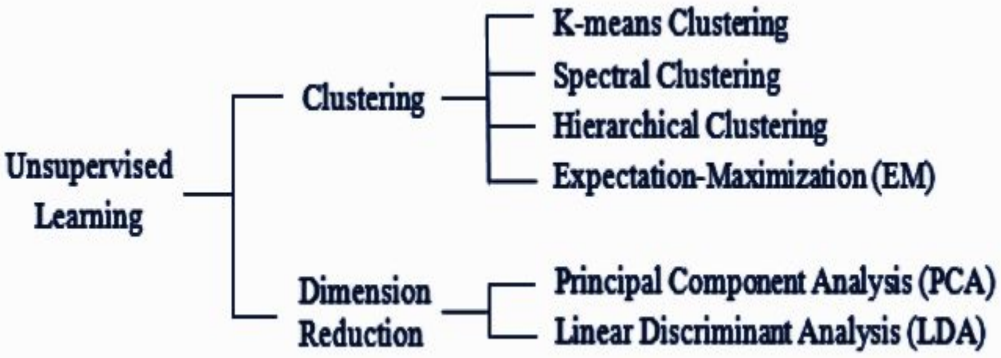
Task guided taxonomy construction

Sanchita Badkas
2019H1030520P

Introduction

- Task guided Taxonomy construction - A novel approach
  - Input : User provided seed taxonomy and domain specific corpus
  - Output : User specific application tasks taxonomy
- Model is based on
  - Weakly supervised relationship extraction
  - Set expansion

Taxonomy Construction:



Taxonomy of unsupervised learning algorithms

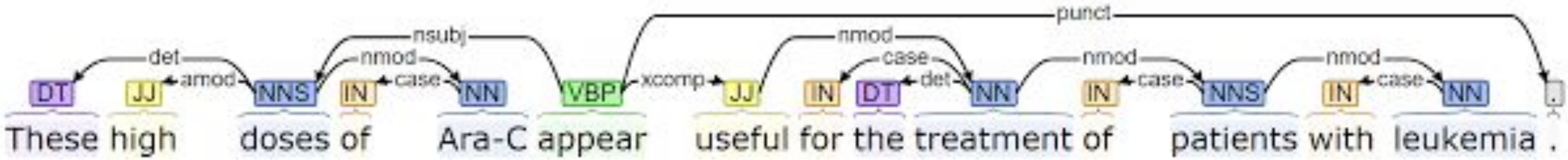
Used for naming and classifying entities in a particular domain.

Approaches for TC:

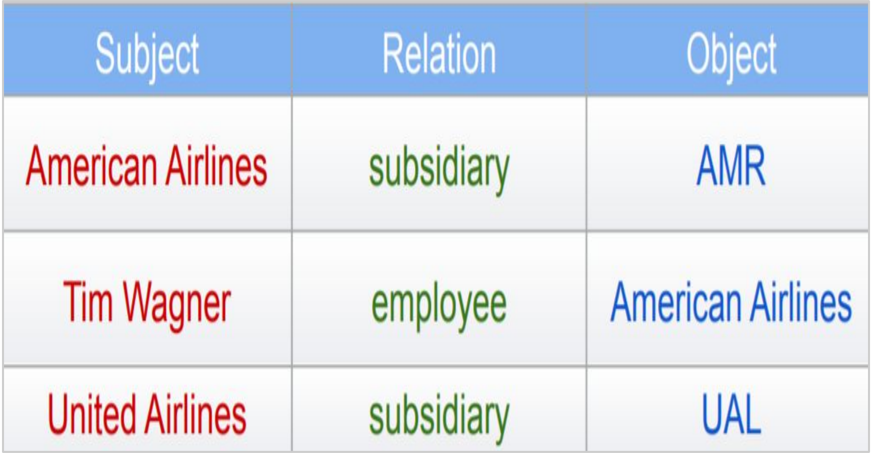
- Manual labour
  - Laborious and subjective
- Bag of words
  - Clustering methodologies applied
  - Relationships ignored
- Rule based
  - Leverage pattern based distributions like “Is-a” patterns
  - Variety in language makes difficult for covering all rules

Relationship Extraction:

- Extracts semantic relationships from a given corpus/document.
- Essential for extracting structured information from unstructured data.



Relationship extraction



Subject	Relation	Object
American Airlines	subsidiary	AMR
Tim Wagner	employee	American Airlines
United Airlines	subsidiary	UAL

‘CHICAGO (AP) — Citing high fuel prices, United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said. United , a unit of UAL, said the increase took effect Thursday night and applies to most routes where it competes against discount carriers, such as Chicago to Dallas and Atlanta and Denver to San Francisco, Los Angeles and New York. ‘

Approaches for RE

- Rule based RE
- Supervised RE
- Weakly supervised RE
- Distantly supervised RE
- Unsupervised RE

Approaches for RE

Rule based RE:

Extracting relations based on predefined rules.

E.g. “USA is a country ” here _ is a _ would be a rule.

Advantages : Simplicity and ease of extraction of relations for a specific domain

Drawbacks : Huge variety in language a lot of human labour is required for creating rules that would all the relations from a corpus. Thus, scaling this to corpus level is difficult.

Approaches for RE

Supervised RE:

Makes use of ML models like binary classifiers, NN to determine presence of relationship among entities.

Advantages : High recall scores

Drawbacks : Corpus needs to be preprocessed by NLP modules to extract features. Expensive to obtain labelled data for domain specific corpus

Approaches for RE

Weakly supervised RE:

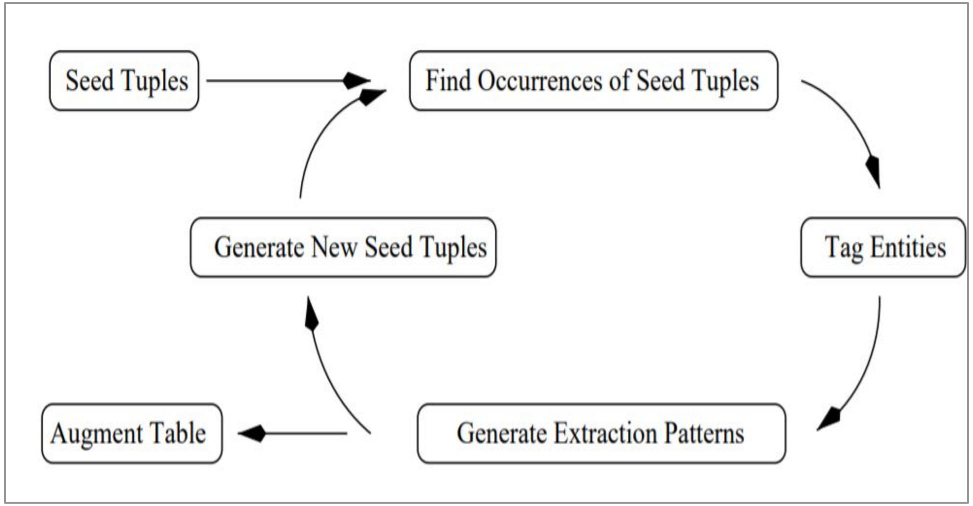
Starts out with handful of seeds and iteratively finds new ones.

Advantages : Expensive labelled data isn’t required in large quantities.

Drawbacks :Seed intrusion errors can cause semantic drift if not monitored.

E.g. :   Seed set - {iron, mercury, carbon} Here, mercury is a planet, element as well as Greek god.

Approaches for RE



Weakly supervised RE:

The snowball method.

Approaches for RE

Distantly supervised RE:

Combination of Weakly supervised and supervised RE approaches. Seed set taken from existing knowledge bases like DBpedia, Yago and iteratively improved.

Advantages : Reduced effort in labelling data.

Drawbacks : Restricted to knowledge base. Also, prone to iterative errors while improving the seed set.

