SynSetExpan: An Iterative Framework for Joint Entity Set Expansion and Synonym Discovery

Jiaming Shen*[1], Wenda Qiu*[1], Jingbo Shang[2], Michelle Vanni[3], Xiang Ren[4], Jiawei Han[1]

[1]University of Illinois at Urbana-Champaign, [2]University of California San Diego

[3]U.S. Army Research Laboratory, [4]University of Southern California

Presented by Jiaming Shen @ EMNLP 2020

Paper Link: https://arxiv.org/abs/2009.13827

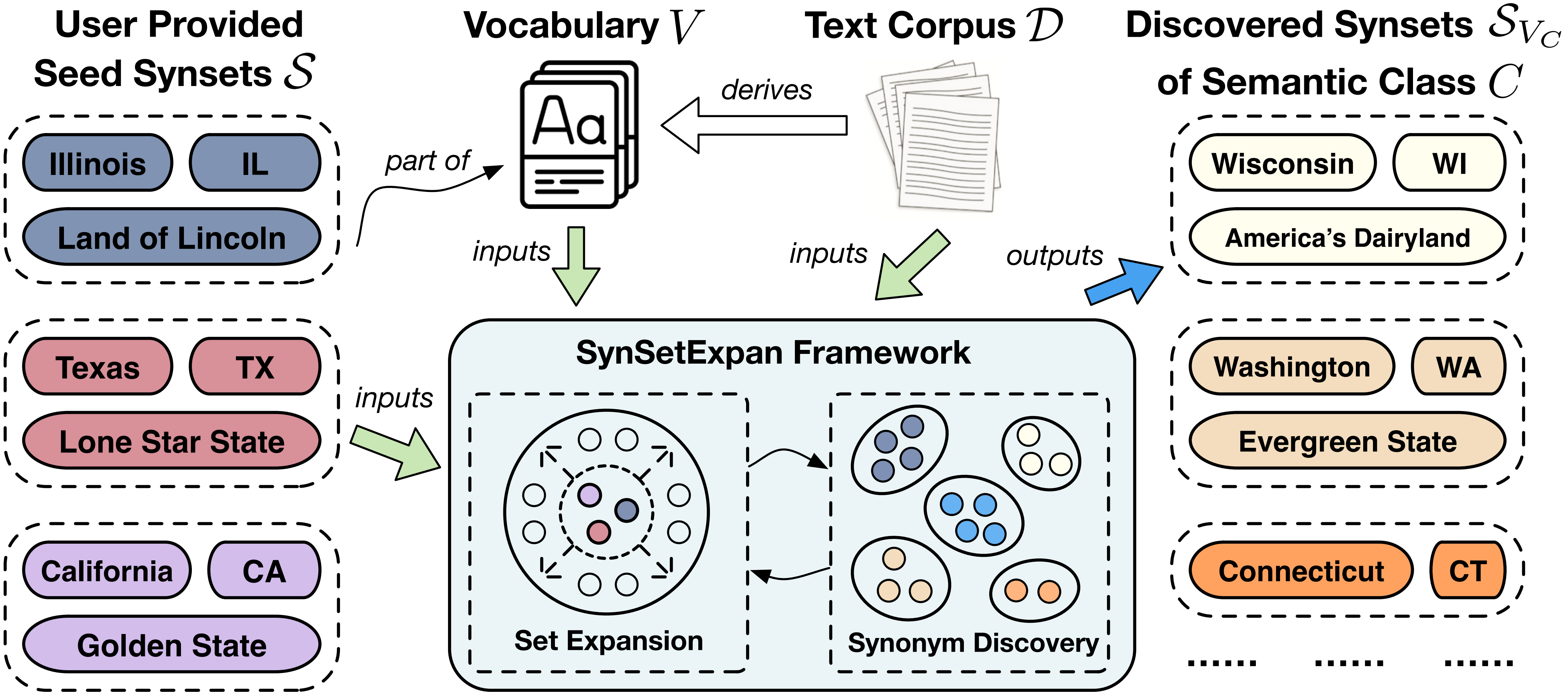
Introduction

- **Entity set expansion (ESE)** aims to expand a small number of seed entities into a larger set of entities that belong to the same semantic class
  - E.g.: {Illinois, California} → {Illinois, California, Florida, Arizona, …}
- **Entity Synonym discovery (ESD)** intends to group all terms that refer to the same real-world entity into a synonym set (in short *synset*)
  - E.g.: {America, USA}, {Illinois, IL, Land of Lincoln}, …
- Both tasks can benefit many entity-aware applications but previously they are regarded as two orthogonal tasks and accomplished independently

