Part II: Revisiting Text Mining Fundamentals with Pre-Trained Language Models

AAAI 2022 Tutorial

Pre-Trained Language Representations for Text Mining

Yu Meng, Jiaxin Huang, Yu Zhang, Jiawei Han

Computer Science, University of Illinois at Urbana-Champaign

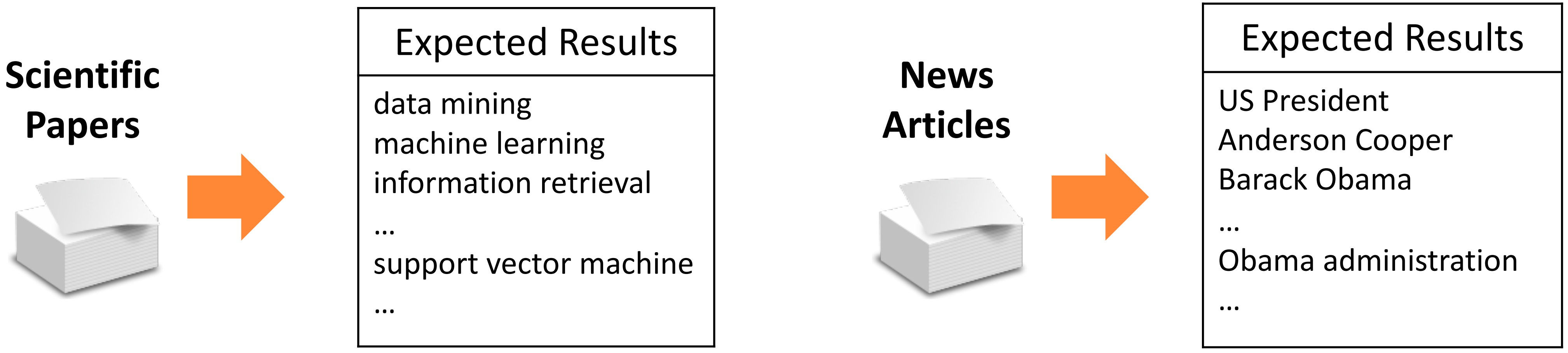
February 23, 2022

Outline

- Phrase Mining
  - Phrase Mining Introduction
  - UCPhrase: Unsupervised Context-aware Quality Phrase Tagging
- Named Entity Recognition
- Taxonomy Construction

Previous Phrase Mining/Chunking Models

- Identifying and understanding quality phrases from context is a fundamental task in text mining.



Scientific Papers

Expected Results
data mining machine learning information retrieval … support vector machine …

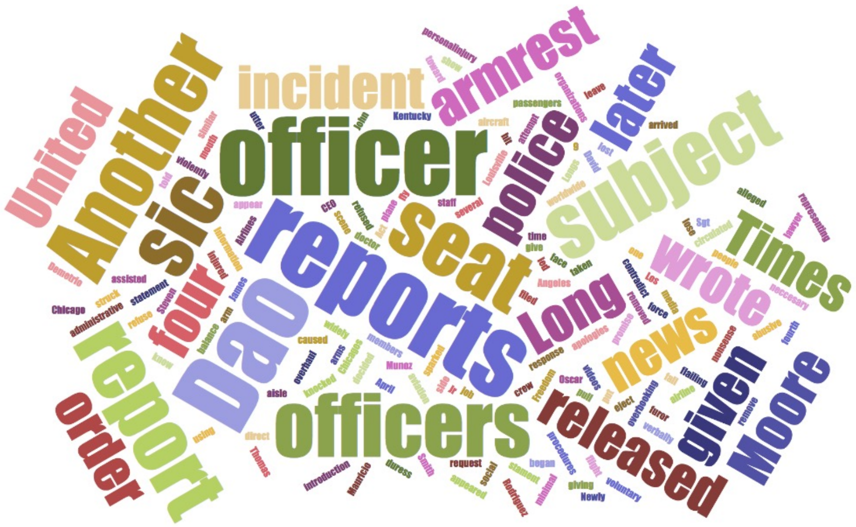
News Articles

Expected Results
US President Anderson Cooper Barack Obama … Obama administration …

- Quality phrases refer to informative multi-word sequences that "*appear consecutively in the text, forming a complete semantic unit in certain contexts or the given document*" [1].

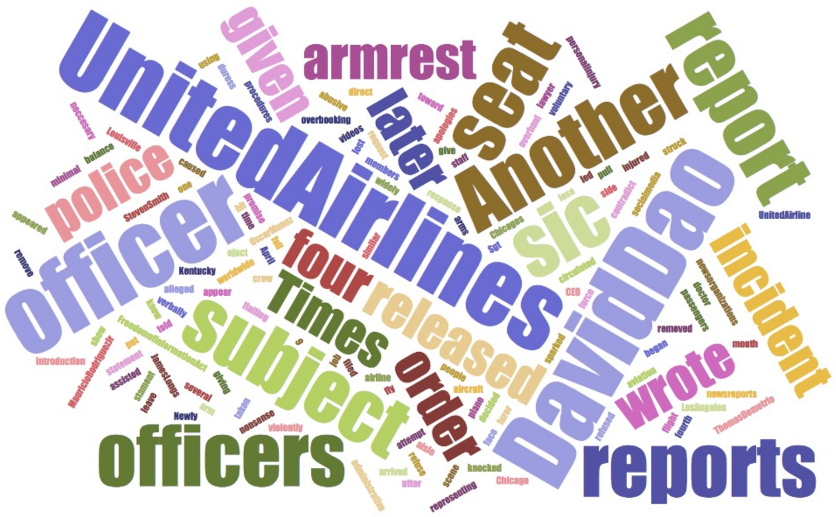
[1] Geoffrey Finch. 2016. Linguistic terms and concepts. Macmillan International Higher Education

Why Phrase Mining?



w/o phrase mining

- What’s “United”?
- Who’s “Dao”?



w/ phrase mining

- United Airline!
- David Dao!

- Applications in NLP, IR, Text Mining
  - Text Classification
  - Indexing in search engine
  - Keyphrases for topic modeling
  - Text Summarization

Outline

- Phrase Mining
  - Phrase Mining (introduction)
  - UCPhrase: Unsupervised Context-aware Quality Phrase Tagging
- Named Entity Recognition
- Taxonomy Construction

Previous Phrase Mining/Chunking Models

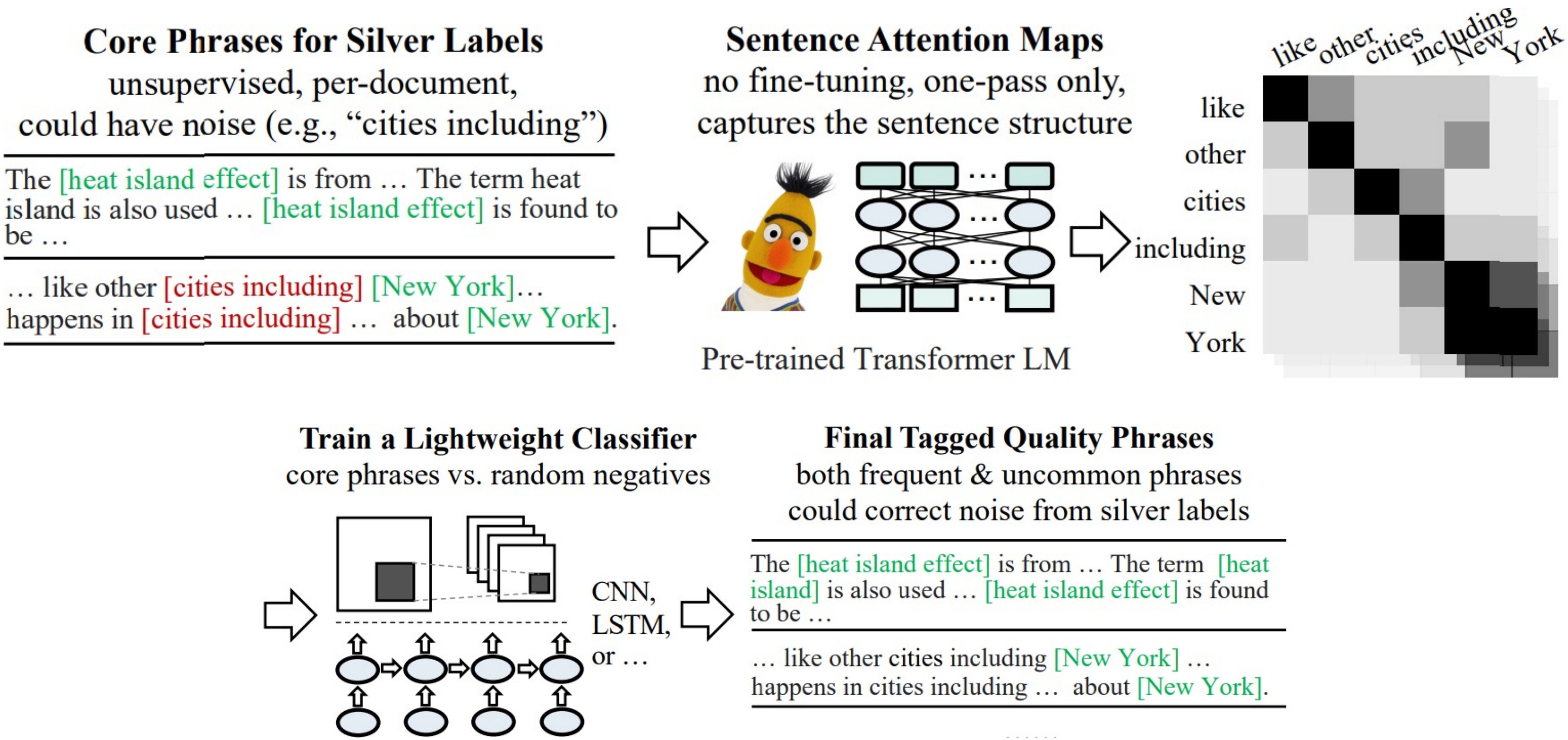
- Statistics-based models (*TopMine, SegPhrase, AutoPhrase)*
  - only work for frequent phrases, ignore valuable **infrequent / emerging phrases**
- Tagging-based models (*Spacy, StanfordNLP*)
  - do not have requirements for frequency
  - require **expensive and unscalable** sentence-level annotations for model training

Different Types of Supervisions

- Supervision
  - Human annotation
    - expensive, **hard to scale** to larger corpora and new domains
  - Distant supervision
    - tend to produce **incomplete labels** due to context-agnostic matching
      - e.g. "Heat [island effect] is found to be ..."
      - e.g. "Biomedical [data mining] is an important task where ..."
    - tend to match popular phrases, which form a small seen phrase vocabulary
      - easy for an embedding-based system to **memorize / overfit**

Framework of UCPhrase

- Silver Label Generation + Attention Map-based Span Prediction



Core Phrases for Silver Labels
unsupervised, per-document,
could have noise (e.g., "cities including")

The [heat island effect] is from … The term heat island is also used … [heat island effect] is found to be …

… like other [cities including] [New York]… happens in [cities including] … about [New York].

Sentence Attention Maps
no fine-tuning, one-pass only,
captures the sentence structure

Pre-trained Transformer LM

Train a Lightweight Classifier
core phrases vs. random negatives

Final Tagged Quality Phrases
both frequent & uncommon phrases
could correct noise from silver labels

The [heat island effect] is from … The term [heat island] is also used … [heat island effect] is found to be …

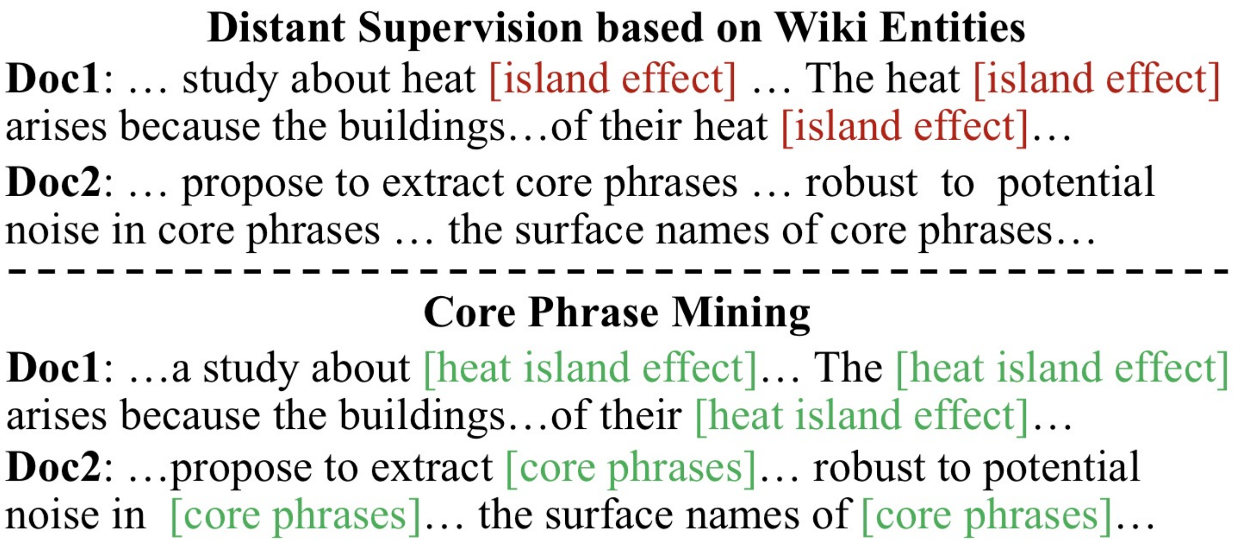
… like other cities including [New York] … happens in cities including … about [New York].