Approaches for RE

Unsupervised RE:

Uses heuristics and general rules for RE.

Advantages : Labelled data not required

Drawbacks : Completely dependent on heuristics

Weakly supervised RE:

Two different approaches:

- Pattern based
- Distribution based

Weakly supervised RE

Pattern based approach:

Used to learn textual pattern for RE. Predicts relationship between entity pair from sentences including both entities.

Advantage : Simplicity.

Limitation : Faces difficulties during matching learned patterns to unseen context.

1. Beijing , the capital of China, is a megacity rich in history.
2. Tokyo , Japan’s capital , was originally a small village .

Here, the pattern X the capital of Y is extracted easily. But, faces difficulty while extracting the same pattern in the second sentence due to the variety in the language.

Weakly supervised RE

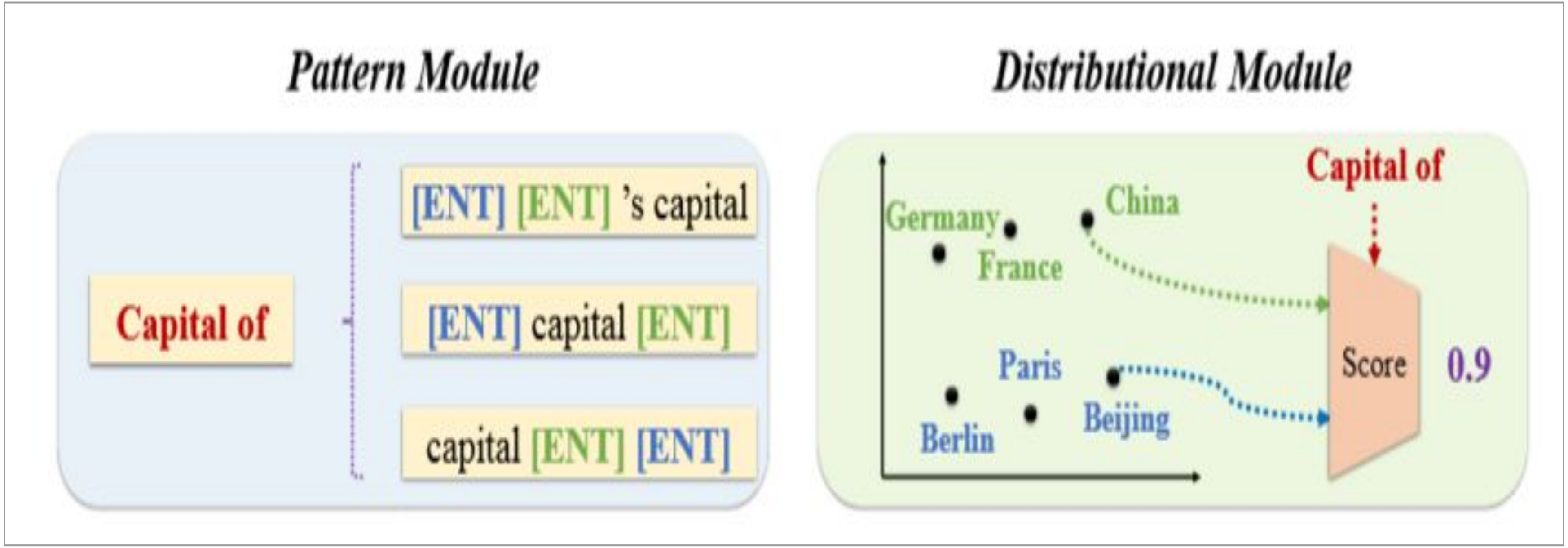
Distributional approach

Use the corpus level co-occurrence statistics of entities.

Focus on learning the low-dimensional representations to preserve these statistics which leads to entities having similar semantic meanings having similar representations.

Parameters like context type, frequency weighting, similarity measures used.

Weakly supervised RE



REPEL

REPEL - Relation Extraction with Pattern-enhanced Embedding Learning

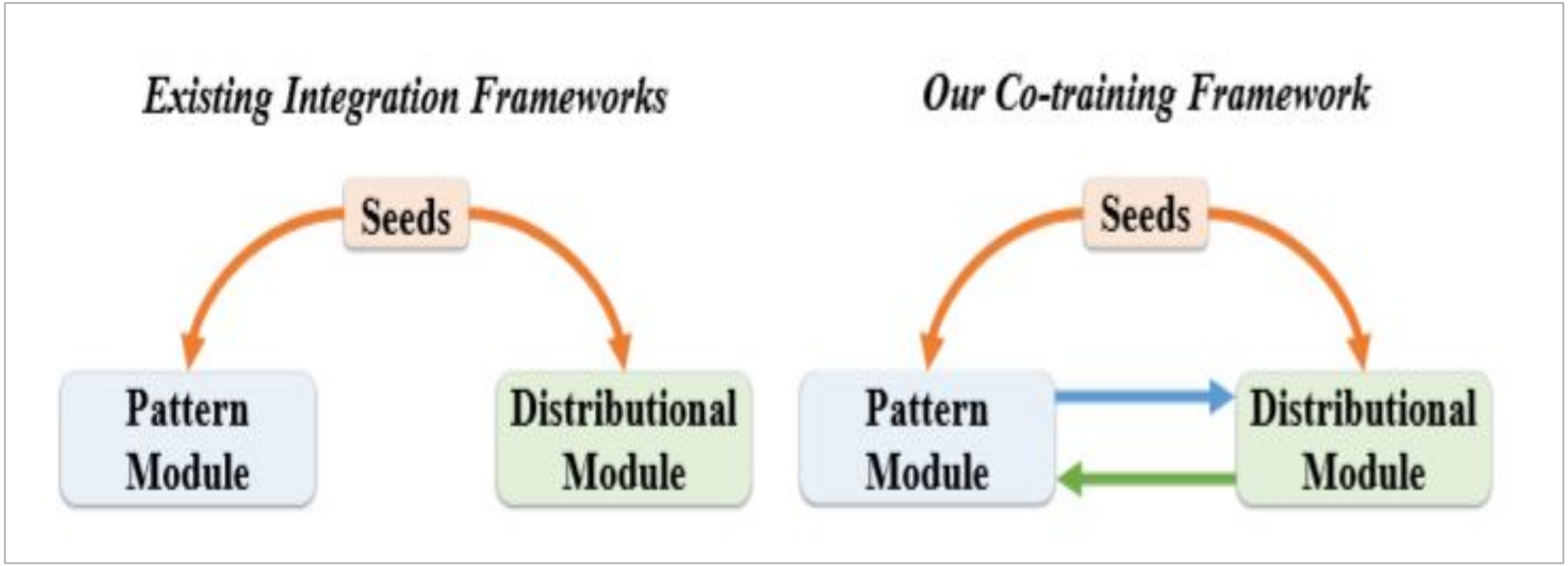
Model works by combining the perspectives of pattern and distribution based approaches via co training.

Individually, supervision of these frameworks comes completely from initially provided instances. Instead, allow both models to supervise each other.