Introduction

- Entity set expansion and synonym discovery are tightly coupled
  - One entity can be the synonym of another entity *only if* they both belong to the same semantic class → **Set Expansion helps Synonym Discovery**
  - Knowing the class membership of one entity enables us to infer the class membership of all its synonyms → **Synonym Discovery helps Set Expansion**
- We develop **SynSetExpan**, a framework that jointly conducts two tasks and enables them to mutually enhance each other

Problem Formulation



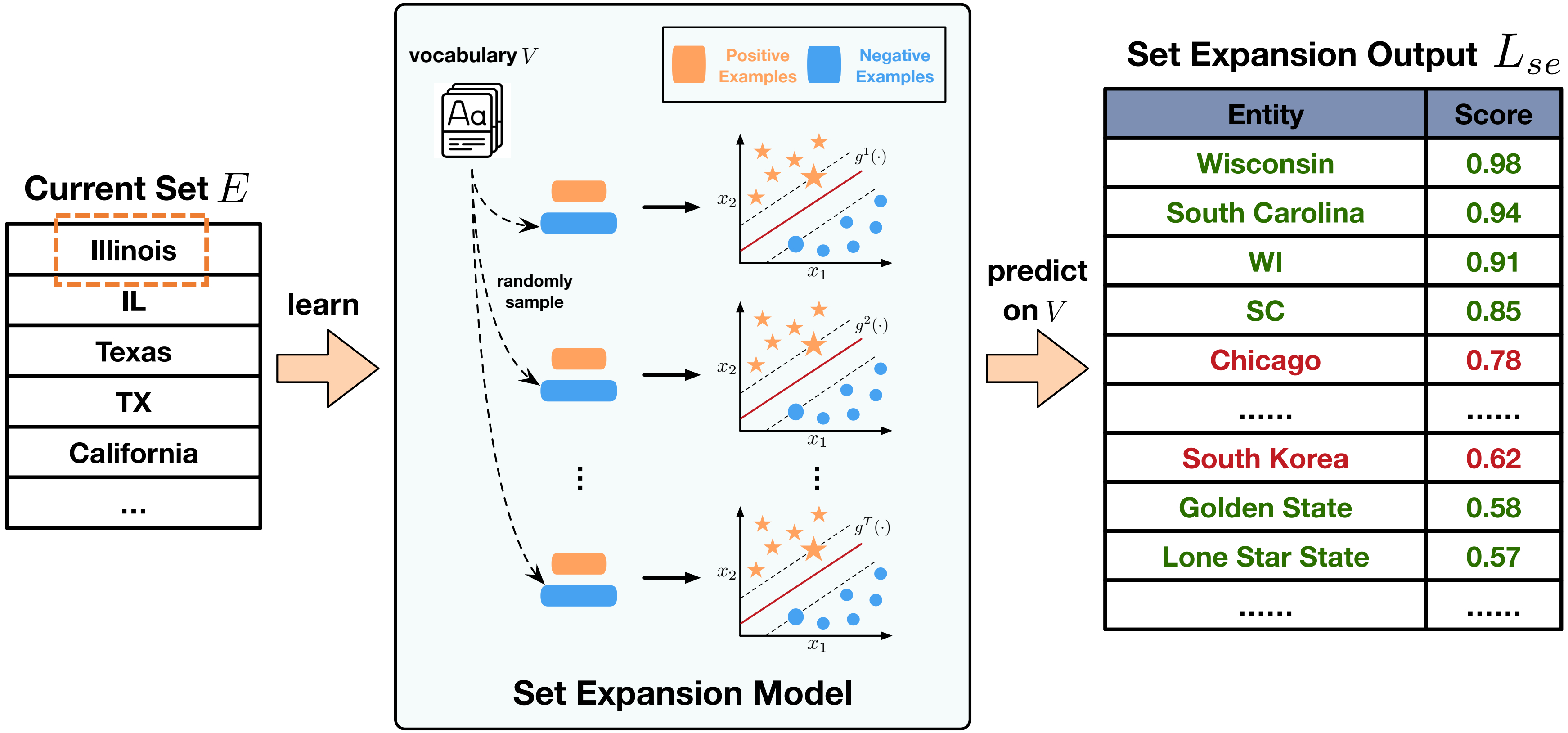
SynSetExpan Framework – Overview

- SynSetExpan is an iterative framework consisting of two models:
  - ***Set Expansion Model*** which predicts whether an entity belongs to the class
  - ***Synonym Discovery Model*** which predicts whether two entities are synonyms
- Before the main iteration, we learn a general synonym discovery model
  - This synonym discovery model is NOT tailed for a target semantic class
- Within the iterative process, we enable two models to mutually enhance each other ← **one of our main contributions**
- After the iterative process, we cluster entities into synsets

Set Expansion Model

