hypothesis, we propose CaSE (Corpus-based Set Expansion) framework that combines the two distributional similarity approaches. CaSE constructs a pool of candidate entities with lexical features and improves the ranking scores of target entities using the similarity of distributed representations with regard to user input. Among the two major approaches in corpus-based set expansion, CaSE is categorized as a one-time entity ranking method. Compared to iterative pattern-based bootstrapping, it is much more efficient at query time and is capable of avoiding semantic drift. In addition, unlike many other corpus-based set expansion techniques [7, 16, 18], CaSE does not rely on prior knowledge of relations among entities (e.g., web lists, knowledge bases) to work well. This is crucial because such external resources might not be available for certain languages or domains.

The major contributions of this paper are: (1) we propose the CaSE framework, which combines lexical context matching and distributed representations for set expansion; and, (2) our analysis discloses that inclusion relation between the entity sets and discrimination power of entity contexts can affect set expansion performance. The implementation and evaluation dataset described here are publicly available.

2 RELATED WORK
Web-based Set Expansion: Web-based methods – including Google Sets [22], SEAL [23] and Lyretail [2] – submit queries consisting of seed entities to search engines and analyze the retrieved documents. The assumption that top-ranked webpages cover other entities in the same semantic class is not always true. Also, extracting data from online platforms can be time-consuming at query time. Therefore, most recent studies are proposed in an offline setting.

Corpus-based Set Expansion: Thelen and Riloff [21] described using certain contextual patterns to tag words with limited coarse-grained types. Roark and Charniak [15] first introduced a general set expansion solution based on co-occurrence of entities. Later,
methods that define membership functions based on co-occurrences of entities with contexts were proposed [5, 17]. Instead of text corpora, SEISA [7] uses offline query logs and web lists, and does set expansion with an iterative similarity aggregation function. EgoSet [16] constructs clusters of entities using textual patterns and user-generated ontology respectively, and outputs clusters after refinement.

The most recent and comparable methods to our approach are SetExpan [18] and SetExpaner [12]. Besides selecting contexts based on distributional similarity, SetExpan also leverages coarse-grained types from Wikipedia as features. SetExpan proposed resetting the context pool before each iteration to address the “semantic drift” problem, which turned out to be unsolved since false entities persist in later iterations. In addition, SetExpan takes hundreds of seconds per issued query, making it difficult to use with applications which involve user interaction. SetExpaner takes the second perspective of distributional similarity, and generates variants of distributed representations from different patterns. Similarity scores of each candidate computed per representation with seed entities are treated as features, based on which an MLP binary classifier decides whether a candidate should be in the expanded set. Besides the limitation of solely using distributed representations, patterns such as explicit lists [17] cover only a small portion of entities.

3 METHODOLOGY

Intuitively, CaSE expands input seed entities by semantically related entities that frequently share important contexts with seeds. The first step is to extract features from the contexts of seed entities in the corpus. Different features can be extracted from contexts of entities. Potential features for entity e0 in sentence “w−2w−1w0w1w2” include unigrams (w0), n-grams (w1w2), and skip-grams (w−1w1). Skip-grams impose strong positional constraints [16], reducing the risk of finding relevant concepts rather than true sibling entities. The other alternative is to directly use predefined patterns, e.g., “such as e0, e1 and e2”, for set expansion. However, Shi et al. [20] showed that for large corpora, the construction of syntactic contexts has better accuracy and introduces less noise compared to pattern based methods. Therefore, we extract skip-gram features from entity contexts.

Some preprocessing steps are performed on the text corpus to improve run-time efficiency. First, we extract the set of entities $E = \{e_i | i = 1, 2, \ldots, N\}$ in the given text corpus. We then consider a window of size 4 around each entity mention in the corpus and extract four skip-grams $[-3, 0], [-2, 1], [-1, 2],$ and $[0, 3]$ where $[-x, y]$ means keeping $x$ words before and $y$ words after the entity mention. This setting allows more matchings and thus creates candidate pool with higher recall. Let $\Sigma = \{\sigma_{ij} | j = 1, 2, \ldots, M\}$ denote the extracted skip-grams for $e_i$. Then, the set of all skip-grams in the corpus is $\Sigma = \bigcup_{i=1}^{N} \Sigma_i$. Based on these, we create a frequency matrix $\Phi_{N \times M} = (\phi_{ij}) | i = 1, 2, \ldots, N; j = 1, 2, \ldots, M\}$, where $N = |E|, M = |\Sigma|$, and cell value $\phi_{ij}$ is the number of co-occurrences of entity $i$ with skip-gram $j$.