Silver Label Generation

- How do human readers accumulate new phrases?
  - we look for repeatedly used word sequences in a document, which are likely to be phrases by definition
  - e.g., *task name, method name, dataset name, concepts* in a publication
  - e.g., *human name, organization, locations* in a news article
  - even without any prior knowledge we can recognize these consistently used patterns from a document
- Mining core phrases as silver labels
  - independently mine **max word sequential patterns** within each document
  - filter out uninformative patterns (e.g. “of a”) with a stopword list
  - with each document as context
    - preserve contextual completeness (“biomedical data mining” vs. “data mining”)
    - avoid potential noises from propagating to the entire corpus

Silver Label Generation

- ❑ Compare core phrases with distant supervision
  - ❑ core phrases have advantages in both **quantity** and **quality**
  - ❑ core phrases preserve better **contextual completeness**
  - ❑ core phrase mining discover more **infrequent phrases** in the corpus
  - ❑ core phrase mining does not depend on any existing KB



Distant Supervision based on Wiki Entities

Doc1: … study about heat [island effect] … The heat [island effect] arises because the buildings…of their heat [island effect]…

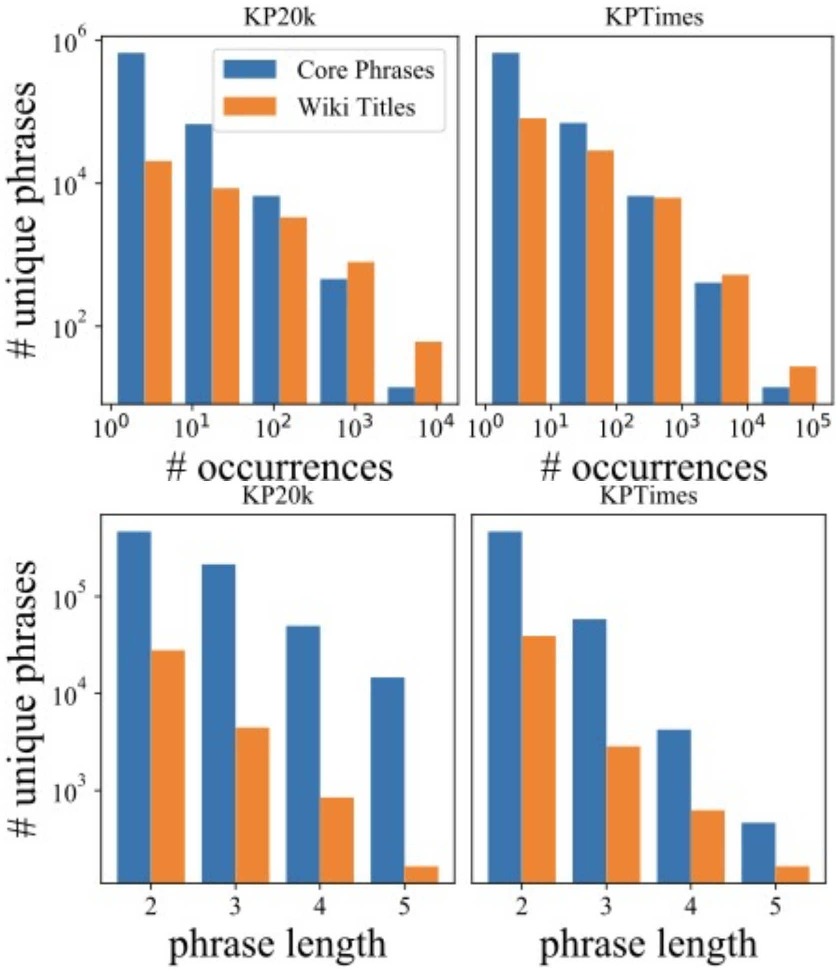
Doc2: … propose to extract core phrases … robust to potential noise in core phrases … the surface names of core phrases…

- -

Core Phrase Mining

Doc1: …a study about [heat island effect]… The [heat island effect] arises because the buildings…of their [heat island effect]…

Doc2: …propose to extract [core phrases]… robust to potential noise in [core phrases]… the surface names of [core phrases]…

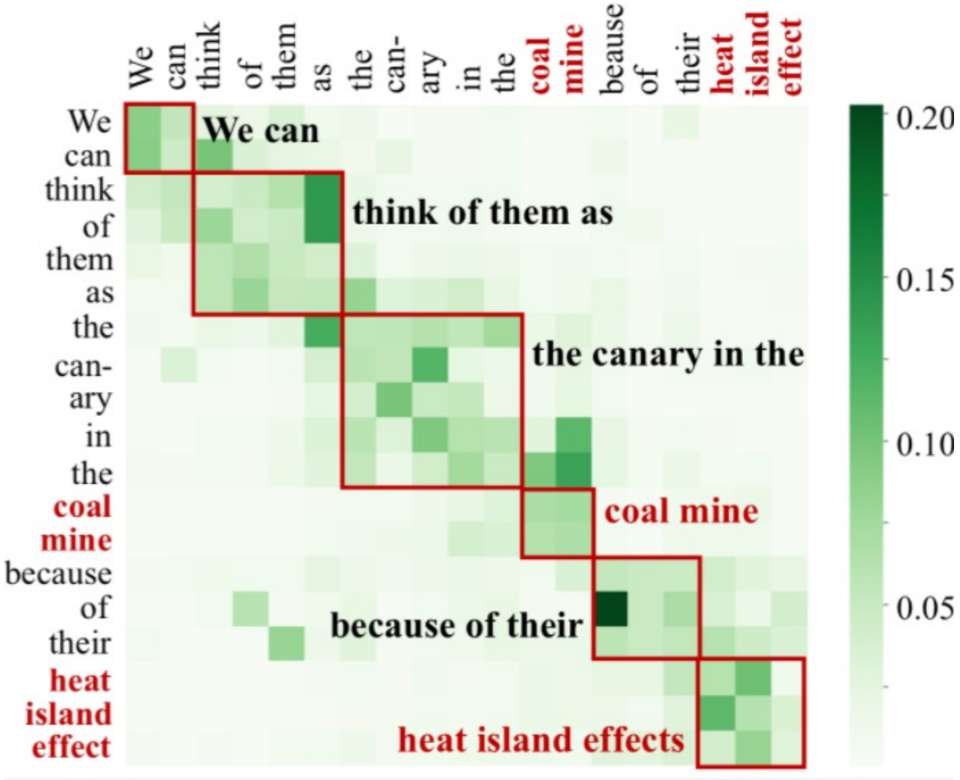


Surface-Agnostic Feature Generation

- What’s wrong with traditional embedding-based features?
  - embedding features are word identifiable -- it tells you which word you are looking at
  - easy to rigidly memorize all seen phrases / words in the training set
  - a dictionary matching model can easily achieve 0% training error, but cannot generalize to unseen phrases
- Good features for phrase recognition should be
  - agnostic to word **surface names** (so the model cannot rely on rigid memorization)
  - reveal the role that the span plays in the entire sentence (look at **sentence structure** rather than phrase names)

Attention Map

- Extract knowledge directly from a pre-trained language model
  - the **attention map** of a sentence vividly visualizes its **inner structure**
  - high quality phrases should have **distinct attention patterns** from ordinary spans



Phrase Tagging as Image Classification

- Given a sentence, treat all possible ngrams as candidates
- For each candidate of length K, extract its K*K attention map as feature
  - each attention head from each layer of a Transformer model will generate one attention map
  - for a RoBERTa base model, each candidate will have a (12*12 x K*K) = (144 x K*K) attention map
- Viewing the generated feature as a 144-channel image of size K*K
  - train a lightweight 2-layer CNN model for binary classification: is a phrase or not
  - why CNN: capture word interactions (attentions) from various ranges, also fast for training and inference
- Efficient implementation
  - only train the CNN module, without fine-tuning LM
  - only preserve attentions from the first 3 layers of LM (turns out to have similar performance with full attentions)

Quantitative Evaluation

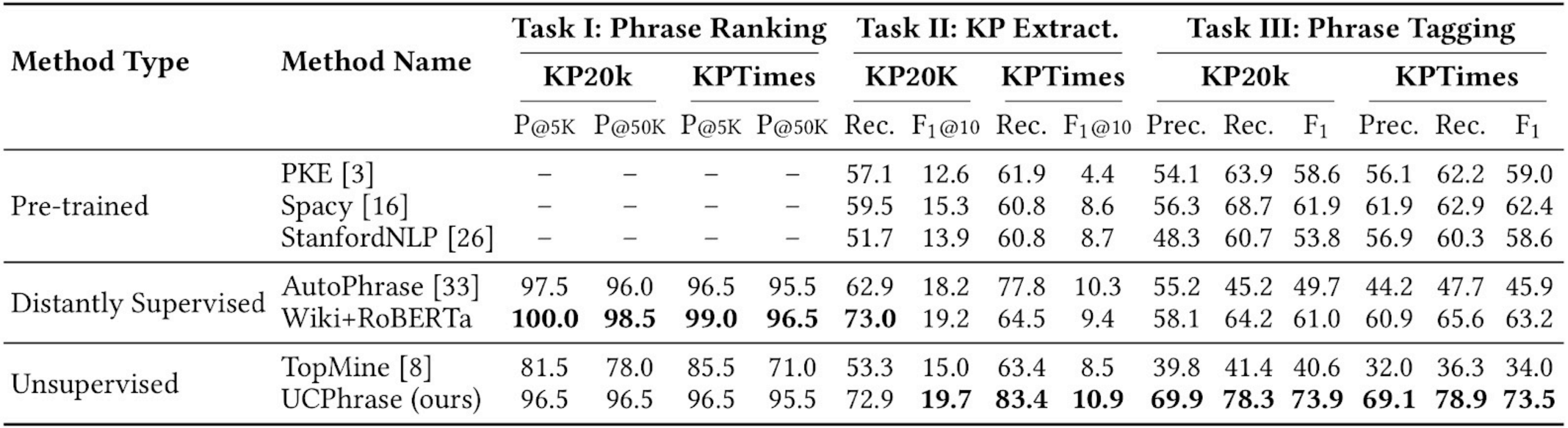
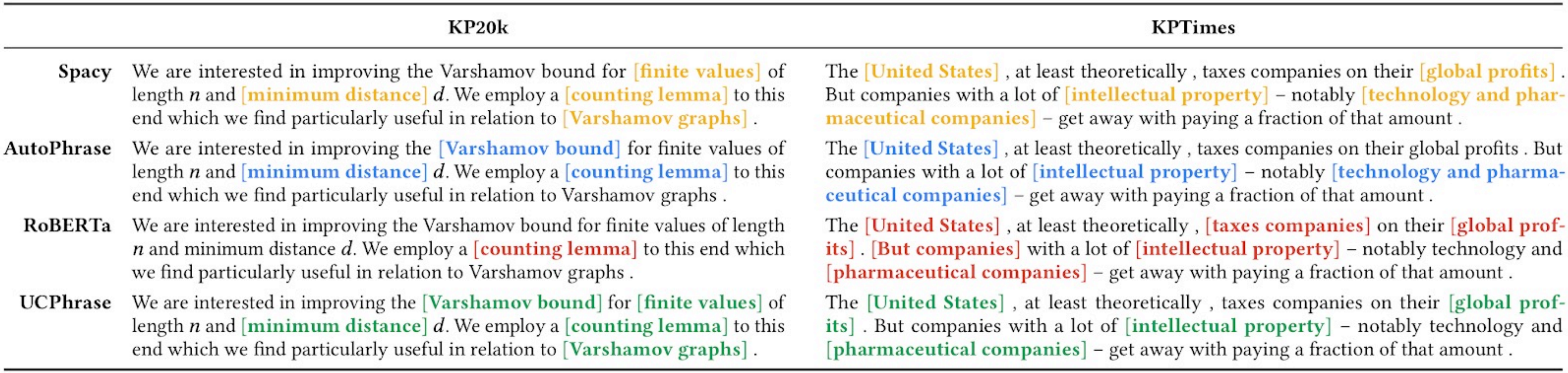


Table 2: Evaluation results (%) of three tasks for all compared methods on datasets on two domains.

Method Type	Method Name	Task I: Phrase Ranking				Task II: KP Extract.				Task III: Phrase Tagging					
		KP20k		KPTimes		KP20K		KPTimes		KP20k			KPTimes		
		P@5K	P@50K	P@5K	P@50K	Rec.	F_1@10	Rec.	F_1@10	Prec.	Rec.	F_1	Prec.	Rec.	F_1
Pre-trained	PKE [3]	–	–	–	–	57.1	12.6	61.9	4.4	54.1	63.9	58.6	56.1	62.2	59.0
	Spacy [16]	–	–	–	–	59.5	15.3	60.8	8.6	56.3	68.7	61.9	61.9	62.9	62.4
	StanfordNLP [26]	–	–	–	–	51.7	13.9	60.8	8.7	48.3	60.7	53.8	56.9	60.3	58.6
Distantly Supervised	AutoPhrase [33]	97.5	96.0	96.5	95.5	62.9	18.2	77.8	10.3	55.2	45.2	49.7	44.2	47.7	45.9
	Wiki+RoBERTa	**100.0**	**98.5**	**99.0**	**96.5**	**73.0**	19.2	64.5	9.4	58.1	64.2	61.0	60.9	65.6	63.2
Unsupervised	TopMine [8]	81.5	78.0	85.5	71.0	53.3	15.0	63.4	8.5	39.8	41.4	40.6	32.0	36.3	34.0
	UCPhrase (ours)	96.5	96.5	96.5	95.5	72.9	**19.7**	**83.4**	**10.9**	**69.9**	**78.3**	**73.9**	**69.1**	**78.9**	**73.5**

Case Study: comparing different methods

Table 6: Sentences tagged with different methods described in Section 4.3.