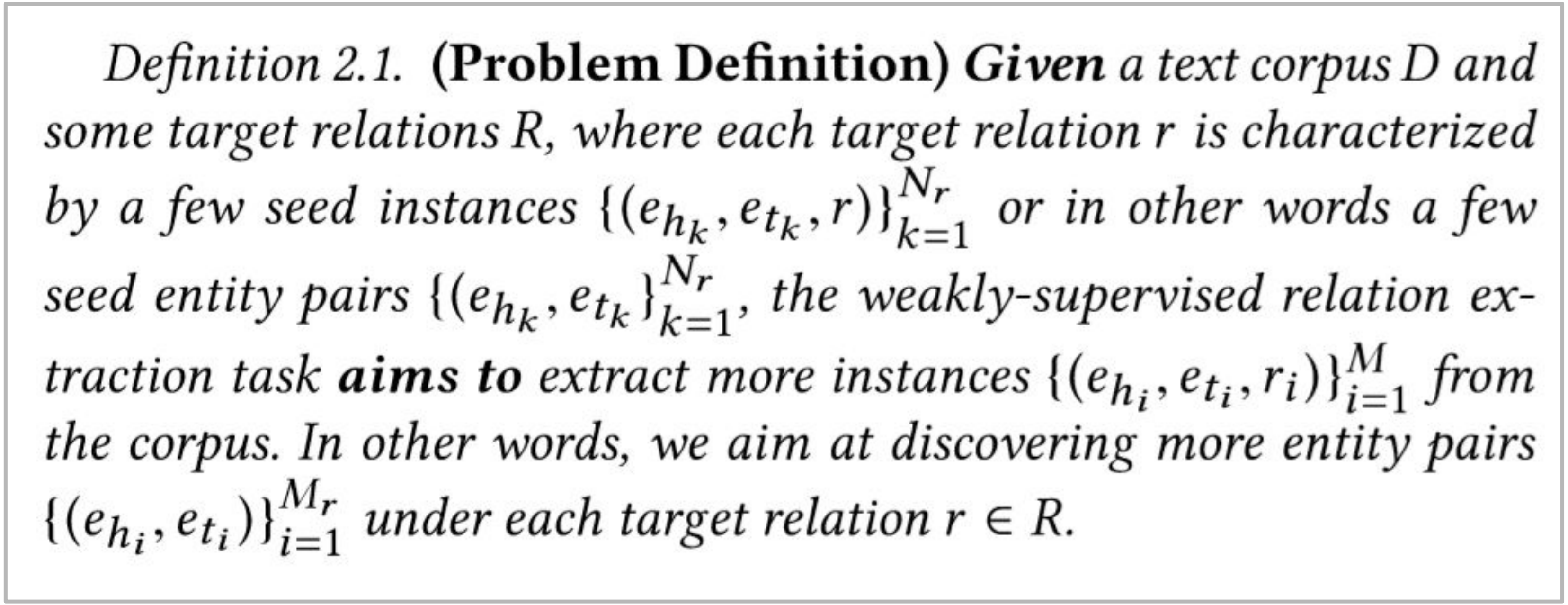
Pattern Module - Generator - Extracts candidate seed instances from corpus

Distributional Module - Discriminator - Evaluates each instance which serve as extra signals for generator to generate highly confident instances who would act as seeds from discriminator.

REPEL


Existing Integration Frameworks
Seeds
Pattern Module
Distributional Module
Our Co-training Framework
Seeds
Pattern Module
Distributional Module

REPEL - Problem definition



Definition 2.1. **(Problem Definition)** ***Given*** *a text corpus D and some target relations R, where each target relation r is characterized by a few seed instances* $\{(e_{h_k}, e_{t_k}, r)\}_{k=1}^{N_r}$ *or in other words a few seed entity pairs* $\{(e_{h_k}, e_{t_k}\}_{k=1}^{N_r}$*, the weakly-supervised relation extraction task* ***aims to*** *extract more instances* $\{(e_{h_i}, e_{t_i}, r_i)\}_{i=1}^{M}$ *from the corpus. In other words, we aim at discovering more entity pairs* $\{(e_{h_i}, e_{t_i})\}_{i=1}^{M_r}$ *under each target relation* $r \in R$*.*

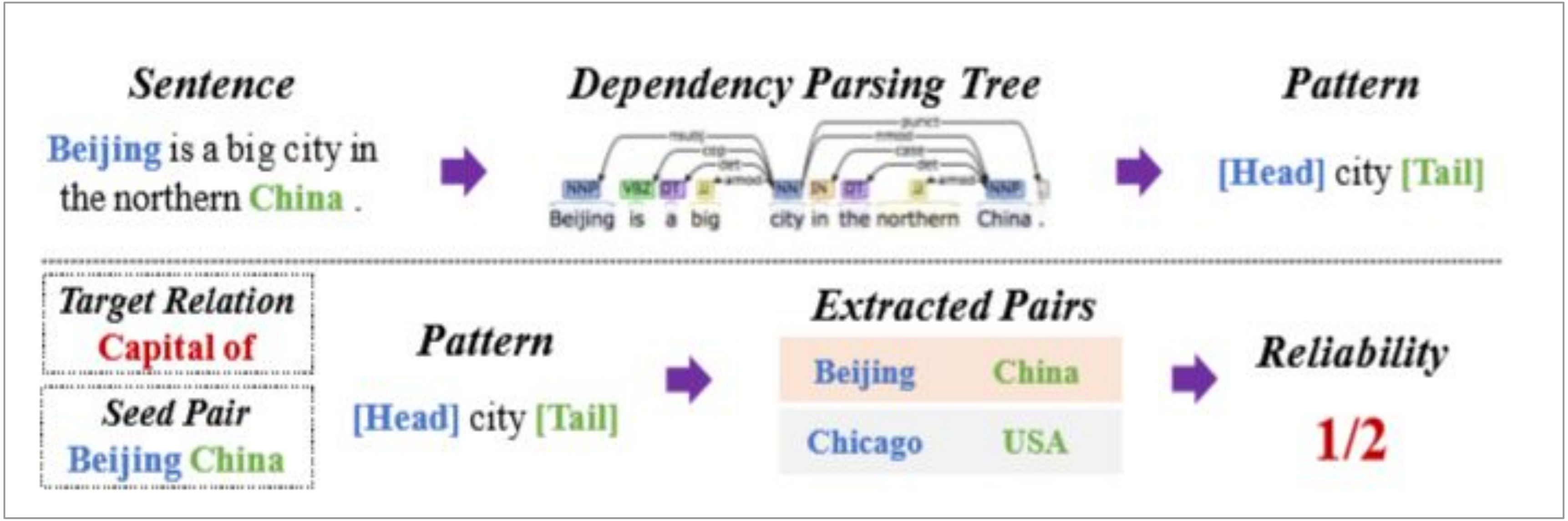
REPEL - Pattern Module

$$R(\pi) = \frac{|G(\pi) \cap S_{pair}|}{|G(\pi)|}$$

Π - Represents the patterns
R(π) -Represents the reliability of the p
G(π)- Represents all the entity pairs extracted by the pattern
S_{pair} - Represents the set of seed entity pairs under the target relation

If a pattern can extract many seed entity pairs under the target relation, then it will be considered reliable.

REPEL - Pattern Module



Sentence

Beijing is a big city in the northern China .

Dependency Parsing Tree

Pattern

[Head] city [Tail]

Target Relation

Capital of

Seed Pair

Beijing China

Pattern

[Head] city [Tail]

Extracted Pairs

Beijing	China
Chicago	USA

Reliability

1/2

REPEL - Distributional Module

Build a bipartite network between all entities and words.

The weight between the entity and a word is defined as the number of sentences in which they co-occur. For an entity 'e' and a word 'w', the conditional probability is inferred as

$$P(w|e) = \frac{\exp(\mathbf{x}_e \cdot \mathbf{c}_w)}{Z}$$

$\mathbf{x}_e$ is the vector representation of entity
$\mathbf{c}_w$ is the embedding vector of word w
Z is the normalization

Both the functions of the modules are simplified to form objective functions which are optimized.

REPEL - Joint Optimization

Another objective function is introduced whose goal is to encourage the agreement of both the modules. This objective function is referred to as the joint optimization problem introduced in the paper.

Set Expansion

Process of expanding the original set of seeds.

Previous studies like Google Set, SEAL and Lyretail give good quality results but seed oriented online data extraction is costly.

Offline approaches are categorized into

1. One time entity ranking
2. Iterative pattern based bootstrapping

Set Expansion - One time entity ranking

Assumes similar entities would appear in similar contexts

Make a one time ranking of candidate entities based on their distributional similarity with the seed entities.