We also acquire a distributed representation for each entity either by training on the local corpus or using pre-trained representations. Each entity $e_i$ is thus represented as a $D$ dimensional embedding $\psi_i$, in matrix $\Psi_{N \times D} = \{\psi_{ik} | i = 1, 2, \ldots, N; k = 1, 2, \ldots, D\}$.

3.1 Context Feature Selection

At query time, we first build the set of candidate entities. Suppose the set of seeds $S = \{s_q | q = 1, 2, \ldots, L\}$ is a subset of $E$, then the union of the skip-grams of seed entities, $\Sigma_s$, is a subset of $\Sigma$. For a particular query, we derive a sub-matrix $\Phi_s$ from $\Phi$ by column projection; columns of $\Phi_s$ are the context features of seeds, $\Sigma_s$, and the rows represent all entities that share at least one context with at least one seed. These entities are considered as candidate entities for expansion.

We use $\Phi_s$ to quantitatively measure the correlation between seeds and skip-grams. First, we compute $c_{qj}$ as the co-occurrences of seed entity $s_q$ with skip-gram $\sigma_j$ over the total occurrences of $\sigma_j$ in the corpus. Then, the $c$-weight for skip-gram $\sigma_j$ given the current query is defined as:

$$c_j = \frac{\sum_{q=1}^{L} c_{qj}}{\sum_{q=1}^{L} \sum_{s_q \in \Sigma_s} \phi_{qj}}$$

This weight shows the quality of skip-grams, in that the higher the $c$-weight, the more relevant the skip-gram is to the seeds. Since candidate entities are obtained by selecting entities that share skip-grams with seed entities, weighting skip-grams of seed entities can be used to rank candidate entities.

3.2 Entity Search via Semantic Representation

We use semantic similarity between seed and candidate entities to further evaluate candidate entities. In preprocessing steps, we acquire a $D$ dimensional word embedding matrix $\Psi$. The comparison between a seed entity and a candidate entity is equivalent to computing the cosine similarity of two corresponding rows. Denoting the cosine similarity of seed entity $s_q$ and candidate entity $e_i$ as $\cos(e_i, s_q)$, the relatedness of $e_i$ to all seeds is

$$e_i = \frac{1}{L} \sum_{q=1}^{L} h(\cos(e_i, s_q))$$

where $L$ is the length of the query and $h(\cdot)$ is an increasing and strictly positive function. The intuition behind $h(\cdot)$ is that the mathematical difference between $\cos(a, x) = 0.9$ and $\cos(a, y) = 0.8$ is not a sufficient description of the semantic difference between $x$ and $y$. Finally, The score of entity $e_i$ with skip-gram $\sigma_j$, denoted by $\rho_{ij}$, comprises three parts: the $c$-weight of $\sigma_j$, the semantic similarity with seeds of $e_i$, and the smoothed frequency of entity skip-gram co-occurrences. Formally, $\rho_{ij} = c_j \cdot e_i \cdot g(\phi_{ij})$, where $g(\cdot)$ is a concave function. Because an entity could associate with multiple skip-grams, the final score of $e_i$ is the summation over all possible skip-grams.

$$\rho_i = \sum_j \rho_{ij} = \left(\frac{1}{L} \sum_q h(\cos(e_i, s_q))\right) \sum_j \left(\sum_q c_{qj}\right) g(\phi_{ij})$$

We compute $\rho_i$ for each entity in the candidate pool. The set expansion result is the set of entities with top $x$ highest scores, where $x$ is a predefined cutoff.
4 EXPERIMENTS

4.1 Compared Methods

- Word2Vec [13]: We trained word embedding on our corpus using skip-gram Word2Vec model, where window size and number of iterations are set to 6 and 15, respectively. We then use embedding vectors of entities to retrieve the K nearest neighbors of seed entities as the expansion result.
- BERT [4]: BERT is an empirically powerful embedding model for several NLP tasks. We use a pre-trained BERT model (uncased, Large, 1,024 dimensions) to generate embeddings for all entities and perform KNN ranker similar to Word2Vec baseline.
- SetExpander [12]: We perform preprocessing, training and inference in the default setting on evaluation corpora. Implementation is distributed under Intel’s NLP Architect Framework 2.
- SetExpan [18]: We run SetExpan in its default settings with preprocessing steps identical to CaSE.
- CaSE: The unsupervised set expansion framework we proposed. Functions h(·) and g(·) in our model are set to power and root functions as $h(\cos(e_i, s_q)) = \cos(e_i, s_q)^7$, and $g(\phi_{ij}) = \sqrt{\phi_{ij}}$. There are three variations of CaSE:
  - CaSE-mdr: A simpler version of CaSE without distributional embeddings of entities, i.e., $\rho_{ij} = c_j \cdot g(\phi_{ij})$.
  - CaSE-BERT: CaSE model where distributed representations are acquired from a pre-trained BERT model.
  - CaSE-W2V: CaSE model where distributed representations are acquired from a locally trained Word2Vec model.