	KP20k	KPTimes
Spacy	We are interested in improving the Varshamov bound for [finite values] of length n and [minimum distance] d. We employ a [counting lemma] to this end which we find particularly useful in relation to [Varshamov graphs] .	The [United States] , at least theoretically , taxes companies on their [global profits] . But companies with a lot of [intellectual property] – notably [technology and pharmaceutical companies] – get away with paying a fraction of that amount .
AutoPhrase	We are interested in improving the [Varshamov bound] for finite values of length n and [minimum distance] d. We employ a [counting lemma] to this end which we find particularly useful in relation to Varshamov graphs .	The [United States] , at least theoretically , taxes companies on their global profits . But companies with a lot of [intellectual property] – notably [technology and pharmaceutical companies] – get away with paying a fraction of that amount .
RoBERTa	We are interested in improving the Varshamov bound for finite values of length n and minimum distance d. We employ a [counting lemma] to this end which we find particularly useful in relation to Varshamov graphs .	The [United States] , at least theoretically , [taxes companies] on their [global profits] . [But companies] with a lot of [intellectual property] – notably technology and [pharmaceutical companies] – get away with paying a fraction of that amount .
UCPhrase	We are interested in improving the [Varshamov bound] for [finite values] of length n and [minimum distance] d. We employ a [counting lemma] to this end which we find particularly useful in relation to [Varshamov graphs] .	The [United States] , at least theoretically , taxes companies on their [global profits] . But companies with a lot of [intellectual property] – notably technology and [pharmaceutical companies] – get away with paying a fraction of that amount .

Outline

- ❑ Phrase Mining
- ❑ Named Entity Recognition (NER)
  - ❑ Few-shot NER
  - ❑ Distantly-supervised NER
- ❑ Taxonomy Construction

Motivation

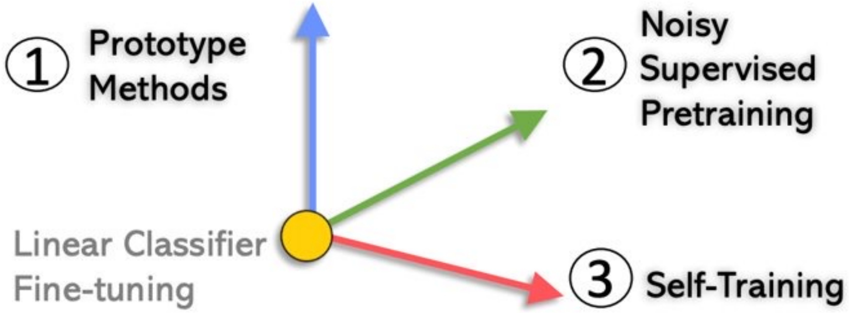
- Named entity recognition (NER) is a fundamental task in NLP with a wide spectrum of applications
  - question answering
  - knowledge base construction
  - dialog systems
  - …
- Deep neural models have achieved enormous success for NER
- However, a common bottleneck of training deep learning models is the acquisition of abundant high-quality human annotations (every entity in the sequence needs to be labeled!)

Few-shot NER

- Current NER models are trained for a series of fixed categories (e.g., PERSON, LOCATION, etc.) using large amounts of labeled data.
- Few-shot NER learns to transfer to new domains/categories with only a few training examples.

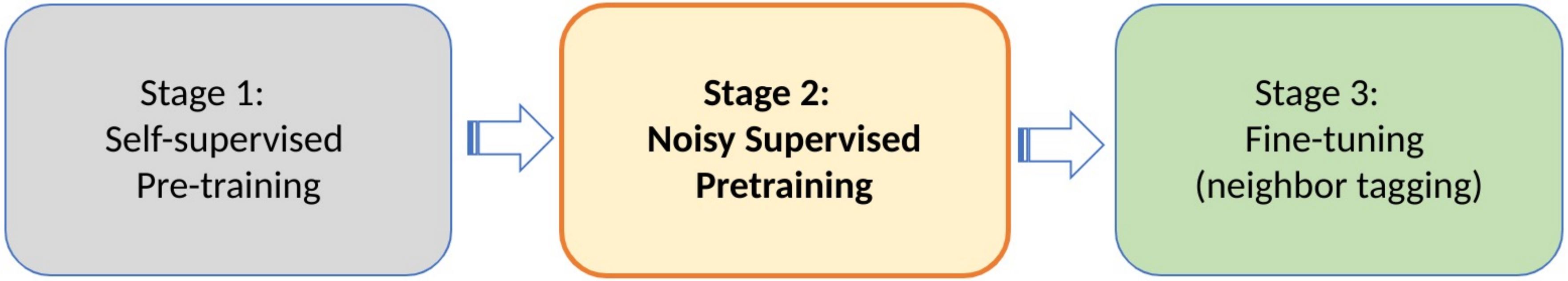
Our Empirical Study on Three Directions

- We explore three directions to improve the generalization ability of models in limited NER data settings.
- Prototype Methods (P) : A training objective typically used in few-shot learning setting to represent each class as a prototype
- Noisy Supervised Pretraining (NSP): Let the feature extractor model learn a discriminative NER space
- Self-Training (ST) : Leverage unlabeled data in target domain to improve the model



Noisy Supervised Pretraining

- ❑ Generic representations via self-supervised pre-trained language models are pre-trained with the task of randomly masked token prediction on massive corpora, and are agnostic to the downstream tasks.
- ❑ The goal of NER: Identifying named entities as emphasized tokens and assigning labels to them. –> Outweigh the representations of entities for NER.
- ❑ Noisy Supervised Pretraining (NSP): Let the feature extractor model learn a discriminative NER space



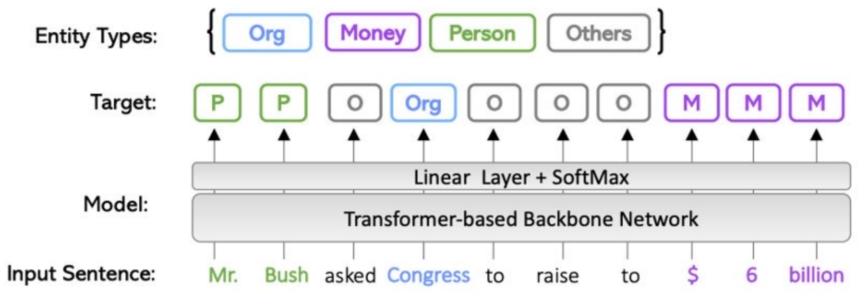
Stage 1:
Self-supervised
Pre-training

Stage 2:
Noisy Supervised
Pretraining

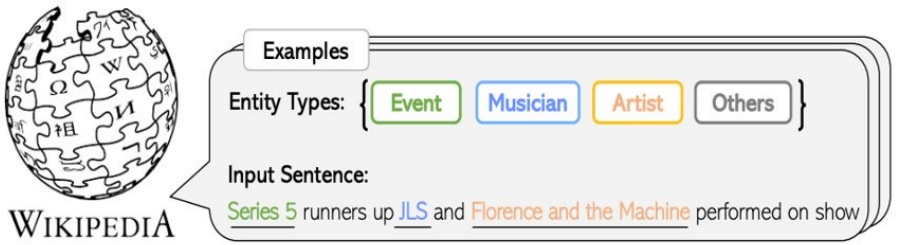
Stage 3:
Fine-tuning
(neighbor tagging)

Noisy Supervised Pretraining

- The WiFine[1] dataset: 113 entity types; over 50 million sentences.



(a) Baseline: NER with a linear classifier


WIKIPEDIA
Examples
Entity Types:
Event
Musician
Artist
Others
Input Sentence:
Series 5 runners up JLS and Florence and the Machine performed on show

(c) Noisy supervised pre-training

	Wikipedia (6.8GB)	CONLL-2003	OntoNER	…
Research Topic	NER	NER	NER	
# Entity Types	113	4	18	
# Entity Instances	70,000,000+	23,499	11,066	
# Training Sent.	52,000,000+	14,041	8,528	
# Training Token.	1,300,000,000+	203,621	147,724	

Target

[1] Transforming Wikipedia into a Large-Scale Fine-Grained Entity Type Corpus. Abbas Ghaddar, Philippe Langlais, 2018

Self-Training

- ❑ Learn teacher model θ_tea via cross-entropy loss with labeled tokens.
- ❑ Generate soft labels using a teacher model on unlabeled tokens.

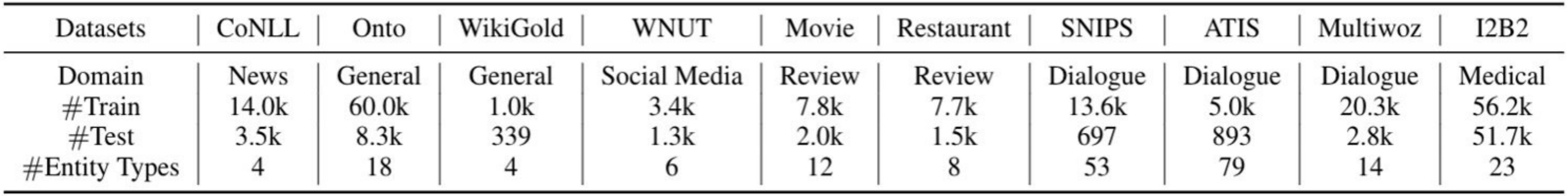
$$\tilde{\boldsymbol{y}}_i = f_{\boldsymbol{\theta}^{\text{tea}}}(\tilde{\boldsymbol{x}}_i), \forall \tilde{\boldsymbol{x}}_i \in \mathcal{D}^{\text{U}}$$

- ❑ Learn a student model θ_stu via cross entropy loss on both labeled and unlabeled tokens.

$$\begin{aligned}\mathcal{L}_{\text{ST}} = & \frac{1}{|\mathcal{D}^{\text{L}}|} \sum_{\boldsymbol{x}_i \in \mathcal{D}^{\text{L}}} \mathcal{L}(f_{\boldsymbol{\theta}^{\text{stu}}}(\boldsymbol{x}_i), \boldsymbol{y}_i) \\ & + \frac{\lambda_{\text{U}}}{|\mathcal{D}^{\text{U}}|} \sum_{\tilde{\boldsymbol{x}}_i \in \mathcal{D}^{\text{U}}} \mathcal{L}(f_{\boldsymbol{\theta}^{\text{stu}}}(\tilde{\boldsymbol{x}}_i), \tilde{\boldsymbol{y}}_i)\end{aligned}$$

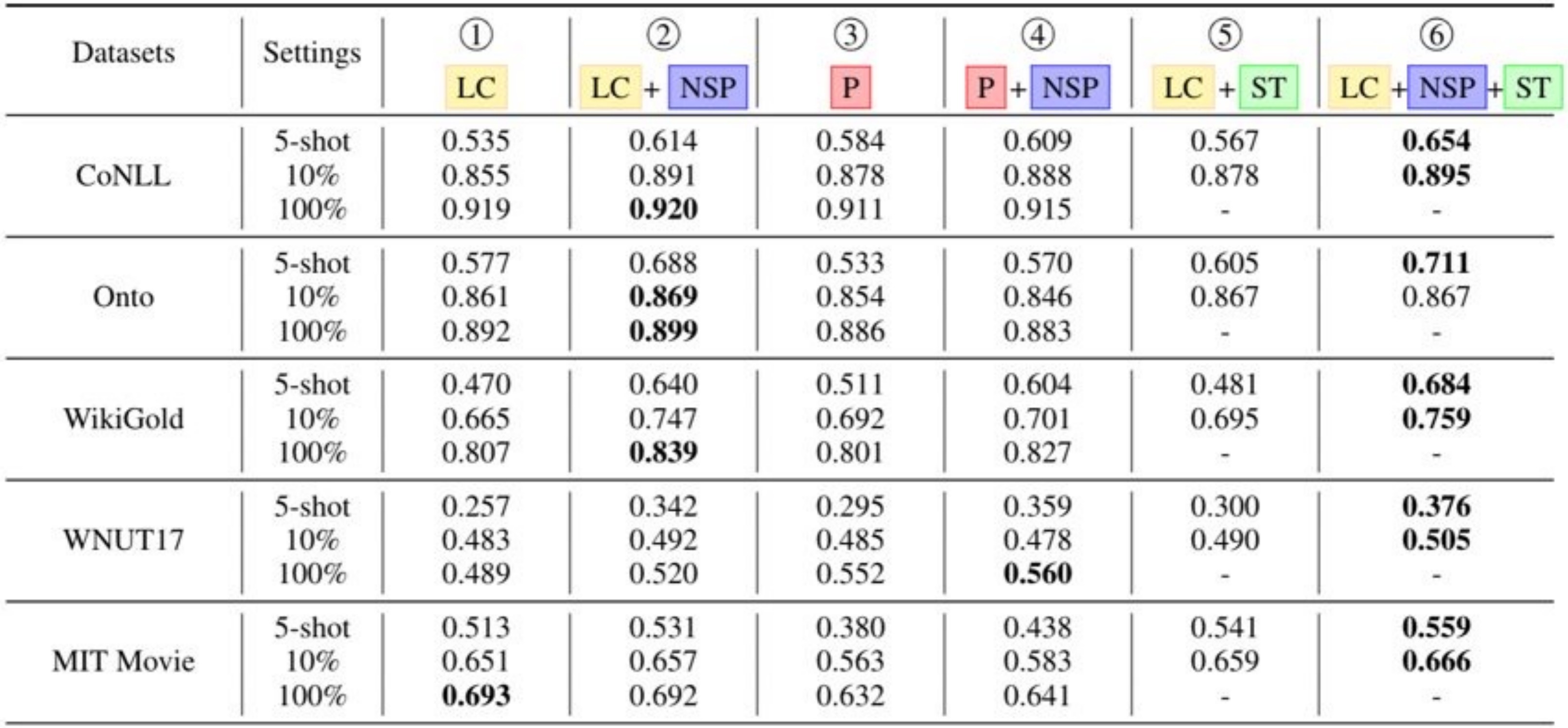
Experiments

- We collect 10 benchmark datasets for evaluating the model.
- The reason that we use multiple datasets across different domains is that they contain various entity types that could not be covered by the pretraining dataset.