Context required for the algorithms brought in by wikipedia lists and free text pattern and entity-entity distributional similarity is calculated based on all context features.

Limitation : Since all the contexts are used, high chances of seed intrusion error. Also, the extractions take place based on the initial seeds provided.

Set Expansion - Iterative pattern based bootstrapping

Starts from seed entities and extracts quality patterns, based on a predefined pattern scoring mechanism, and it then applies extracted patterns to obtain even higher quality entities using a different entity scoring method.

The process iteratively accumulates high quality patterns which are used for future iterations.

The quality of these patterns needs to be strictly maintained as seed intrusion problem can grow exponentially with each iteration causing semantic shift. Thus, the entity scoring methods are crucial and due to sensitivity towards the patterns, a huge amount of caution needs to be exercised.

Limitation : Having a perfect scoring mechanism is difficult due to diversity in the text data.

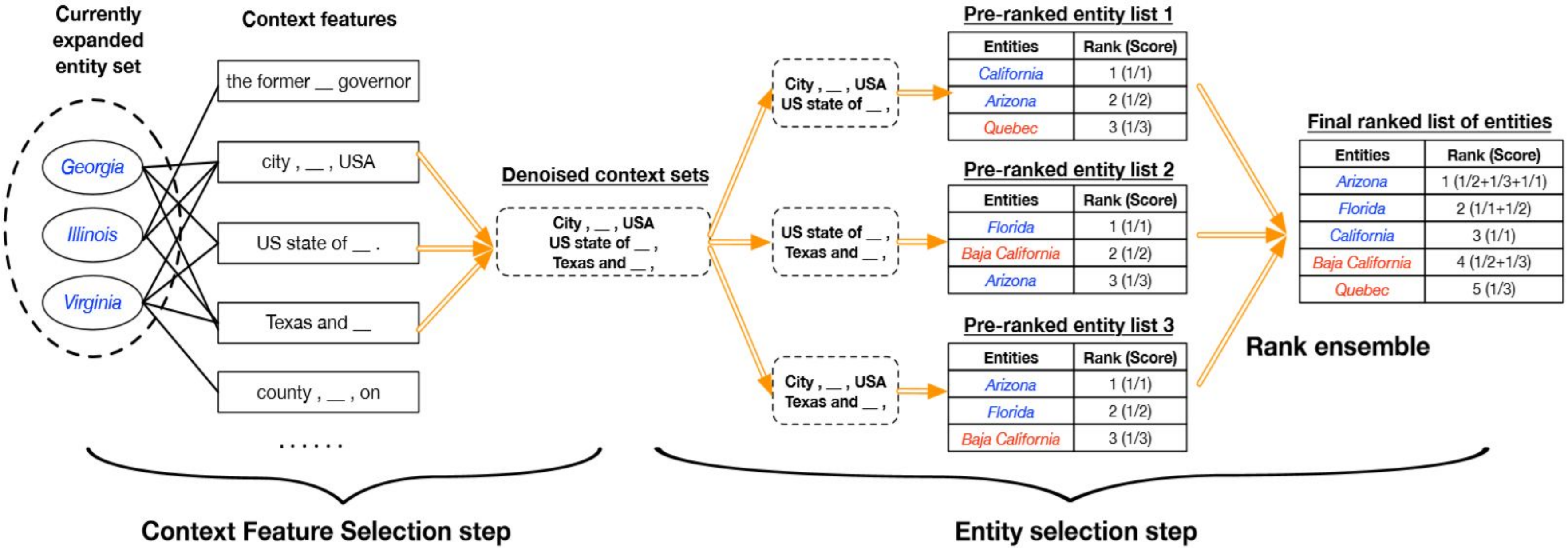
SetExpan - Novel framework

To address these challenges of previous approaches

To overcome the seed intrusion problem, it selects the context features based on distributional similarity instead of using all the available contexts.

To overcome the semantic drift problem, reset the feature pool at the beginning of each iteration. Makes use of an unsupervised ranking based ensemble method at each iteration to refine the entities.

SetExpan - An iteration



Pre-ranked entity list 1

Entities	Rank (Score)
California	1 (1/1)
Arizona	2 (1/2)
Quebec	3 (1/3)

Pre-ranked entity list 2

Entities	Rank (Score)
Florida	1 (1/1)
Baja California	2 (1/2)
Arizona	3 (1/3)

Pre-ranked entity list 3

Entities	Rank (Score)
Arizona	1 (1/1)
Florida	2 (1/2)
Baja California	3 (1/3)

Final ranked list of entities

Entities	Rank (Score)
Arizona	1 (1/2+1/3+1/1)
Florida	2 (1/1+1/2)
California	3 (1/1)
Baja California	4 (1/2+1/3)
Quebec	5 (1/3)

SetExpan:

- Data Model and Context Features
- Context dependent similarity
- Context feature selection
- Entity selection via rank ensemble

SetExpan - Data Model and Context features

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

Xe,c is the raw co-occurrence count between entity e and context feature c

|E| is the total number of candidate entities

Here, each entity e is treated as a “document” and each of its context feature c as a “term”.

Data is modeled as a bipartite graph, with candidate entities on one side and their context features on the other. These features are obtained using skip grams and coarse grained types.

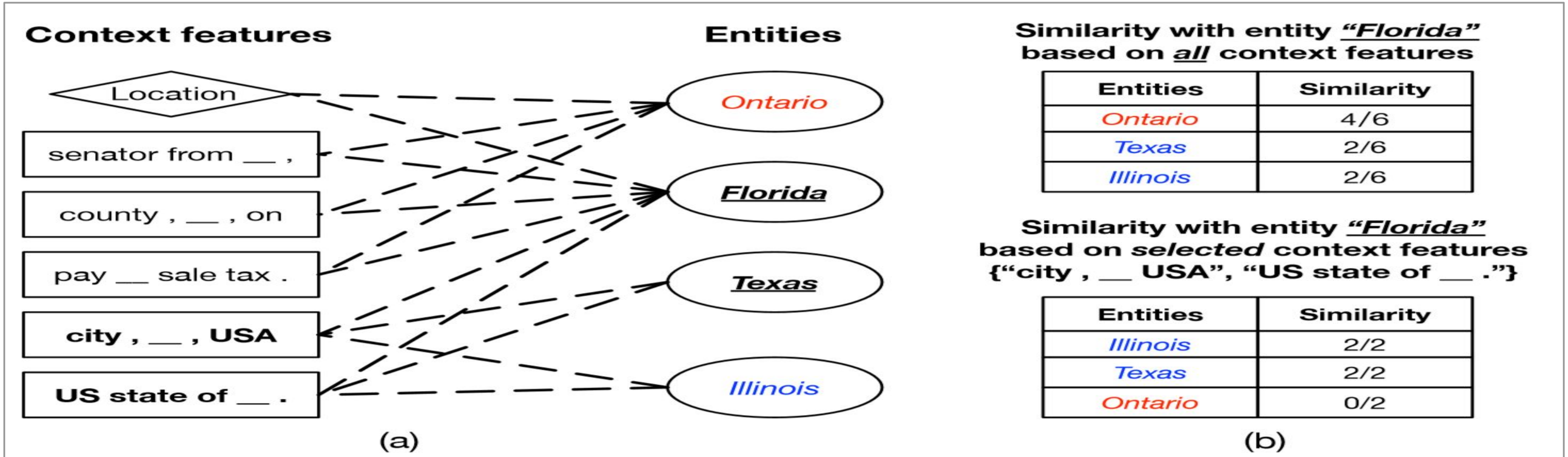
Between each pair of entity ‘e’ and context feature ‘c’ weight is calculated using TF-IDF transformation.