- We learn an ***ensemble*** classifier based on *T=50* independently trained SVM classifiers with randomly sampled negative samples

Each entity is represented by a <u>*set*</u> of embeddings



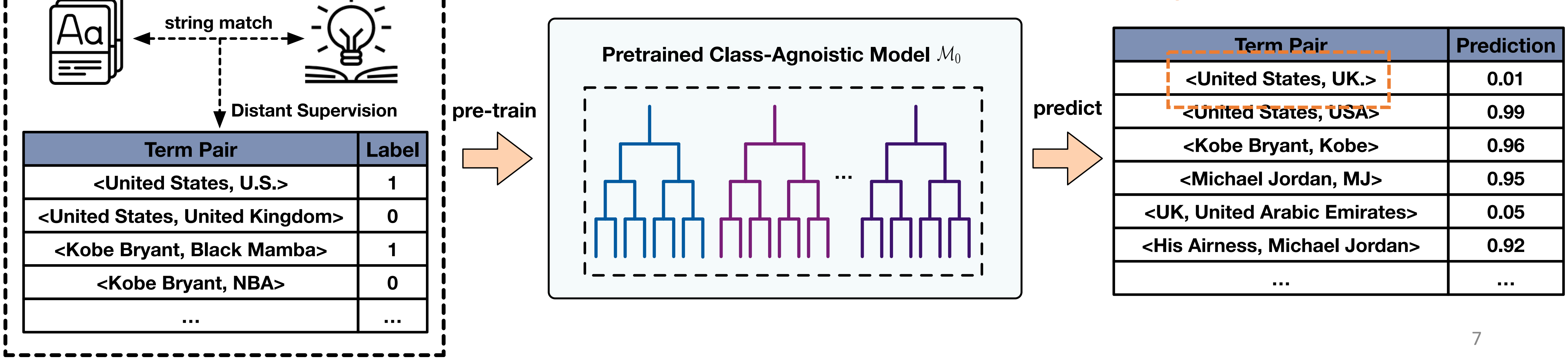
Synonym Discovery Model

- We learn an ***additive*** tree-based classifier
  - We derive *distant supervision* from KB
  - We manually define term pair features:
    - String-level features and Semantic features

Table 1: Entity pair features used in synset discovery model.

Feature Description	Example
IsPrefix	(Florida, FL) → 1
IsInitial	(North Carolina, NC) → 1
Edit distance	(North Carolina, Texas) → 13
Jaro-Winkler similarity	(Arizona, Texas) → 0.4476
Characters in common	(Lone Star State, Texas) → 2
Tokens in common	(North Carolina, South Carolina) → 1
Difference in #tokens	(Land of Lincoln, Illinois) → |3-1| = 2
Initial edit distance	(North Carolina, State of North Carolina) → 2
Longest token edit distance	(North Dakota, North Carolina) → 5
Cosine similarity of embedding	(Texas, Lone Star State) → 0.9
Transformed cosine similarities	(Texas, Lone Star State) → $[\frac{1}{0.9}, \sqrt{0.9}, (0.9)^2]$
Multiplication of two entities' PCA-reduced embedding	(Illinois, Land of Lincoln) → [0.006, 0.072, -0.008, 0.074, · · · , -0.004]