Datasets	CoNLL	Onto	WikiGold	WNUT	Movie	Restaurant	SNIPS	ATIS	Multiwoz	I2B2
Domain	News	General	General	Social Media	Review	Review	Dialogue	Dialogue	Dialogue	Medical
#Train	14.0k	60.0k	1.0k	3.4k	7.8k	7.7k	13.6k	5.0k	20.3k	56.2k
#Test	3.5k	8.3k	339	1.3k	2.0k	1.5k	697	893	2.8k	51.7k
#Entity Types	4	18	4	6	12	8	53	79	14	23

Fine-tuning on Unseen Tasks



Datasets	Settings	① LC	② LC + NSP	③ P	④ P + NSP	⑤ LC + ST	⑥ LC + NSP + ST
CoNLL	5-shot	0.535	0.614	0.584	0.609	0.567	**0.654**
	10%	0.855	0.891	0.878	0.888	0.878	**0.895**
	100%	0.919	**0.920**	0.911	0.915	-	-
Onto	5-shot	0.577	0.688	0.533	0.570	0.605	**0.711**
	10%	0.861	**0.869**	0.854	0.846	0.867	0.867
	100%	0.892	**0.899**	0.886	0.883	-	-
WikiGold	5-shot	0.470	0.640	0.511	0.604	0.481	**0.684**
	10%	0.665	0.747	0.692	0.701	0.695	**0.759**
	100%	0.807	**0.839**	0.801	0.827	-	-
WNUT17	5-shot	0.257	0.342	0.295	0.359	0.300	**0.376**
	10%	0.483	0.492	0.485	0.478	0.490	**0.505**
	100%	0.489	0.520	0.552	**0.560**	-	-
MIT Movie	5-shot	0.513	0.531	0.380	0.438	0.541	**0.559**
	10%	0.651	0.657	0.563	0.583	0.659	**0.666**
	100%	**0.693**	0.692	0.632	0.641	-	-

Columns: Different Models

NF: Naïve Softmax Finetuning

NSP: Noisy Supervised Pretraining

P: Prototype-based Methods

ST: Self-Training

Rows: Different Tasks

5-shot: 5 example sentences for each entity type

10%: only use 10 percent of training data

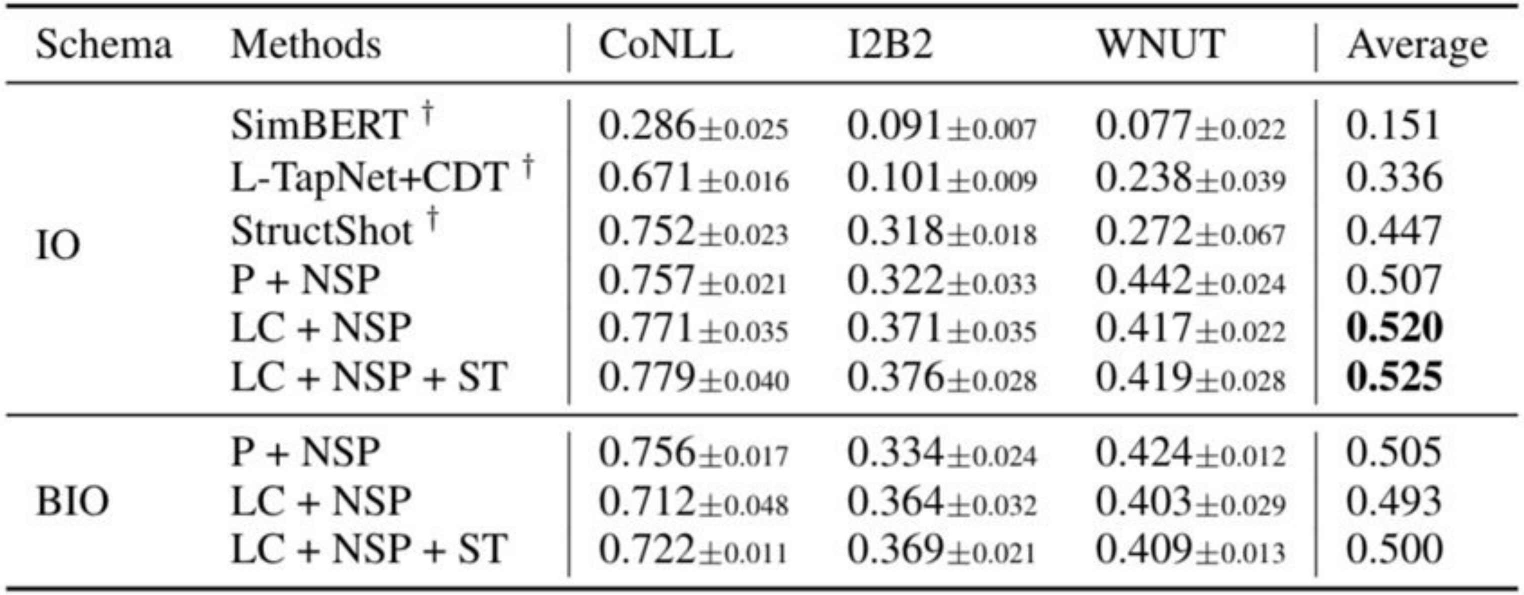
100%: use all training data

Observations: 1. Noisy supervised pretraining creates a better discriminative NER space, resulting in better results in most datasets.

2. Prototype-based methods can be better than naive softmax finetuning when the size of both labels and entity types are small.

3. Self-training methods that leverage unlabeled data constantly improve the results.

Fine-tuning on 5-shot NER Tasks: Comparison with SOTA



Schema	Methods	CoNLL	I2B2	WNUT	Average
IO	SimBERT †	0.286±0.025	0.091±0.007	0.077±0.022	0.151
	L-TapNet+CDT †	0.671±0.016	0.101±0.009	0.238±0.039	0.336
	StructShot †	0.752±0.023	0.318±0.018	0.272±0.067	0.447
	P + NSP	0.757±0.021	0.322±0.033	0.442±0.024	0.507
	LC + NSP	0.771±0.035	0.371±0.035	0.417±0.022	**0.520**
	LC + NSP + ST	0.779±0.040	0.376±0.028	0.419±0.028	**0.525**
BIO	P + NSP	0.756±0.017	0.334±0.024	0.424±0.012	0.505
	LC + NSP	0.712±0.048	0.364±0.032	0.403±0.029	0.493
	LC + NSP + ST	0.722±0.011	0.369±0.021	0.409±0.013	0.500

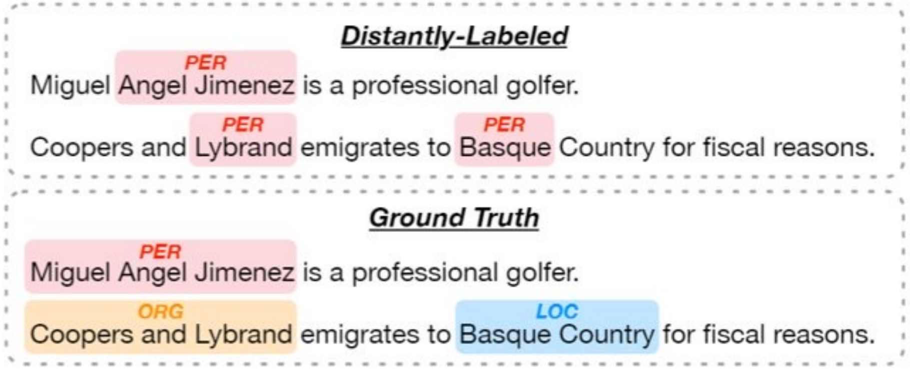
- ❑ Although IO schema is a defective schema, it can lead to higher performance. Results of both BIO and IO schemas are reported for fair comparison.
- ❑ We observe that our proposed methods consistently outperform the StructShot model across all three datasets.

Outline

- Phrase Mining
- Named Entity Recognition (NER)
    - Few-shot NER
    - Distantly-supervised NER
- Taxonomy Construction

Challenge

- ❑ The biggest challenge of distantly-supervised NER is that the distant supervision may induce **incomplete and noisy labels,** because
  - ❑ the distant supervision source has **limited coverage** of the entity mentions in the target corpus
  - ❑ some entities can be matched to multiple types in the knowledge bases---such **ambiguity** cannot be resolved by the context-free matching process
- ❑ Straightforward application of supervised learning will lead to deteriorated model performance, as neural models have the strong capacity to fit to the given (noisy) data



Distantly-Labeled

Miguel Angel Jimenez is a professional golfer.

Coopers and Lybrand emigrates to Basque Country for fiscal reasons.

Ground Truth

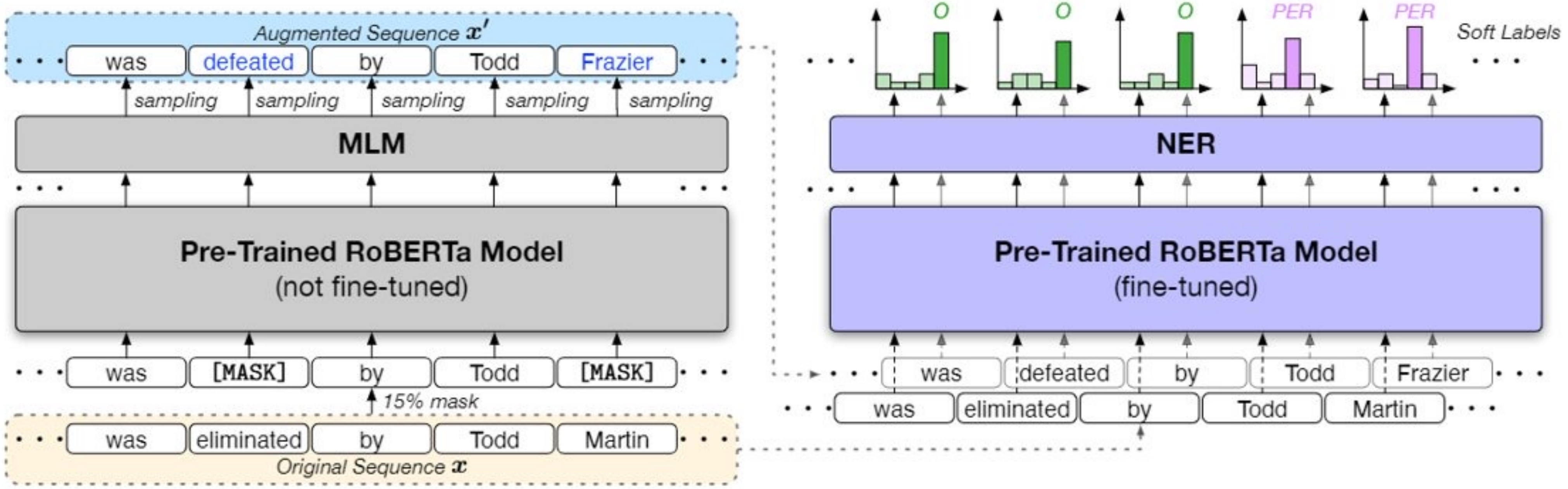
Miguel Angel Jimenez is a professional golfer.

Coopers and Lybrand emigrates to Basque Country for fiscal reasons.

Figure 1: Distant labels obtained with knowledge bases may be incomplete and noisy, resulting in wrongly-labeled tokens.

RoSTER

- RoSTER: Distantly-Supervised Named Entity Recognition with Noise-Robust Learning and Language Model Augmented Self-Training



Method

- Noise-Robust Learning: Why straightforward application of supervised NER learning on noisy data is bad?
- When the labels are noisy, training with the Cross Entropy (CE) loss can cause **overfitting** to the **wrongly-labeled** tokens
- Generalized Cross Entropy Loss (GCE)

$$\mathcal{L}_{\text{GCE}} = \sum_{i=1}^{n} w_i \frac{1 - f_{i,y_i}(\boldsymbol{x};\boldsymbol{\theta})^{1-q}}{1-q} \qquad w_i = \mathbb{1}\left(f_{i,y_i}(\boldsymbol{x};\boldsymbol{\theta}) > \tau\right)$$

Only use reliable labels
(model prediction agrees)

- Rationale: Since our loss function is noise-robust, the learned model will be dominated by the **correct majority** in the distant labels instead of quickly overfitting to label noise; if the model prediction disagrees with some given labels, they are potentially wrong

Method

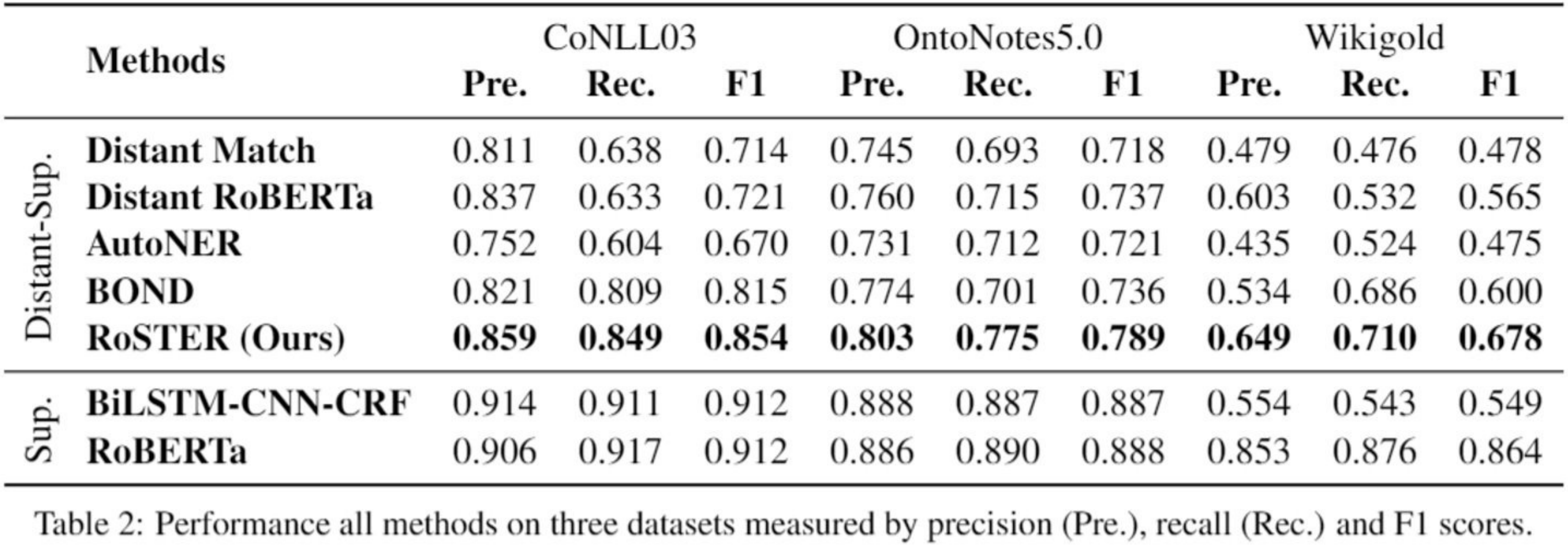
- Contextualized Augmentations with PLMs
- Randomly mask out 15% of tokens in the original sequence
- Feed the partially masked sequence into the pre-trained RoBERTa model
- Augmented sequence is created by sampling from the MLM output probability for each token
- Further enforce the label-preserving constraint:
  - sample only from the top-5 terms of MLM outputs
  - if the original token is capitalized or is a subword, so should the augmented one