SetExpan - Context Dependent Similarity

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

To find the set of entities that are most similar to the current set. Weighted Jaccard similarity measure is used

Given a set of context features F, context-dependent similarity is calculated



Similarity with entity *"Florida"* based on ***all*** context features

Entities	Similarity
Ontario	4/6
Texas	2/6
Illinois	2/6

Similarity with entity *"Florida"* based on *selected* context features {"city , __ USA", "US state of __ ."}

Entities	Similarity
Illinois	2/2
Texas	2/2
Ontario	0/2

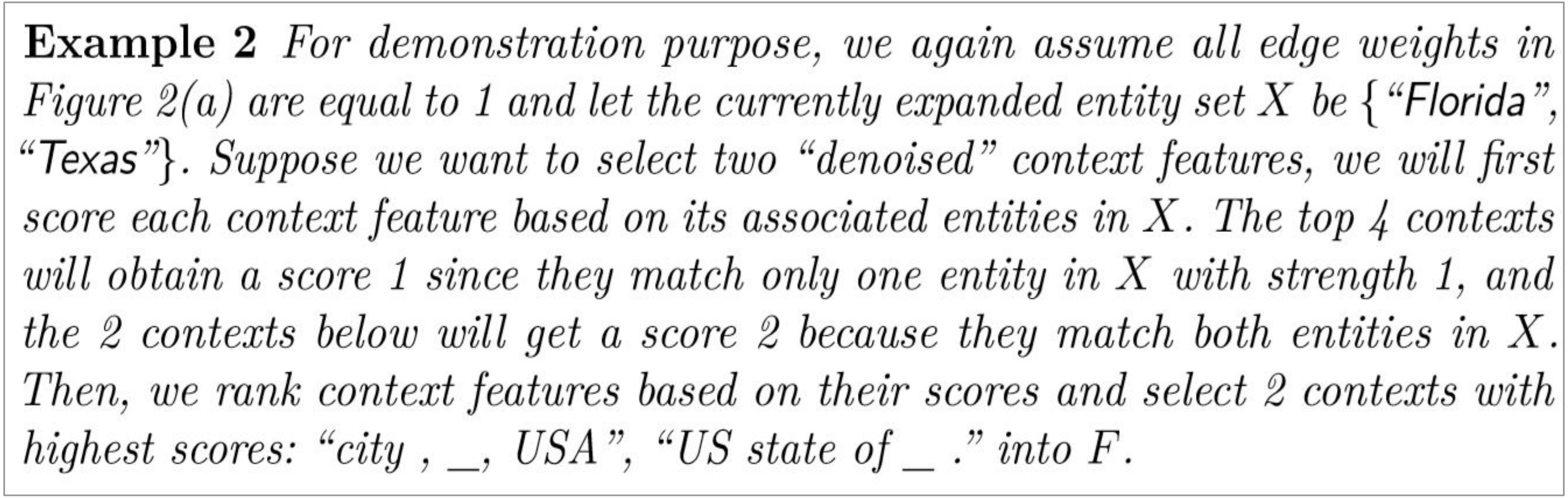
SetExpan - Context Feature Selection

Helps find a feature subset F∗ of fixed size Q that best “profiles” the target semantic class.

Therefore, given such F∗, the entity-entity similarity conditioned on it can best reflect their distributional similarity with regard to the target class.

However, this is an NP hard problem. Thus, 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 is used.

SetExpan - Context Feature Selection



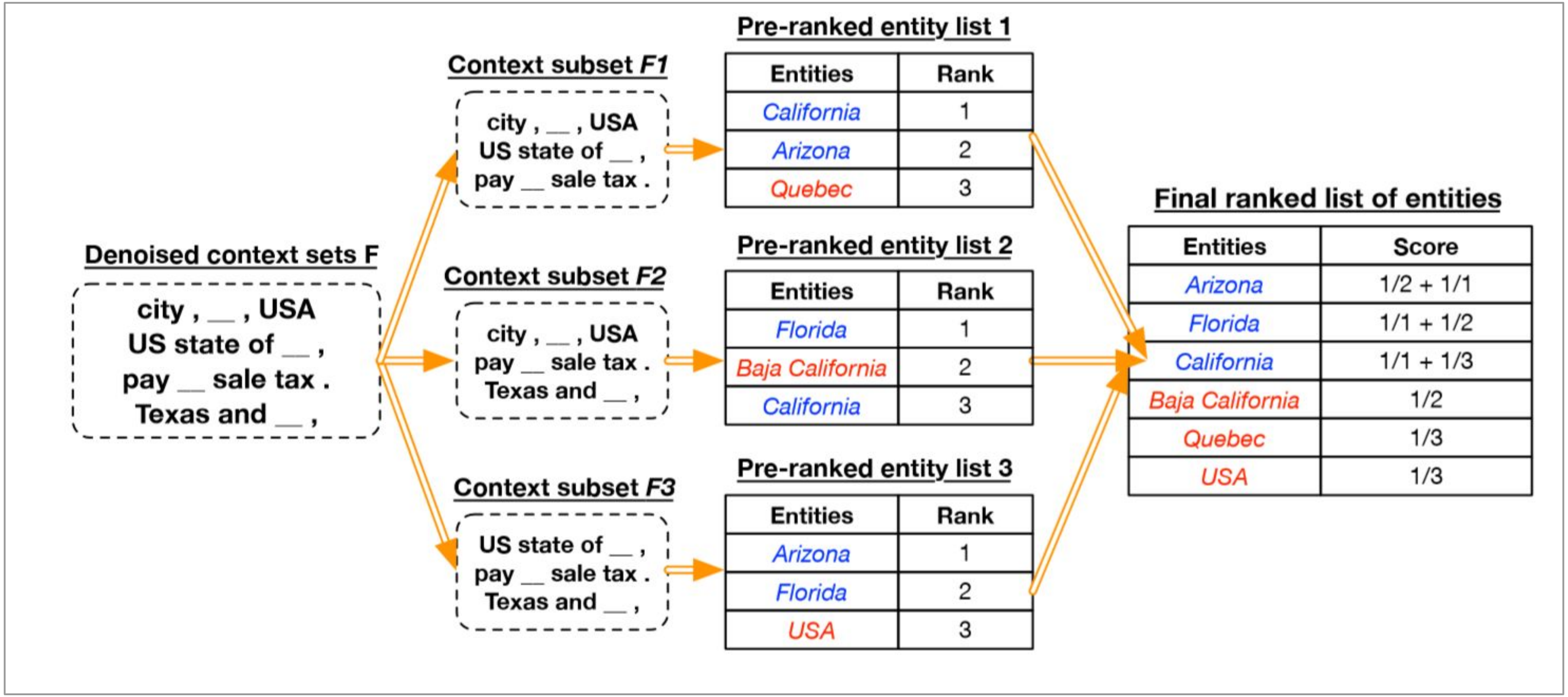
Example 2 *For demonstration purpose, we again assume all edge weights in Figure 2(a) are equal to 1 and let the currently expanded entity set X be {“Florida”, “Texas”}. Suppose we want to select two “denoised” context features, we will first score each context feature based on its associated entities 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.*

SetExpan - Entity selection via rank ensemble:

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

The algorithm ranks each candidate entity based on its score calculated by the following formula and then adds top-ranked ones into the expanded set.

SetExpan - Entity selection via rank ensemble:



Denoised context sets F

city , __ , USA
US state of __ ,
pay __ sale tax .
Texas and __ ,

Context subset *F1*

city , __ , USA
US state of __ ,
pay __ sale tax .