Each term pair is represented by the above set of features

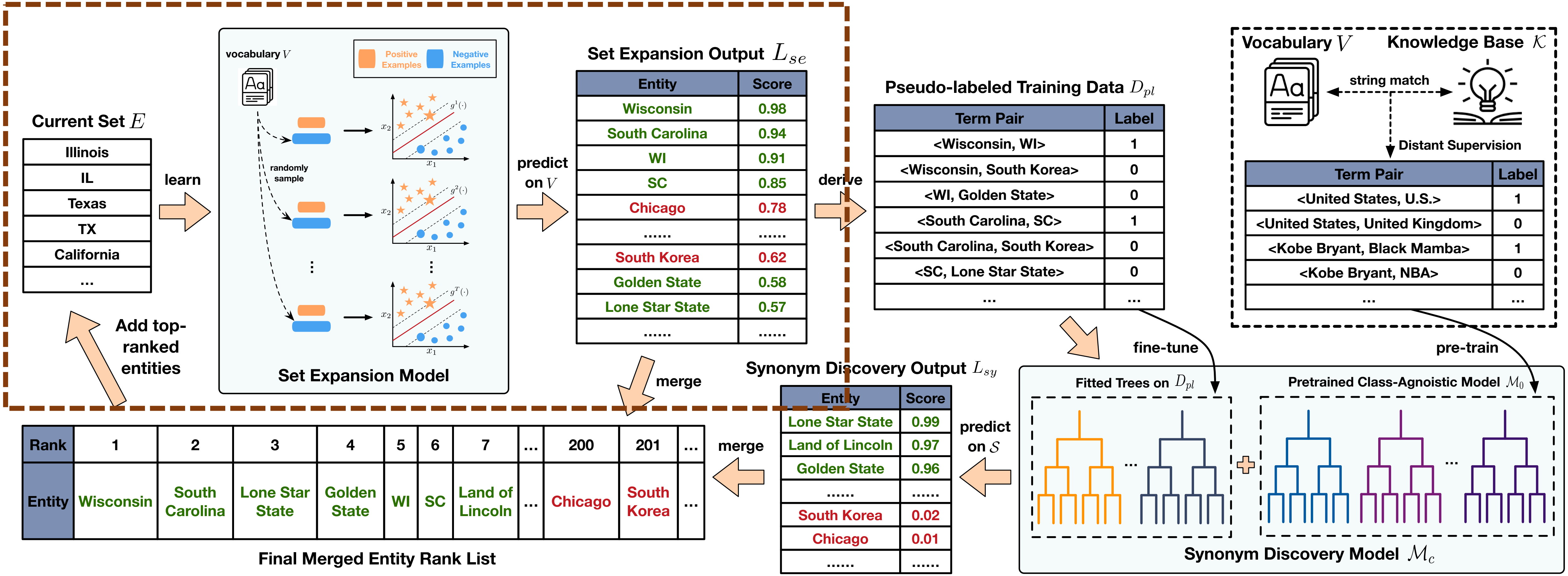


SynSetExpan Framework – Motivation Cases

- Standalone set expansion model may miss infrequent long-tail entities
  - *Example*: Starting from seed set {"Illinois", "IL", "Land of Lincoln", "Texas", "TX"}, we can only find state full names (*e.g.*, "Florida", "Arizona") but miss all state abbreviations (*e.g.*, "FL", "AZ") and slogans (*e.g.*, "America's Dairyland")
- Standalone synonym discovery model fixes feature weights for all classes
  - *Example*: For semantic class <u>*US States*</u>, many synonyms come from simple prefix (*e.g.*, "Florida" → "FL") and thus string-level features play a key role. For semantic class <u>*NBA Players*</u>, however, most entities get their synonyms from nicknames (*e.g.*, "Michael Jordan" → "His Airness") and thus we should emphasize more on embedding-based semantics features