Method

- Self-Training
- The goals of self-training (ST) are two-fold:
  - use the model's **high-confident predictions** that are likely to be reliable for guiding the model refinement on all tokens
  - encourage the model to **generate consistent predictions** on original sequences and augmented ones, based on the principle that a generalizable model should produce similar predictions for similar inputs
- Iteratively use the model's current predictions to derive **soft labels** and gradually update the model so that its predictions on both the original and the augmented sequences approximate the soft labels

Experiment Results

❑ Main Results



	Methods	CoNLL03			OntoNotes5.0			Wikigold		
		Pre.	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
Distant-Sup.	**Distant Match**	0.811	0.638	0.714	0.745	0.693	0.718	0.479	0.476	0.478
	Distant RoBERTa	0.837	0.633	0.721	0.760	0.715	0.737	0.603	0.532	0.565
	AutoNER	0.752	0.604	0.670	0.731	0.712	0.721	0.435	0.524	0.475
	BOND	0.821	0.809	0.815	0.774	0.701	0.736	0.534	0.686	0.600
	RoSTER (Ours)	**0.859**	**0.849**	**0.854**	**0.803**	**0.775**	**0.789**	**0.649**	**0.710**	**0.678**
Sup.	**BiLSTM-CNN-CRF**	0.914	0.911	0.912	0.888	0.887	0.887	0.554	0.543	0.549
	RoBERTa	0.906	0.917	0.912	0.886	0.890	0.888	0.853	0.876	0.864

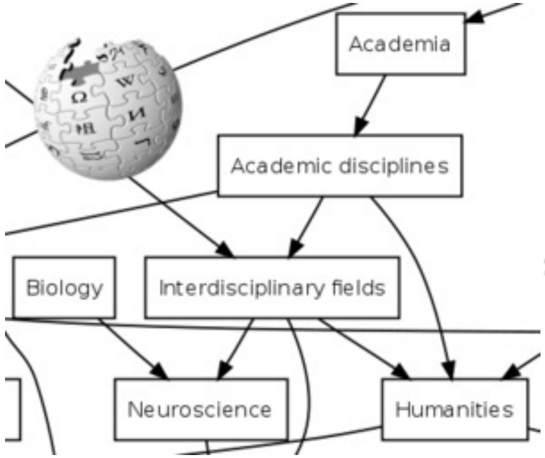
Table 2: Performance all methods on three datasets measured by precision (Pre.), recall (Rec.) and F1 scores.

Outline

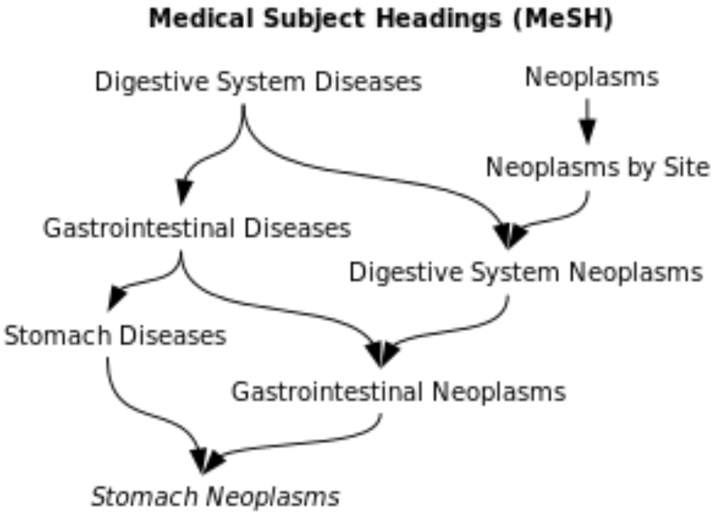
- Phrase Mining
- Named Entity Recognition
- Taxonomy Construction
  - Taxonomy Basics and Construction
  - Taxonomy Construction with Minimal User Guidance
  - Taxonomy Expansion

What is a Taxonomy?

- Taxonomy is a hierarchical organization of concepts
  - For example: Wikipedia category, ACM CCS Classification System, Medical Subject Heading (MeSH), Amazon Product Category, Yelp Category List, WordNet, and etc.

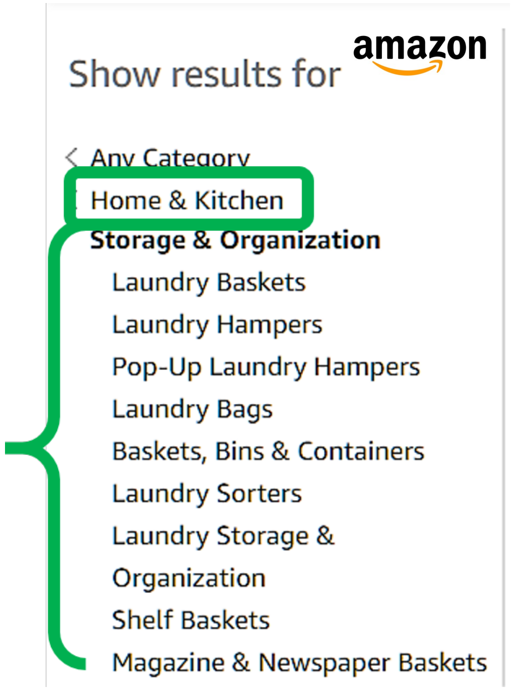


Wikipedia Category



Medical Subject Headings (MeSH)
Digestive System Diseases
Neoplasms
Neoplasms by Site
Gastrointestinal Diseases
Digestive System Neoplasms
Stomach Diseases
Gastrointestinal Neoplasms
Stomach Neoplasms

MeSH



Show results for amazon

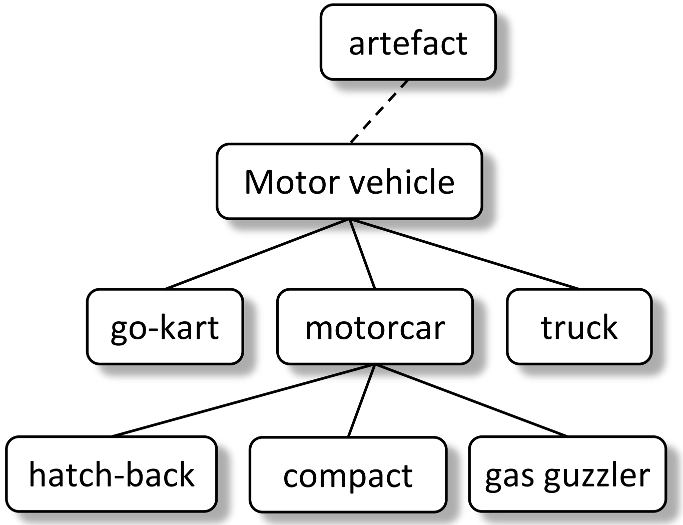
< Any Category

Home & Kitchen

Storage & Organization

- Laundry Baskets
- Laundry Hampers
- Pop-Up Laundry Hampers
- Laundry Bags
- Baskets, Bins & Containers
- Laundry Sorters
- Laundry Storage & Organization
- Shelf Baskets
- Magazine & Newspaper Baskets

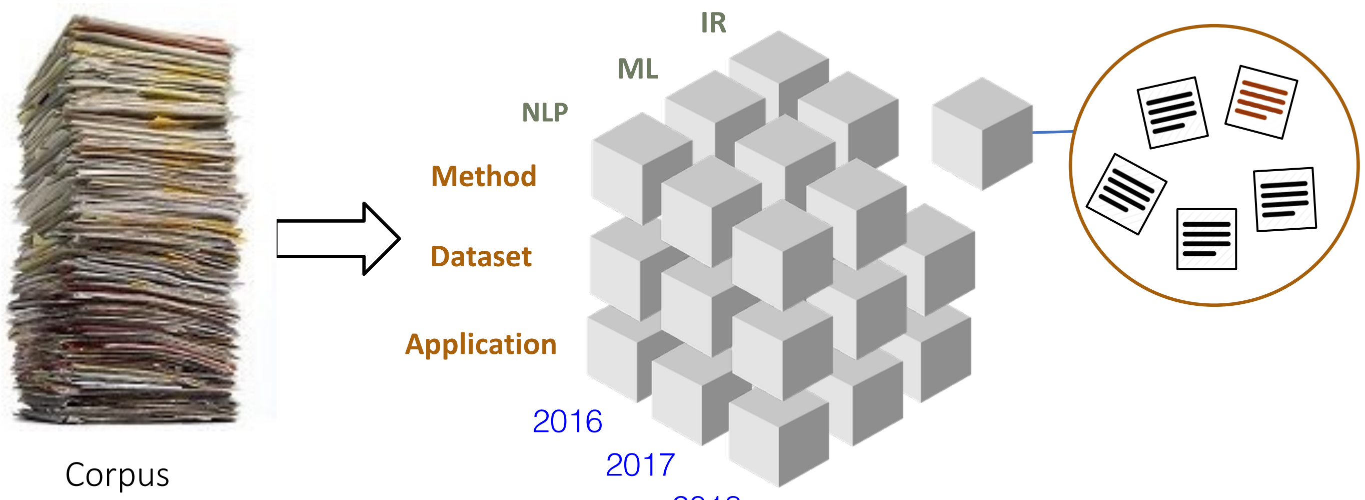
Amazon Product Category



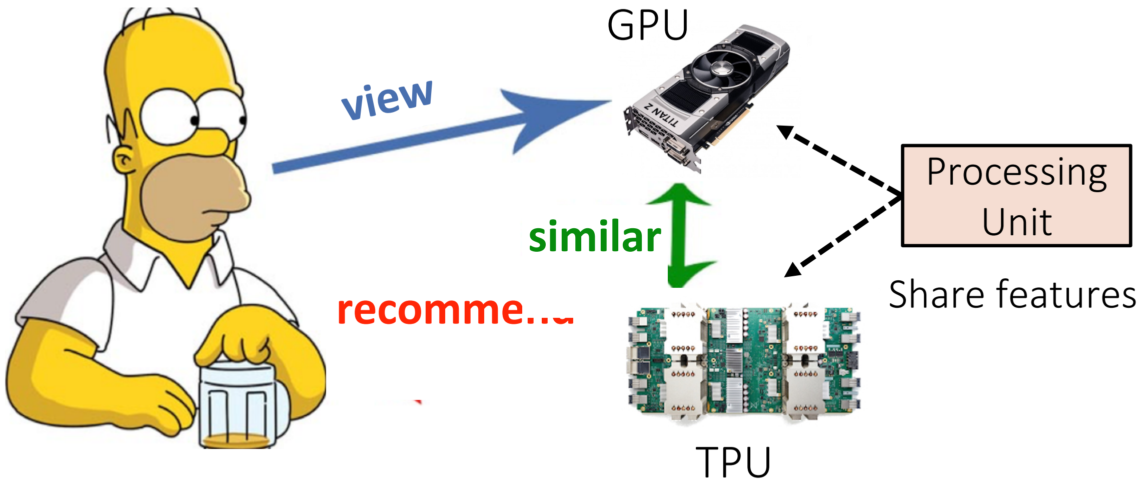
WordNet

Why do we need a Taxonomy?