4.2 Experimental Setup

Datasets and Preprocessing: We use three corpora to evaluate CaSE. (1) AP89 is a collection of 84,678 news reports published by Associated Press in 1989. (2) WaPo is the TREC Washington Post Corpus which contains 608,180 news articles and blog posts from Jan. 2012 to Aug. 2017. (3) Wiki is a subset of English Wikipedia data dump from Oct. 2013, containing 463,819 Wikipedia entries. Consistent with prior work [18], we primarily use a data-driven phrase mining tool AutoPhrase [11] to obtain entity mentions. We adopt the entity mention list from Word2Phrase (part of the Word2Vec [13] Toolkit) as a trivial filter to improve precision. To reduce noise in the larger WaPo and Wiki corpora, four or fewer occurrences of entities in skip-grams are ignored, i.e., cells in $\Phi$ with values $\phi_{ij} < 5$ are set to 0.

Constructing queries: We build a collection of 62 semantic sets for evaluating set expansion algorithms as the selected combination of MRSCs [16], INEX-XER sets [3], SemSearch sets [10], and 12 additional sets from web resources [8]. To evaluate the sensitivity of our algorithm to the number of seed entities, we build queries with length ranging from 2 to 5. For each set consisting of $n$ entities, we build $\min(100, n C_m)$ queries with $m$ random seeds.

Evaluation Metrics: Set expansion algorithms retrieve a ranked list of entities in response to a query. We evaluate the top 100 retrieved entities for each query by all methods described in Section 4.1, except the SetExpan method where all retrieved entities after 10 iterations are evaluated. Mean Average Precision (MAP) is calculated for different queries with the same length across all evaluation sets. Statistical significant tests are performed using the two-tailed paired t-test at the 0.05 level.

4.3 Results and Discussion

Table 1 summarizes the overall performance of different methods for queries with different lengths on three corpora. The results indicate that the best variation of CaSE is CaSE-W2V, which shows robust improvements upon baselines on all corpora for queries of different length (Table 1 and Figure 2). In set-wise comparison, CaSE-W2V outperforms SetExpan and SetExpander with few exceptions (Figure 1) where entities hardly share skip-grams.

Robustness against input length: Intuitively, one might expect better performance given longer queries. SetExpan removes sub-optimal contexts in feature selection, thus showing the expected trend. Embeddings based methods demonstrate contrary behaviors, mainly because more seeds introduce more twin entities at top. CaSE does not remove features but weights them, and further weights entities with distributed similarity. As Table 1 shows, CaSE performs well even with few seeds, and improves slowly as the number of seeds increases.

Gap among evaluation sets: Figure 1 shows that some semantic sets are easier to expand than others. This result partially confirms earlier work showing that the performance of set expansion models improves as the frequencies of candidate entities increase [17]. To specifically show the correlation between entity frequencies and performance of set expansion, we define a composite property for each set $T$. For each entity $e_i$ in $T$, we first calculate the average of number of entities that occur in each skip-gram associated with entity $e_i$, which is denoted by $k_i$. A higher $k$ value means the entity occurs in general contexts shared by more entities. Then,
Table 1: Retrieval accuracy (MAP) across all evaluation queries of all compared methods on different corpora. ▲: statistically significant (95% confidence interval) improvement compared to SetExpan, the strongest baseline.

<table>
<thead>
<tr>
<th>#seeds</th>
<th>AP89</th>
<th>WaPo</th>
<th>Wiki</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Word2Vec</td>
<td>0.032</td>
<td>0.030</td>
<td>0.027</td>
</tr>
<tr>
<td>BERT</td>
<td>0.103</td>
<td>0.094</td>
<td>0.091</td>
</tr>
<tr>
<td>SetExpan</td>
<td>0.058</td>
<td>0.067</td>
<td>0.073</td>
</tr>
<tr>
<td>CaSE-mdr</td>
<td>0.095</td>
<td>0.103</td>
<td>0.111</td>
</tr>
<tr>
<td>CaSE-BERT</td>
<td>0.117▲</td>
<td>0.117▲</td>
<td>0.118</td>
</tr>
<tr>
<td>CaSE-W2V</td>
<td>0.132▲</td>
<td>0.133▲</td>
<td>0.136▲</td>
</tr>
</tbody>
</table>

REFERENCES