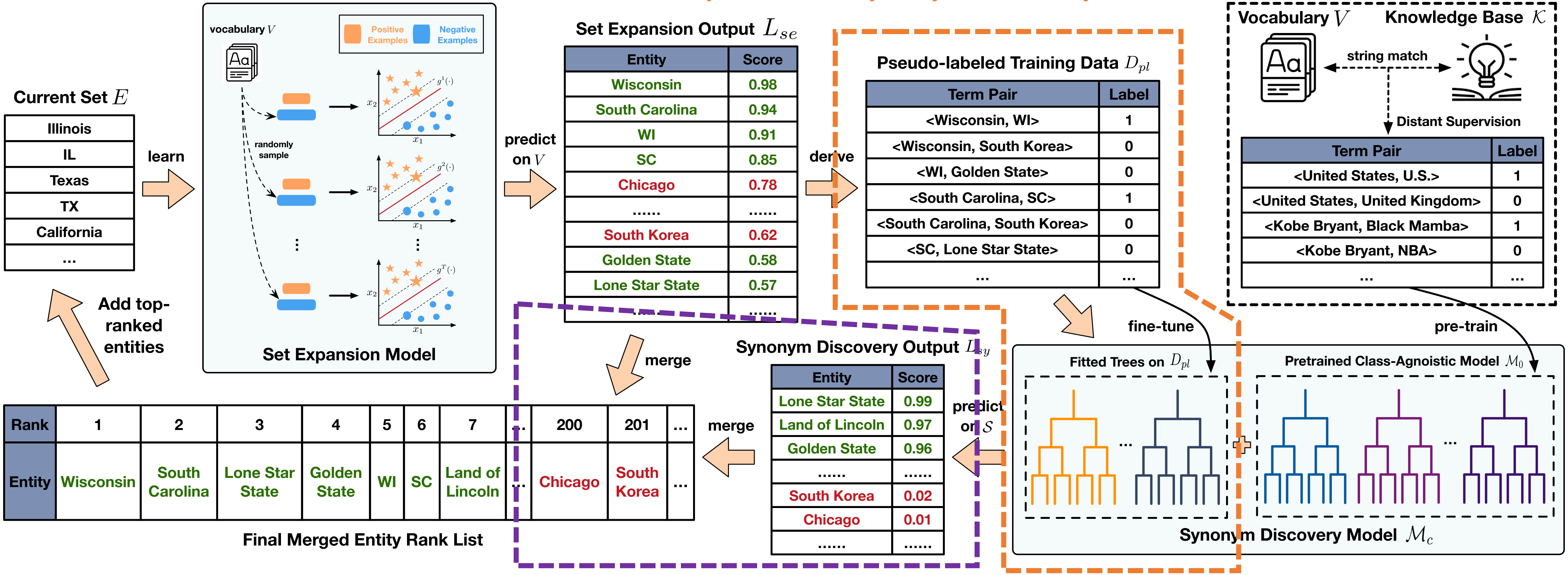
SynSetExpan Framework – Iterative Process

In each iteration, we start with Set Expansion Model



SynSetExpan Framework – Iterative Process

We use set expansion results to generate pseudo-labeled training data to fine-tune our pre-trained synonym discovery model



We use synonym discovery model to enrich set expansion model's original output results

Synonym-Enhanced Set Expansion (SE2) Dataset Construction

- The first crowd-sourcing Synonym-Enhanced Set Expansion (SE2) dataset:
  - A Wikipedia corpus of 1.9B tokens
  - A vocabulary of 1.5M frequent noun phrases
  - 60 semantic classes covering 6 different entity types
  - 1200 seed queries (20 queries per semantic class)

Class ID	Class Name	Class Type (Class Description)	Entities with Synsets
WikiTable-21	U.S. states	LOC (Locations)	[{"*Texas*", "*TX*", "*Lone Star State*"}, {"*Arizona*", "*AZ*"}, {"*California*", "*CA*", "*Golden State*"}, ...]
SemSearch-LS-3	Astronauts who landed on the Moon	PERSON (People)	[{"*Eugene Andrew Cernan*", "*Gene Cernan*"}, {"*Pete Conrad*"}, {"*Neil A. Armstrong*", "*Neil Armstrong*"}, ...]
Enriched-1	Apple Products	PRODUCT (Objects, vehicles, ...)	[{"*MacBook Pro*", "*MBP*"}, { "*iTouch*", "*iPod Touch*"}, ...]
WikiTable-27	Airports in British Isles	FAC (Facilities)	[{"*Ringway Airport*", "*Manchester Airport*" }, {"*RAF Exeter*", "*Exeter International Airport*"}, ...]
Enriched-4	NBA Teams	ORG (Organizations)	[{" *Washington Bullets*", "*Washington Wizards*" }, {"*Los Angeles Lakers*", "*L.A. Lakers*", "*Lakers*"}, ...]
INEX-XER-147	Chemical elements that are named after people	MISC (Miscellaneous classes)	[{"*Gadolinium*"}, {"*Seaborgium*", "*Element 106*"}, {"*Einsteinium*", "*Es99*"}, ...]