- Taxonomy can benefit many knowledge-rich applications
  - Question Answering
  - Knowledge Organization
  - Document Categorization
  - Recommender System

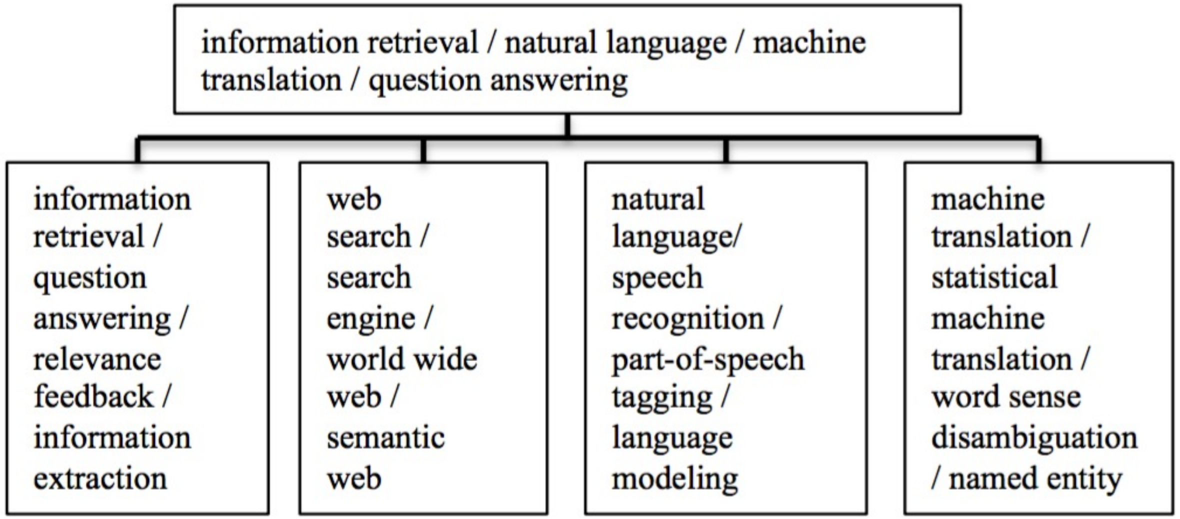


Multi-dimensional Corpus Index



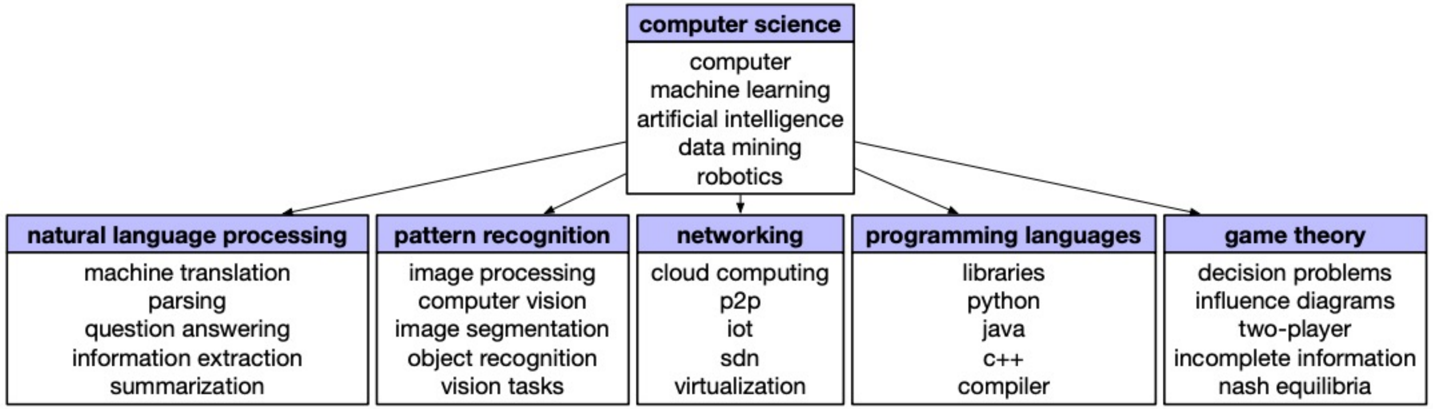
Clustering-based Taxonomy

- Compared to instance-based taxonomy (e.g., WordNet), clustering-based taxonomy has wider semantic coverage and facilitates clearer understanding of concepts.
- We focus on introducing clustering-based taxonomy construction in this tutorial.


information retrieval / natural language / machine translation / question answering
information retrieval / question answering / relevance feedback / information extraction
web search / search engine / world wide web / semantic web
natural language/ speech recognition / part-of-speech tagging / language modeling
machine translation / statistical machine translation / word sense disambiguation / named entity

Multi-faceted Taxonomy Construction

- Limitations of existing taxonomy:
  - A generic taxonomy with fixed “is-a” relation between nodes
  - Fail to adapt to users’ specific interest in special areas by dominating the hierarchical structure of irrelevant terms
- Multi-faceted Taxonomy
  - One facet only reflects a certain kind of relation between parent and child nodes in a user-interested field.



computer science
computer
machine learning
artificial intelligence
data mining
robotics

natural language processing
machine translation
parsing
question answering
information extraction
summarization

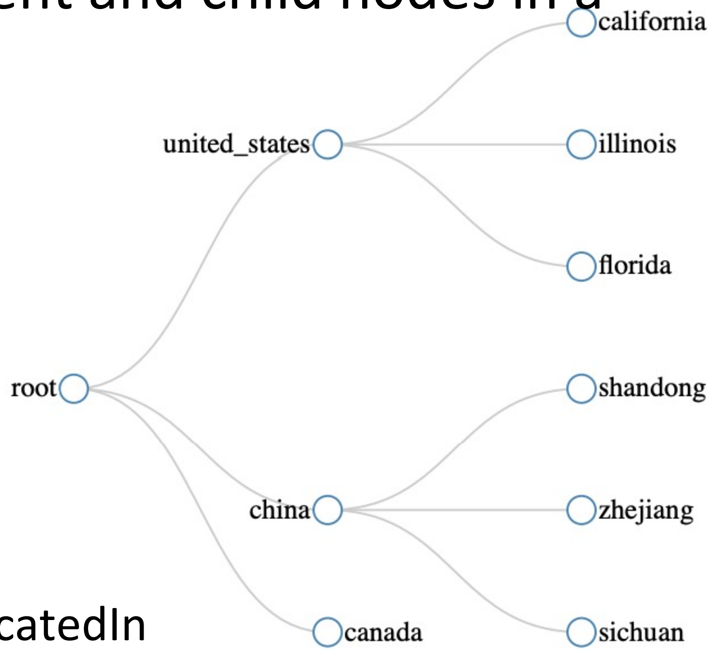
pattern recognition
image processing
computer vision
image segmentation
object recognition
vision tasks

networking
cloud computing
p2p
iot
sdn
virtualization

programming languages
libraries
python
java
c++
compiler

game theory
decision problems
influence diagrams
two-player
incomplete information
nash equilibria

Relation: IsSubfieldOf



Relation: IsLocatedIn

Two stages in constructing a complete taxonomy

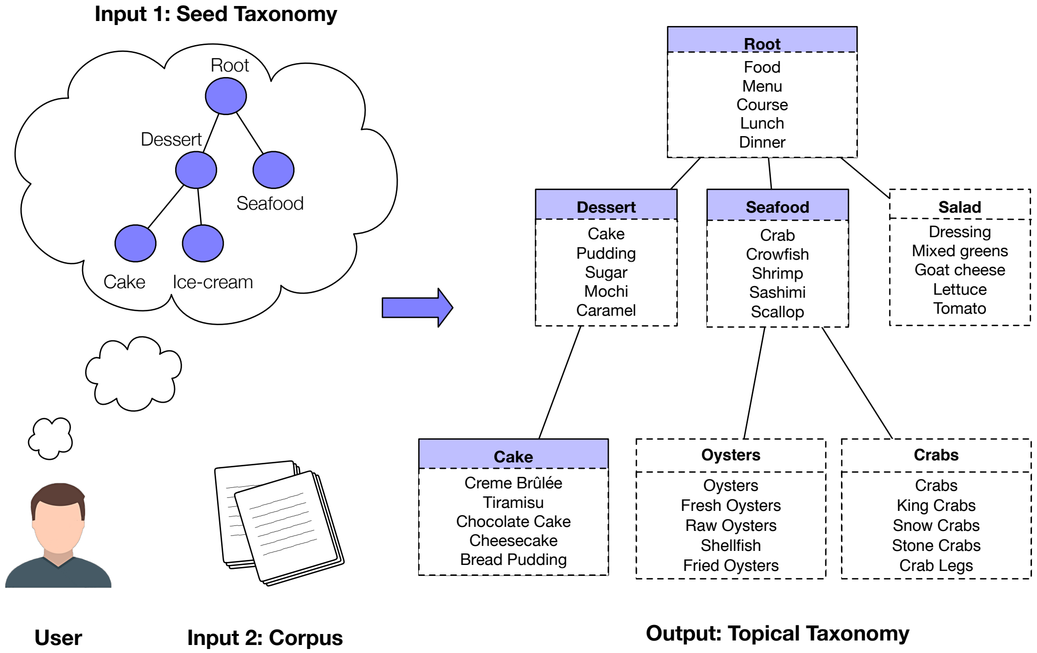
- ❑ Taxonomy Construction with Minimal User Guidance
  - ❑ Use a set of entities (possibly a seed taxonomy in a small scale) and unstructured text data to build a taxonomy organized by certain relations
- ❑ Taxonomy Expansion
  - ❑ Update an already constructed taxonomy by attaching new items to a suitable node on the existing taxonomy. This step is useful since reconstructing a new taxonomy from scratch can be resource-consuming.

Outline

- Phrase Mining
- Named Entity Recognition
- Taxonomy Construction
  - Taxonomy Basics and Construction
  - Taxonomy Construction with Minimal User Guidance
  - Taxonomy Expansion

Seed-Guided Topical Taxonomy Construction

- Previous clustering-based methods generate generic topical taxonomies which cannot satisfy user’s specific interest in certain areas and relations. Countless irrelevant terms and fixed “is-a” relations dominate the instance taxonomy.
- We study the problem of seed-guided topical taxonomy construction, where user gives a seed taxonomy as guidance, and a more complete topical taxonomy is generated from text corpus, with each node represented by a cluster of terms (topics).



Root
Food
Menu
Course
Lunch
Dinner

Dessert
Cake
Pudding
Sugar
Mochi
Caramel

Seafood
Crab
Crowfish
Shrimp
Sashimi
Scallop

Salad
Dressing
Mixed greens
Goat cheese
Lettuce
Tomato

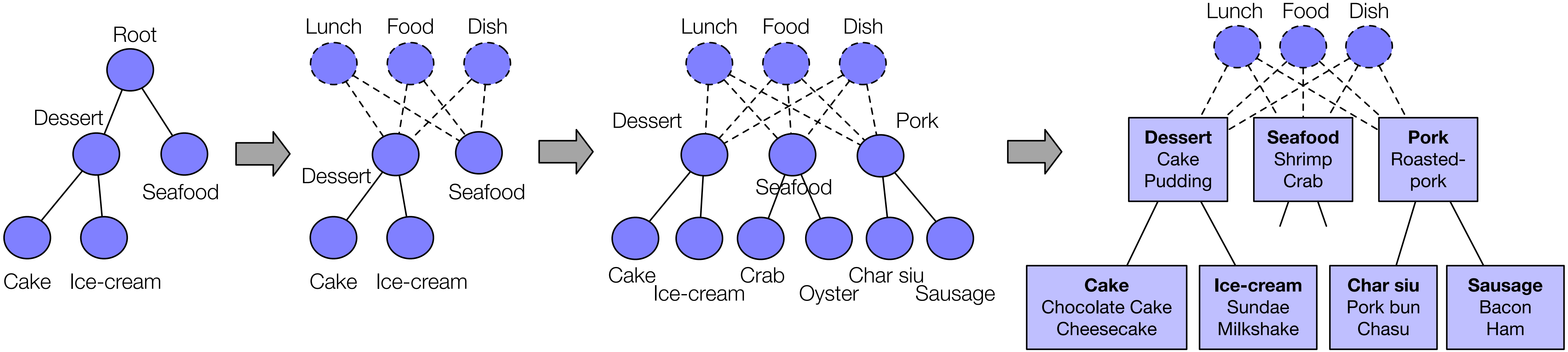
Cake
Creme Brûlée
Tiramisu
Chocolate Cake
Cheesecake
Bread Pudding

Oysters
Oysters
Fresh Oysters
Raw Oysters
Shellfish
Fried Oysters

Crabs
Crabs
King Crabs
Snow Crabs
Stone Crabs
Crab Legs

A user might want to learn about concepts in a certain aspect (e.g., *food* or *research areas*) from a corpus. He wants to know more about other kinds of food.

CoRel: Seed-Guided Topical Taxonomy Construction by Concept Learning and Relation Transferring [KDD'20]



Step 1: Relation transferring upwards

Step 2: Relation transferring downwards

Step 3: Concept learning for generating topical clusters

Step 1: Learn a relation classifier and transfer the relation upwards to **discover common root concepts** of existing topics.

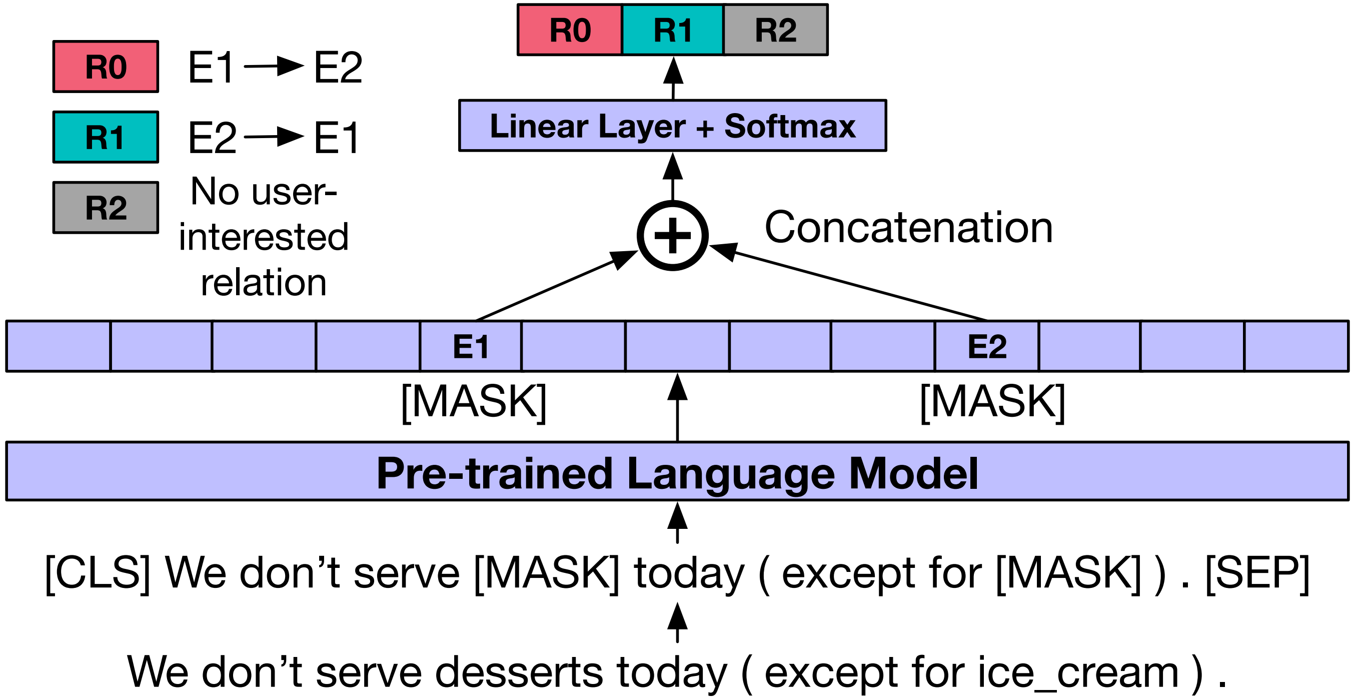
Step 2: Transfer the relation downwards to **find new topics/subtopics** as child nodes of root/topics.