Context subset *F2*

city , __ , USA
pay __ sale tax .
Texas and __ ,

Context subset *F3*

US state of __ ,
pay __ sale tax .
Texas and __ ,

Pre-ranked entity list 1

Entities	Rank
California	1
Arizona	2
Quebec	3

Pre-ranked entity list 2

Entities	Rank
Florida	1
Baja California	2
California	3

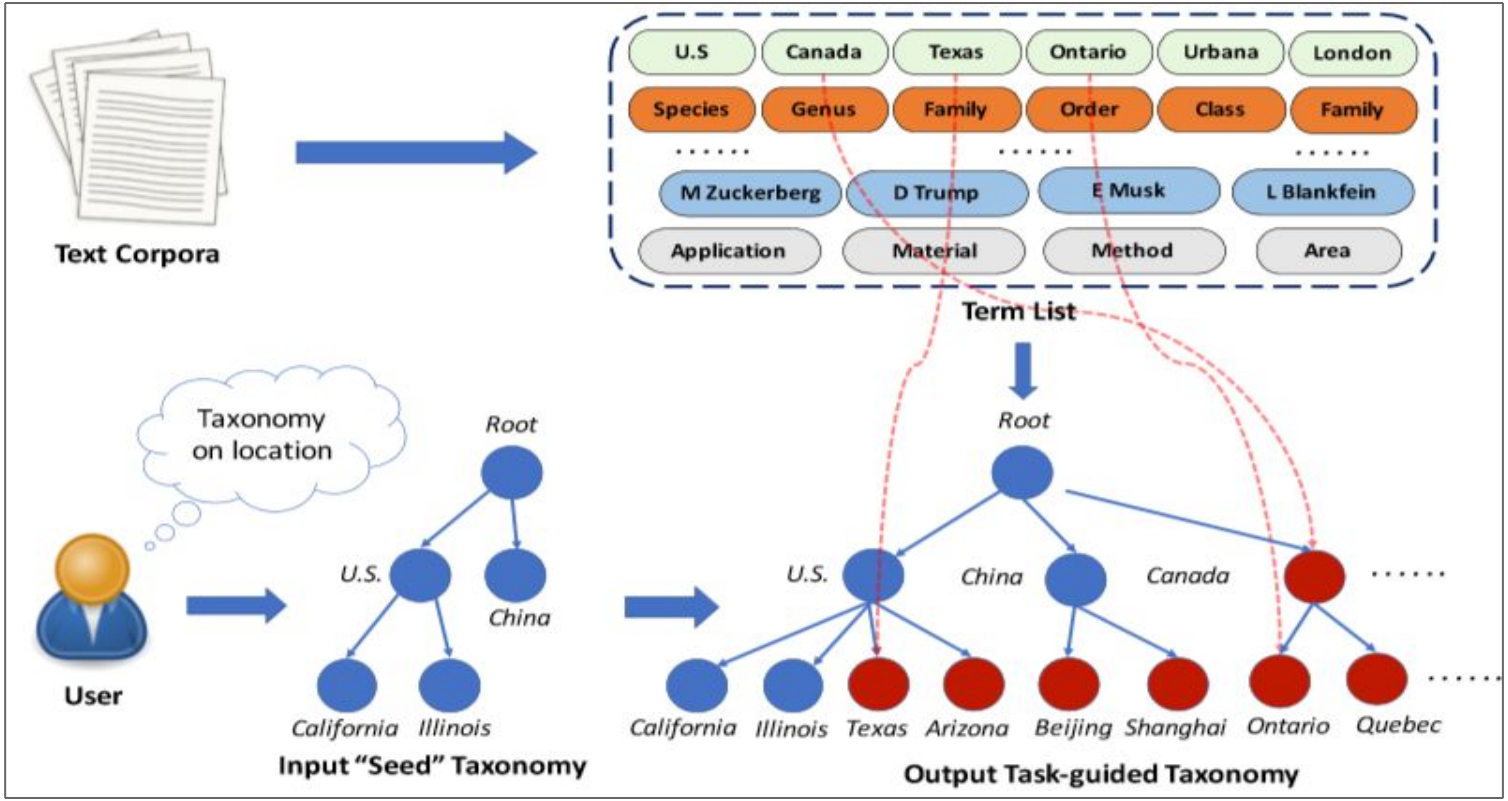
Pre-ranked entity list 3

Entities	Rank
Arizona	1
Florida	2
USA	3

Final ranked list of entities

Entities	Score
Arizona	1/2 + 1/1
Florida	1/1 + 1/2
California	1/1 + 1/3
Baja California	1/2
Quebec	1/3
USA	1/3

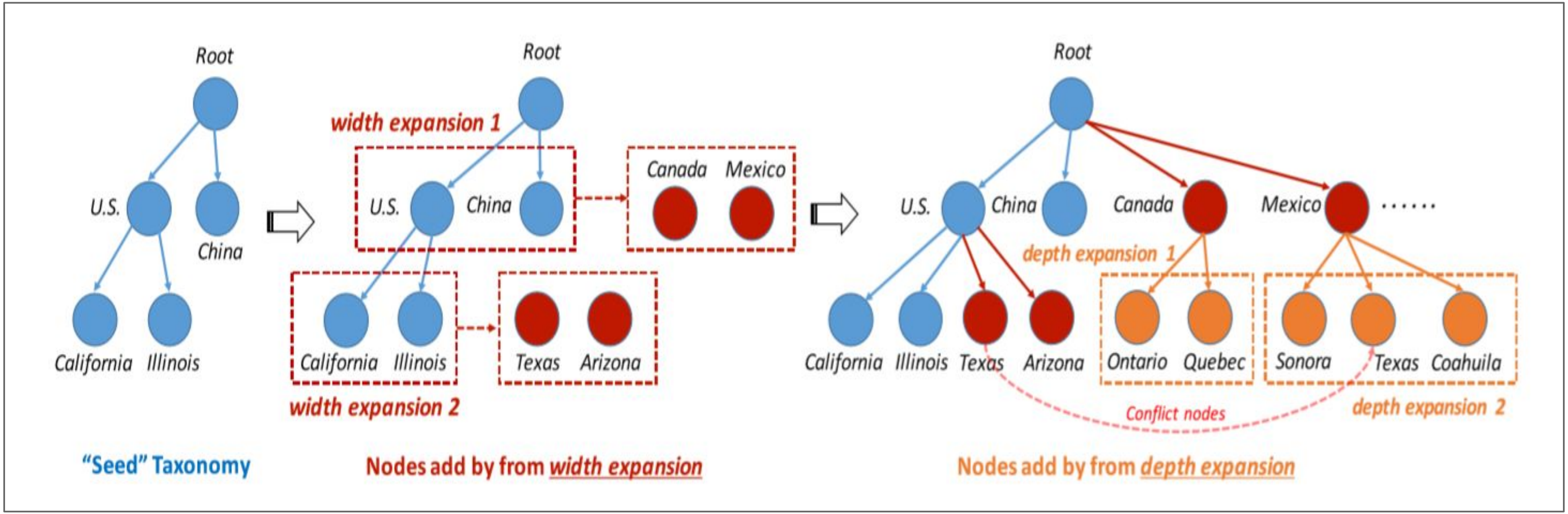
HiExpan Taxonomy Constructor:



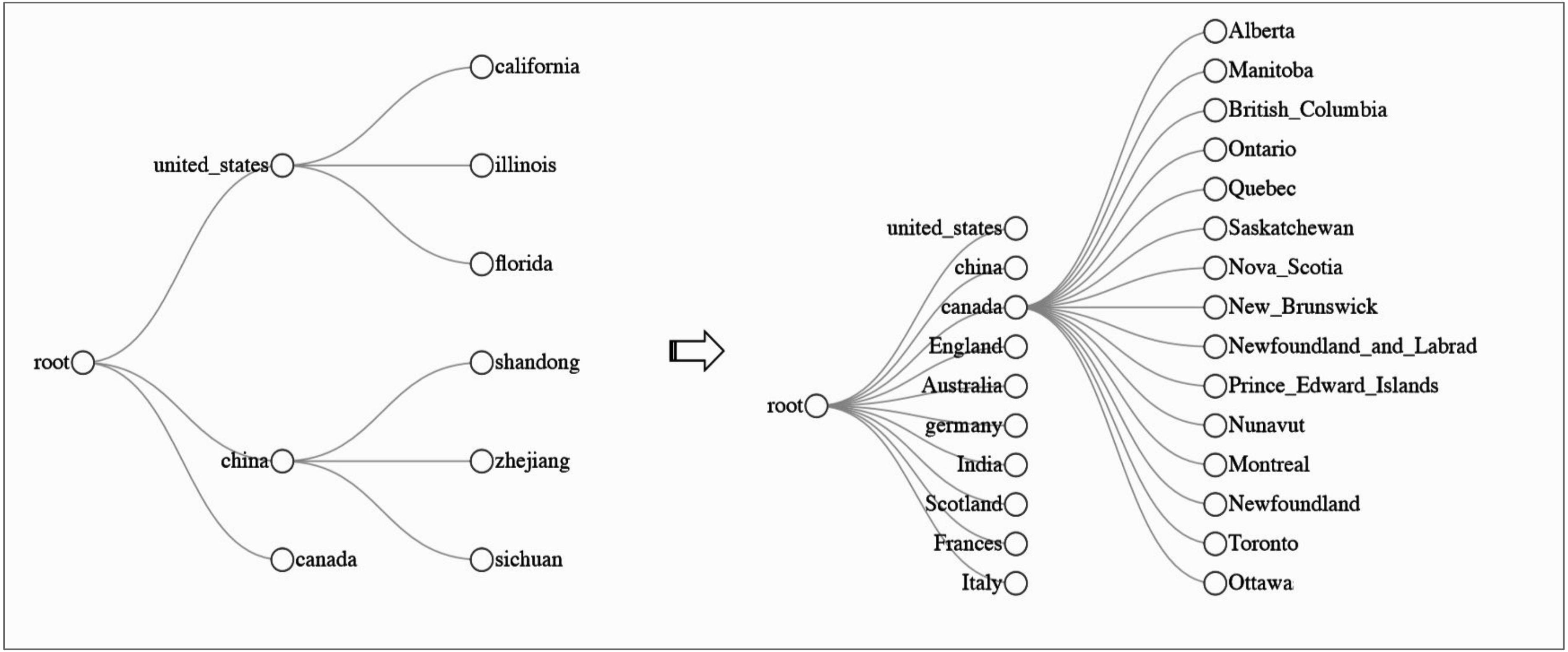
Uses SetExpan - Horizontal Expansion

Uses REPEL - Vertical Expansion

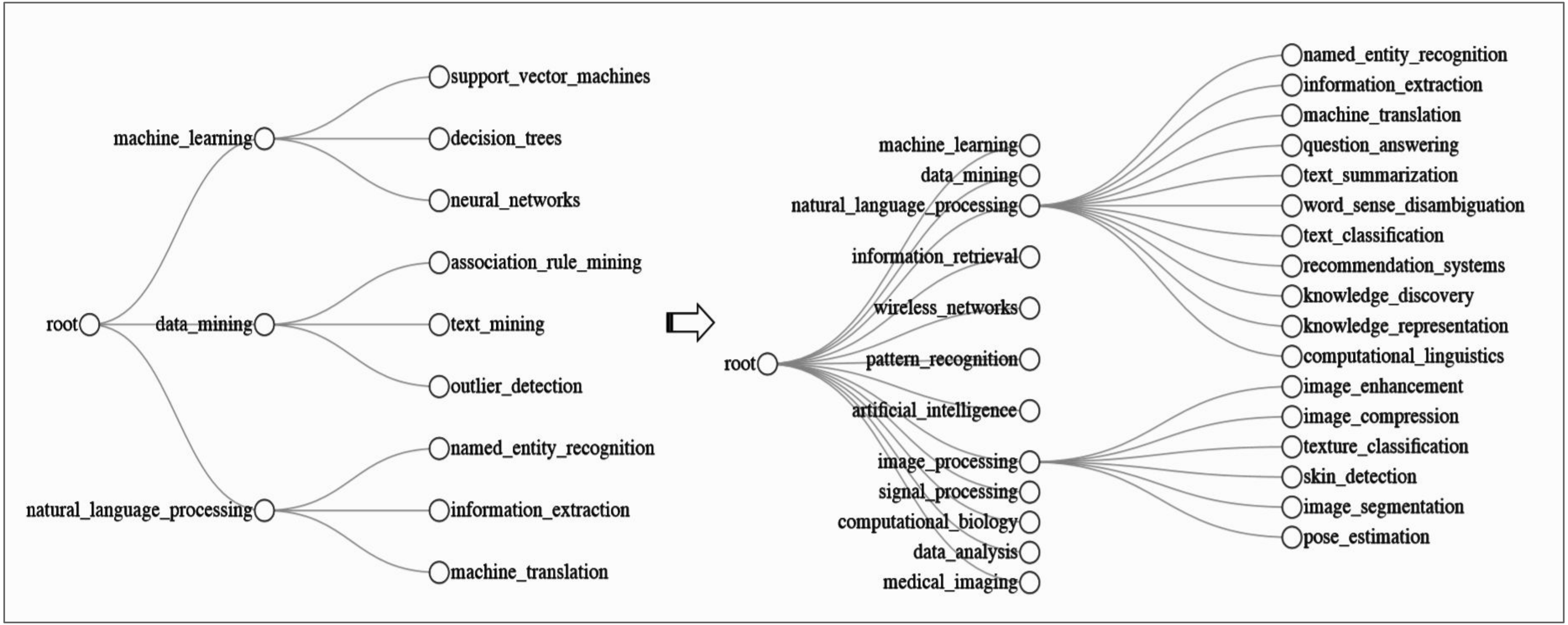
HiExpan Taxonomy Constructor:



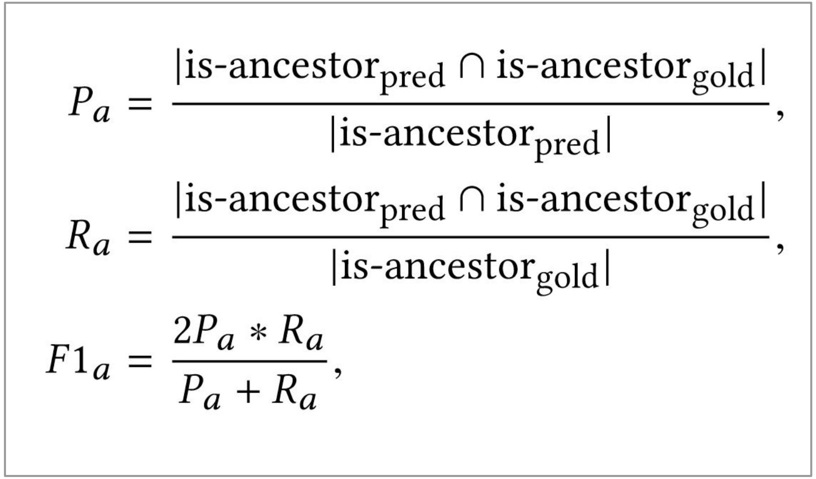
HiExpan - Results on WikiList



HiExpan - Results on DBLP



HiExpan - Quantitative Results:



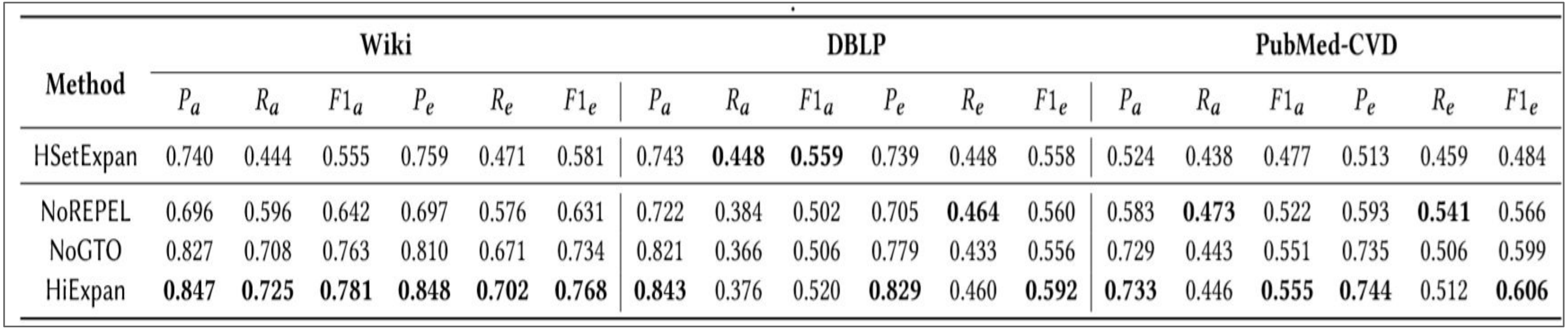
$$P_a = \frac{|\text{is-ancestor}_{\text{pred}} \cap \text{is-ancestor}_{\text{gold}}|}{|\text{is-ancestor}_{\text{pred}}|},$$

$$R_a = \frac{|\text{is-ancestor}_{\text{pred}} \cap \text{is-ancestor}_{\text{gold}}|}{|\text{is-ancestor}_{\text{gold}}|},$$

$$F1_a = \frac{2P_a * R_a}{P_a + R_a},$$

The quantitative results included Pa, Ra, F1a denote the ancestor-Precision, ancestor-Recall, and ancestor-F1-score, respectively. Similarly, edge-based metrics are Pe,Re,and F1e,respectively.

HiExpan - Quantitative Results:



Method	Wiki						DBLP						PubMed-CVD					
	P_a	R_a	$F1_a$	P_e	R_e	$F1_e$	P_a	R_a	$F1_a$	P_e	R_e	$F1_e$	P_a	R_a	$F1_a$	P_e	R_e	$F1_e$
HSetExpan	0.740	0.444	0.555	0.759	0.471	0.581	0.743	**0.448**	**0.559**	0.739	0.448	0.558	0.524	0.438	0.477	0.513	0.459	0.484
NoREPEL	0.696	0.596	0.642	0.697	0.576	0.631	0.722	0.384	0.502	0.705	**0.464**	0.560	0.583	**0.473**	0.522	0.593	**0.541**	0.566
NoGTO	0.827	0.708	0.763	0.810	0.671	0.734	0.821	0.366	0.506	0.779	0.433	0.556	0.729	0.443	0.551	0.735	0.506	0.599
HiExpan	**0.847**	**0.725**	**0.781**	**0.848**	**0.702**	**0.768**	**0.843**	0.376	0.520	**0.829**	0.460	**0.592**	**0.733**	0.446	**0.555**	**0.744**	0.512	**0.606**

1. HSetExpan - Iteratively applies SetExpan at each level of taxonomy. For lower levels it uses the child-parent similarity for deciding the best parent to attach the node.
2. NoREPEL - The HiExpan algorithm without the REPEL module. It instead uses a skip gram model for learning term embeddings.
3. NoGTO - HiExpan algorithm without the optimization module.
4. HiExpan - The proposed framework

What can be improved?

Problems:

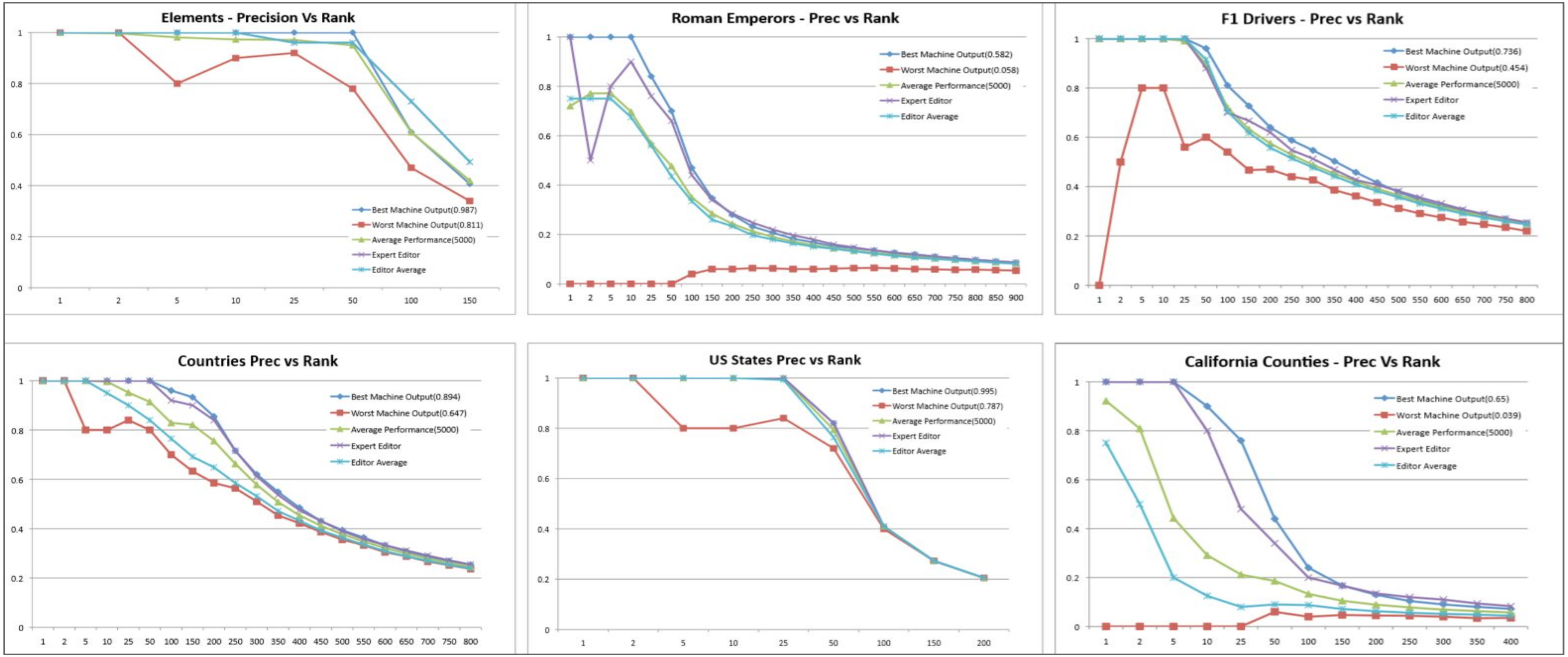
1. HiExpan tends to place synonyms at the same level of taxonomy as they share the semantic meanings and appear in similar contexts. These synonyms tend to make the generated taxonomy less informative, reducing the overall quality.
2. Domain specific corpus needs to be available for input.
   a. Difficult with new domains coming in frequently
   b. Corpus for domain widely available compared to corpus for subdomain causing problem in case of constructing a taxonomy of a subdomain.
3. Choice of seeds is subjective

What can be improved?

Solutions:

1. Need to incorporate “Knowledge + Context” in taxonomy construction. A hierarchical clustering approach for this has been proposed by Microsoft.
2. To remove the subjectivity of the seeds, need to filter them by removing ,
   a. Prototypicality - Words which are most representative of a domain. The high frequency of these words includes a lot of variety in contexts which may cause semantic drift in the generated tree.
   b. Ambiguity - Polysemy of seed can introduce seed intrusion in the taxonomy. This can be avoided by providing a domain specific corpus like in the case of HiExpan, but when it isn’t available this ambiguity needs to be removed.
   c. Coverage - Coverage of a seed set for a concept is the amount of semantic space which the seed shares in common with the semantic space defined by the concept.

Effect of choice of seeds:



Thank you!