Experiments – Set Expansion (Settings)

- Datasets:
  - Previous benchmark datasets **Wiki** and **APR** (*Shen et al., 2017*)
  - Our constructed **SE2** dataset
- Compared Methods:
  - **One-time ranking methods**: EgoSet (*Rong et al., 2016*), SetExpander (*Mamou et al., 2018*), CaSE (*Yu et al., 2019*)
  - **Iterative methods**: SetExpan (*Shen et al., 2017*), MCTS (*Yan et al., 2019*), SetCoExpan (*Huang et al., 2020*), CGExpan (*Zhang et al., 2020*)
  - **Our proposed methods**: SynSetExpan, SynSetExpan-NoSYN
- Evaluation Metrics:
  - MAP@{10, 20, 50}

Experiments – Set Expansion (Overall Results)

- Overall SynSetExpan outperforms other baseline methods
- Adding synonym information helps

Methods	SE2			Wiki			APR		
	MAP@10	MAP@20	MAP@50	MAP@10	MAP@20	MAP@50	MAP@10	MAP@20	MAP@50
Egoset (Rong et al., 2016)	0.583	0.533	0.433	0.904	0.877	0.745	0.758	0.710	0.570
SetExpan (Shen et al., 2017)	0.473	0.418	0.341	0.944	0.921	0.720	0.789	0.763	0.639
SetExpander (Mamou et al., 2018b)	0.520	0.475	0.397	0.499	0.439	0.321	0.287	0.208	0.120
MCTS (Yan et al., 2019)	—	—	—	0.980	0.930	0.790	0.960	0.900	0.810
CaSE (Yu et al., 2019c)	0.534	0.497	0.420	0.897	0.806	0.588	0.619	0.494	0.330
SetCoExpan (Huang et al., 2020)	—	—	—	0.976	0.964	0.905	0.933	0.915	0.830
CGExpan (Zhang et al., 2020)	0.601	0.543	0.438	**0.995**	**0.978**	0.902	**0.992**	**0.990**	0.955
SynSetExpan-NoSYN	0.612	0.567	0.484	0.991	**0.978**	**0.904**	0.985	**0.990**	**0.960**
SynSetExpan	**0.628***	**0.584***	**0.502***	—	—	—	—	—	—

Experiments – Set Expansion (Detailed Results)

- SynSetExpan outperforms its non-synonym version in most cases
- Improvements are more significant in the long-tail end

Class Type	MAP@10	MAP@20	MAP@50
Person	86.7%	80.0%	93.3%
Organization	83.3%	83.3%	100%
Location	69.2%	65.4%	80.8%
Facility	85.7%	71.4%	100%
Product	100%	66.7%	100%
Misc	66.7%	66.7%	100%
Overall	**78.3%**	**71.7%**	**90.0%**

Table 4: Ratio of semantic classes on which SynSetExpan outperforms SynSetExpan-NoSYN.

SynSetExpan vs. Other	MAP@10	MAP@20	MAP@50
vs. CGExpan	78.9%	85.4%	93.8%
vs. SynSetExpan-NoSYN	72.7%	83.0%	91.4%

Table 5: Ratio of seed queries from the SE2 dataset on which the first method outperforms the second one.

Experiments – Synonym Discovery (Settings)

- Datasets:
  - Previous benchmark **PubMed** dataset (*Qu et al., 2017*): 10,486 positive synonym pairs and 193,162 negative synonym pairs
  - Our proposed **SE2** dataset: 3,067 positive pairs and 57,119 negative pairs
- Compared Methods:
  - **Previous methods**: SVM, XGBoost-(stringOnly & embedOnly), DPE (*Qu et al., 2017*), SynSetMine (*Shen et al., 2019*)
  - **Our proposed methods**: SynSetExpan, SynSetExpan-NoFT,
- Evaluation Metrics:
  - **Threshold-free metrics**: Average Precision (**AP**), Area Under the ROC Curve (**AUC**)
  - **Threshold-aware metric**: **F1** @ threshold = 0.5

Experiments – Synonym Discovery (Overall Results)

- Overall SynSetExpan outperforms other baseline methods
- Using set expansion results for fine-tuning helps