Step 3: Learn a discriminative embedding space to **find distinctive terms for each concept** node in the taxonomy.

Relation Learning

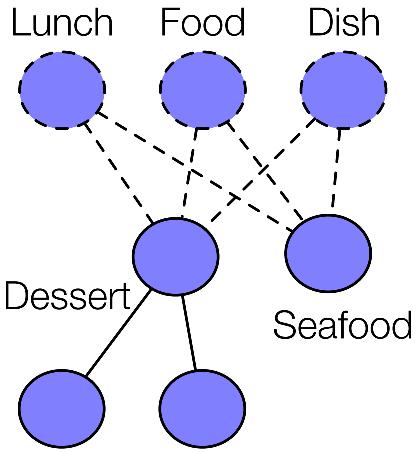
- We adopt a pre-trained deep language model to learn a relation classifier with only the user-given parent-child (<p,c>) pairs.

- **Training samples**: We generate relation statements from the corpus as training samples for this classifier. We assume that if a pair of <p,c> co-occurs in a sentence in the corpus, then that sentence implies their relation.


R0
E1 → E2
R1
E2 → E1
R2
No user-interested relation
R0
R1
R2
Linear Layer + Softmax
Concatenation
E1
E2
[MASK]
[MASK]
Pre-trained Language Model
[CLS] We don't serve [MASK] today (except for [MASK]) . [SEP]
We don't serve desserts today (except for ice_cream) .

Relation Transferring

- We first transfer the relation upwards to discover possible root nodes (e.g., “Lunch” and “Food”). This is because the root node would have more general contexts for us to find connections with potential new topics.

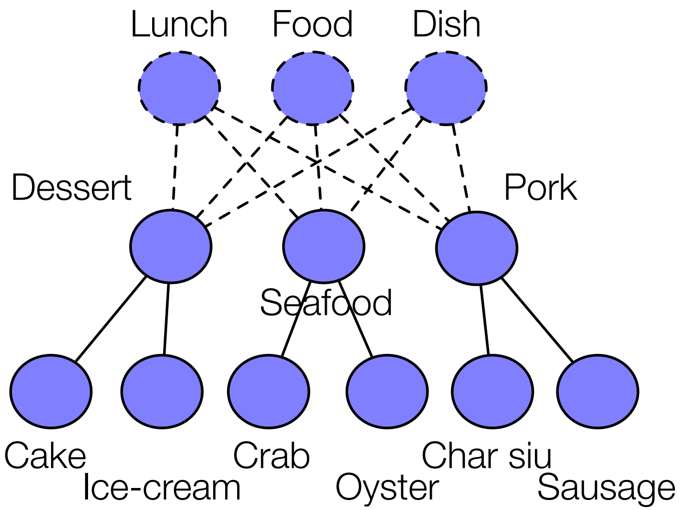


- We extract a list of parent nodes for each seed topic using the relation classifier. The common parent nodes shared by all user-given topics are treated as root nodes.
- To discover new topics (e.g, Pork), we transfer the relation downwards from these root nodes.

Relation Transferring

- We then transfer the relation downwards from each internal topic node to discover their subtopics.
- Since each candidate term has multiple mentions in the corpus, leading to multiple relation statements. We only count those confident predictions, and if the majority of these predictions judge the candidate term w as the child node of e, we retain the candidate term to be clustered later.

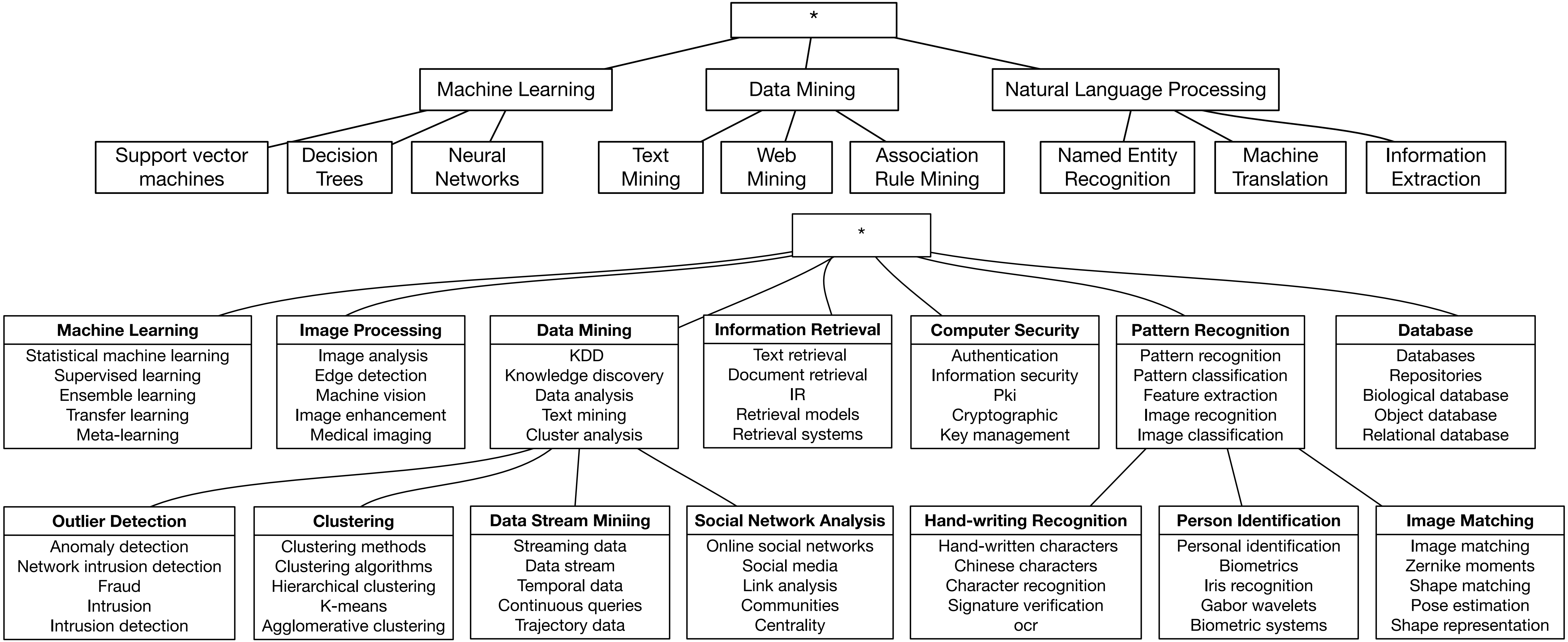
$$\text{Score}(\boldsymbol{e} \rightarrow \boldsymbol{w}) = \frac{\sum_{s_{e\rightarrow w}} \boxed{\mathbb{1}\left(KL\left(\boldsymbol{l} \| \boldsymbol{p}_w\right) > \delta\right)}}{\sum_{q \in Q} \sum_{s_q} \mathbb{1}\left(KL\left(\boldsymbol{l} \| \boldsymbol{p}_w\right) > \delta\right)}$$



Concept Learning

- Our concept learning module is used to learn a discriminative embedding space, so that each concept is surrounded by its representative terms. Within this embedding space, subtopic candidates are also clustered to form coherent subtopic nodes.
- Fine-grained concept names can be close in the embedding space, and directly using unsupervised word embedding might result in relevant but not distinctive terms (e.g., ``food" is relevant to both ``seafood" and ``dessert").
- Therefore, we leverage a **weakly-supervised text embedding framework** to discriminate these concepts in the embedding space, and this algorithm will be introduced in the next section.
- Subtopics should satisfy the following two constraints:
  - 1. must belong to representative words of that parent topic.
  - 2. must share parallel relations with given seed taxonomy.

Qualitative Results



*

Machine Learning

Support vector machines

Decision Trees

Neural Networks

Data Mining

Text Mining

Web Mining

Association Rule Mining

Natural Language Processing

Named Entity Recognition

Machine Translation

Information Extraction

*

Machine Learning
Statistical machine learning
Supervised learning
Ensemble learning
Transfer learning
Meta-learning

Image Processing
Image analysis
Edge detection
Machine vision
Image enhancement
Medical imaging

Data Mining
KDD
Knowledge discovery
Data analysis
Text mining
Cluster analysis

Information Retrieval
Text retrieval
Document retrieval
IR
Retrieval models
Retrieval systems

Computer Security
Authentication
Information security
Pki
Cryptographic
Key management

Pattern Recognition
Pattern recognition
Pattern classification
Feature extraction
Image recognition
Image classification

Database
Databases
Repositories
Biological database
Object database
Relational database

Outlier Detection
Anomaly detection
Network intrusion detection
Fraud
Intrusion
Intrusion detection

Clustering
Clustering methods
Clustering algorithms
Hierarchical clustering
K-means
Agglomerative clustering

Data Stream Miniing
Streaming data
Data stream
Temporal data
Continuous queries
Trajectory data

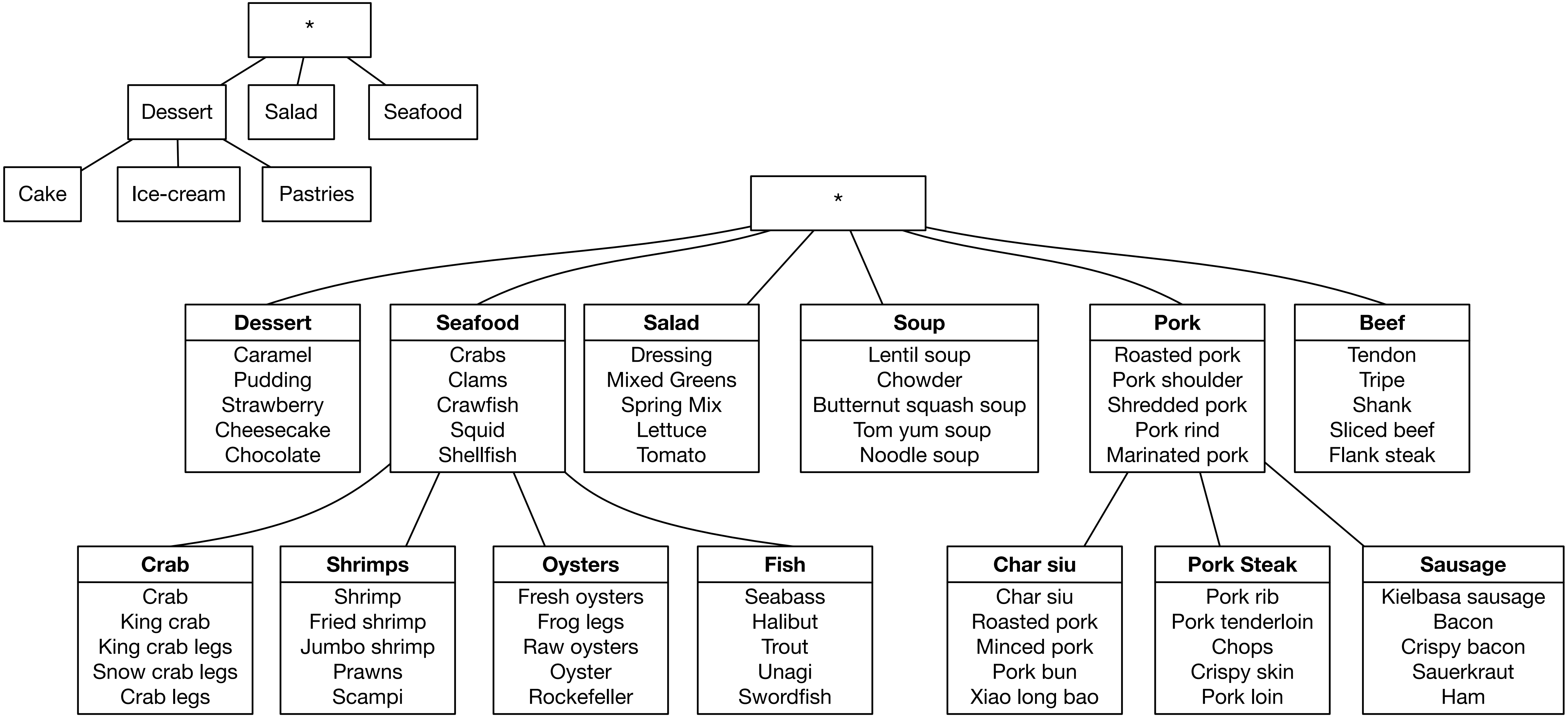
Social Network Analysis
Online social networks
Social media
Link analysis
Communities
Centrality

Hand-writing Recognition
Hand-written characters
Chinese characters
Character recognition
Signature verification
ocr

Person Identification
Personal identification
Biometrics
Iris recognition
Gabor wavelets
Biometric systems

Image Matching
Image matching
Zernike moments
Shape matching
Pose estimation
Shape representation