Method	**SE2**			**PubMed**		
	AP	AUC	F1	AP	AUC	F1
SVM	0.1870	0.8547	0.3300	0.2250	0.8206	0.4121
XGBoost-stringOnly [8]	0.7654	0.9696	0.6389	0.5012	0.8625	0.4968
XGBoost-embedOnly [8]	0.4762	0.8750	0.4810	0.4906	0.9190	0.5388
SynSetMine [34]	0.7562	0.9782	0.6347	0.6757	0.9453	0.6287
DPE [29]	0.7972	0.9792	0.6392	0.6338	0.8979	0.6038
SynSetExpan-NoFT	0.8197	0.9844	0.7159	0.6615	0.9445	0.6204
SynSetExpan	**0.8736**	**0.9953**	**0.7592**	**0.7152**	**0.9695**	**0.6388**

Experiments – Synonym Discovery (Case Studies)

- Entities in green are those entities discovered only by SynSetExpan after the fine-tuning step

Class: Astronauts who walked on the Moon	**Class:** Chinese 1st Level Administrative divisions	**Class:** War involving USA	**Class:** Airport in British Isles	**Class:** Apple Product	**Class:** NBA Teams
{Neil Armstrong, **Neil A. Armstrong**}	{Tibet, **Xizang Province**}	{WW1, WWI, First World War}	{London Heathrow, Heathrow Airport}	{Apple iPhone, **iPhone**, **iPhones**, Apple's iPhone}	{Lakers, **L.A. Lakers**, Los Angeles Lakers}
{Gene Cernan, **Eugene Cerne**}	{Fujian, **Fujian Province**}	{World War II, WWII, Second World War}	{Gatwick Airport, London-Gatwick, LGW, **EGKK**}	{Apple Watch, **iWatch**}	{**St. Louis Hawks**, Atlanta Hawks}
{Pete Conrad, Charles Conrad}	{Inner Mongolia, Nei Mongol}	{Gulf War, **Operation Desert Storm**}	{**Exeter Airport, EXT**}	{iPad Pro}	{New Jersey Nets, Brooklyn Nets}
......	......	......	......	...	......

Conclusions & Future Work

- Conclusions:
  - Set expansion and synonym discovery are two tightly coupled tasks and they can mutually enhance each other
  - Our proposed SynSetExpan is effective for both tasks
- Future Work
  - Integrate synonym enhancement idea with BERT-based ESE methods
  - Multi-faceted set expansion
  - Contextualized set expansion

References

- **Shen et al 2017**: Jiaming Shen, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, and Jiawei Han. 2017. Setexpan: Corpus- based set expansion via context feature selection and rank ensemble. In *ECML/PKDD*.
- **Rong et al., 2016***: Xin Rong, Zhe Chen, Qiaozhu Mei, and Eytan Adar. 2016. Egoset: Exploiting word ego-networks and user-generated ontology for multifaceted set expansion. In WSDM*
- **Mamou et al., 2018**: Jonathan Mamou, Oren Pereg, Moshe Wasserblat, Alon Eirew, Yael Green, Shira Guskin, Peter Izsak, and Daniel Korat. 2018b. Term set expansion based nlp architect by intel ai lab. In EMNLP.
- **Yu et al., 2019**: Puxuan Yu, Zhiqi Huang, Razieh Rahimi, and James Allan. 2019b. Efficient corpus-based set expansion with lexico-syntactic features and distributed representations. In SIGIR.
- **Huang et al., 2020**: Jiaxin Huang, Yiqing Xie, Yu Meng, Jiaming Shen, Yunyi Zhang, and Jiawei Han. 2020. Guiding corpus- based set expansion by auxiliary sets generation and co-expansion
- **Yan et al., 2019***:* Lingyong Yan, Xianpei Han, Le Sun, and Ben He. 2019. Learning to bootstrap for entity set expansion. In *EMNLP*.
- **Zhang et al., 2020**: Yunyi Zhang, Jiaming Shen, Jingbo Shang, and Jiawei Han. 2020. Empower entity set expansion via lan- guage model probing. In ACL.

References

- **Qu et al 2017**: Meng Qu, Xiang Ren, and Jiawei Han. 2017. Automatic synonym discovery with knowledge bases. In KDD.
- **Shen et al., 2019***: Jiaming Shen, Ruiilang Lv, Xiang Ren, Michelle Vanni, Brian Sadler, and Jiawei Han. 2019. Mining entity synonyms with efficient neural set generation. In AAAI.*

Thanks for your attention

Questions?

Email: js2@illinois.edu

Paper Link: https://arxiv.org/abs/2009.13827