Qualitative Results



*

Dessert

Salad

Seafood

Cake

Ice-cream

Pastries

*

Dessert
Caramel
Pudding
Strawberry
Cheesecake
Chocolate

Seafood
Crabs
Clams
Crawfish
Squid
Shellfish

Salad
Dressing
Mixed Greens
Spring Mix
Lettuce
Tomato

Soup
Lentil soup
Chowder
Butternut squash soup
Tom yum soup
Noodle soup

Pork
Roasted pork
Pork shoulder
Shredded pork
Pork rind
Marinated pork

Beef
Tendon
Tripe
Shank
Sliced beef
Flank steak

Crab
Crab
King crab
King crab legs
Snow crab legs
Crab legs

Shrimps
Shrimp
Fried shrimp
Jumbo shrimp
Prawns
Scampi

Oysters
Fresh oysters
Frog legs
Raw oysters
Oyster
Rockefeller

Fish
Seabass
Halibut
Trout
Unagi
Swordfish

Char siu
Char siu
Roasted pork
Minced pork
Pork bun
Xiao long bao

Pork Steak
Pork rib
Pork tenderloin
Chops
Crispy skin
Pork loin

Sausage
Kielbasa sausage
Bacon
Crispy bacon
Sauerkraut
Ham

Outline

- Phrase Mining
- Named Entity Recognition
- Taxonomy Construction
  - Taxonomy Basics and Construction
  - Taxonomy Construction with Minimal User Guidance
  - Taxonomy Expansion

Taxonomy Enrichment: Motivation

- ❑ Why taxonomy enrichment instead of construction from scratch?
  - ❑ Already have a decent taxonomy built by experts and used in production
  - ❑ Most common terms are covered
  - ❑ New items (thus new terms) incoming everyday, cannot afford to rebuild the whole taxonomy frequently
  - ❑ Downstream applications require stable taxonomies to organize knowledge

Taxonomy Enrichment: Motivation

- Why taxonomy enrichment instead of construction from scratch?
  - Already have a decent taxonomy built by experts and used in production
  - Most common terms are covered
  - New items (thus new terms) incoming everyday, cannot afford to rebuild the whole taxonomy frequently
  - Downstream applications require stable taxonomies to organize knowledge
- What is missing then?
  - Emerging terms take time for humans to discover
  - Long-tail / fine-grained terms (leaf nodes) are likely to be neglected

Three Assumptions in Taxonomy Expansion

- First, we assume each concept will have a textual name
  - Therefore, we can get the *initial feature vector* of each concept in the existing taxonomy and of each new concept
- Second, we do not modify the existing taxonomy
  - Modification of existing relations happens less frequently and usually requires high cautiousness from human curators
- Third, we focus on finding parent node(s) of each new concept
  - New concept's parent node(s) typically appear in the existing taxonomy but its children node(s) may not exist the taxonomy

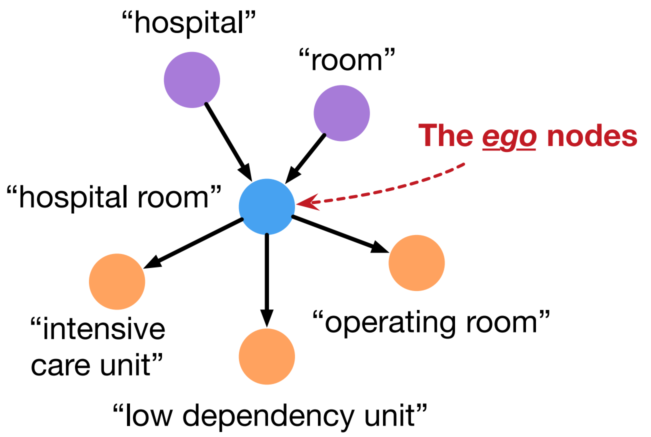
TaxoExpan: Self-supervised Taxonomy Expansion with Position-Enhanced Graph Neural Network [WWW' 20]

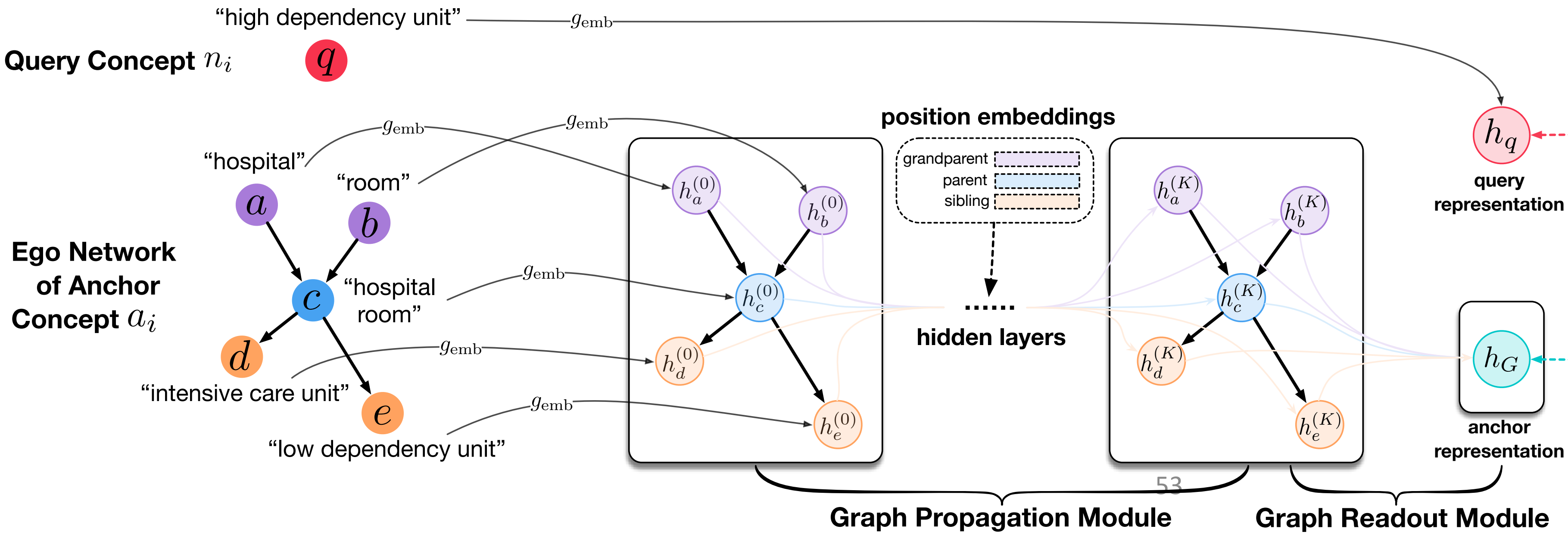
- ❑ **Two steps** in solving the problem:
  - ❑ Self-supervised term extraction
    - ❑ Automatically **extracts emerging terms** from a target domain
  - ❑ Self-supervised term attachment
    - ❑ A multi-class classification to match a new node to its potential parent
    - ❑ Heterogenous sources of information (structural, semantic, and lexical) can be used

Self-supervised Term Attachment

- **TaxoExpan** uses a matching score for each <*query*, *anchor*> pair to indicate how likely the *anchor concept* is the parent of *query concept*
- Key ideas:
  - Representing the *anchor concept* using its ego network (egonet)
  - Adding position information (relative to the *query concept*) into this egonet

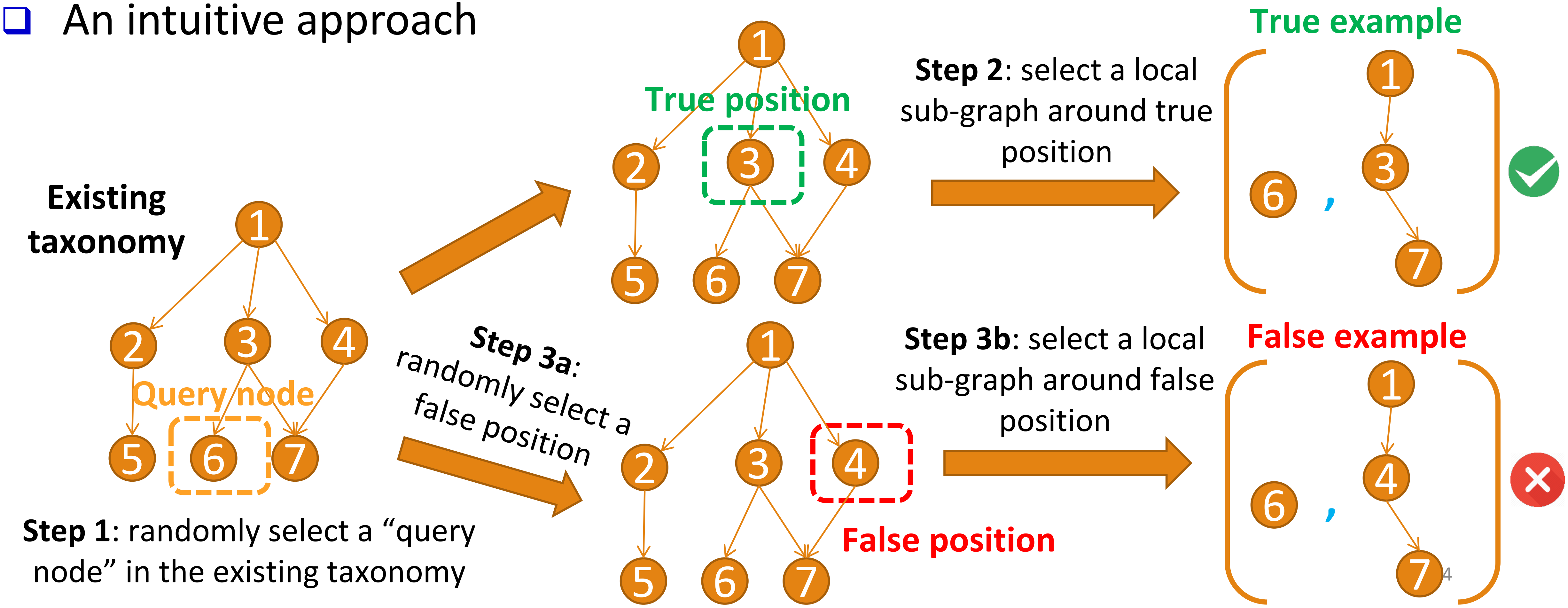
Query: "high dependency unit"





Leveraging Existing Taxonomy for Self-supervised Learning

- How to learn model parameters without relying on massive human-labeled data?
- An intuitive approach



Step 1: randomly select a "query node" in the existing taxonomy

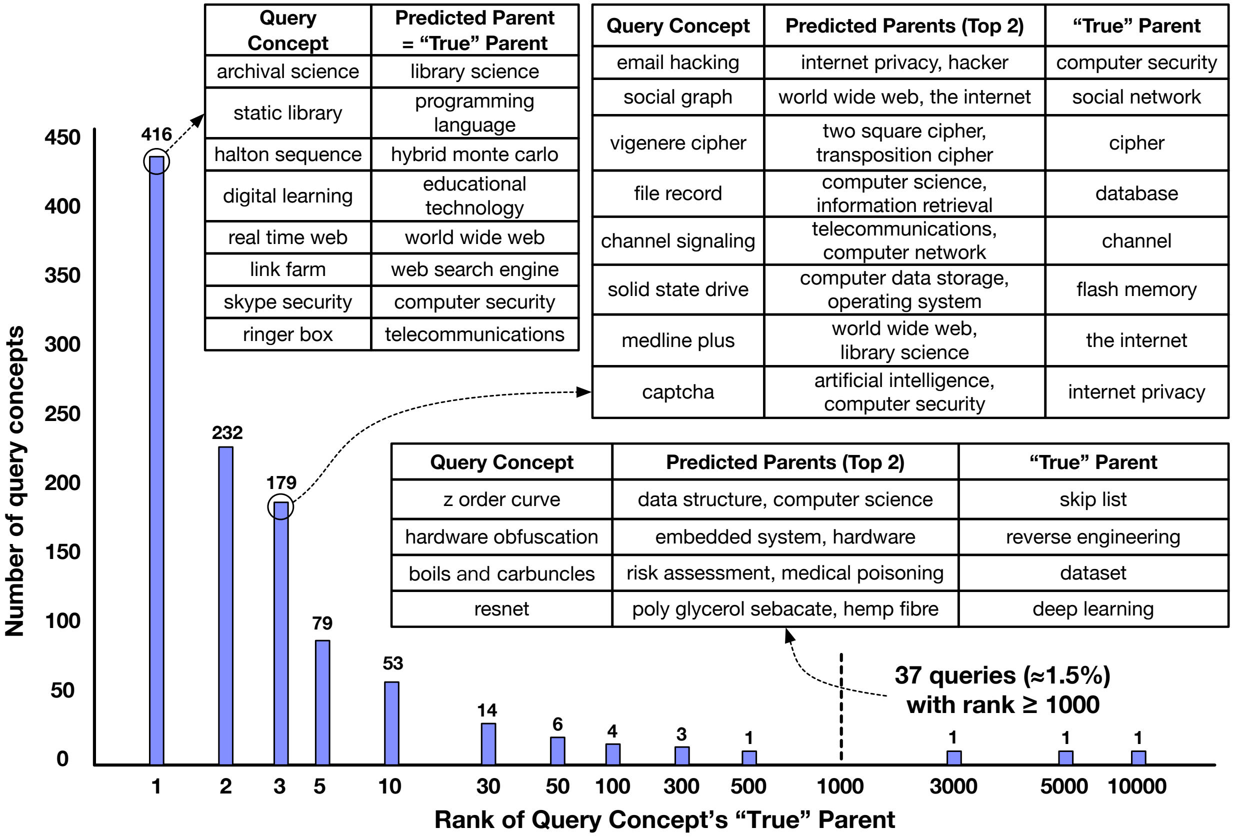
Step 2: select a local sub-graph around true position

Step 3a: randomly select a false position

Step 3b: select a local sub-graph around false position

TaxoExpan Framework Analysis

- Case studies on MAG-CS and MAG-Full datasets

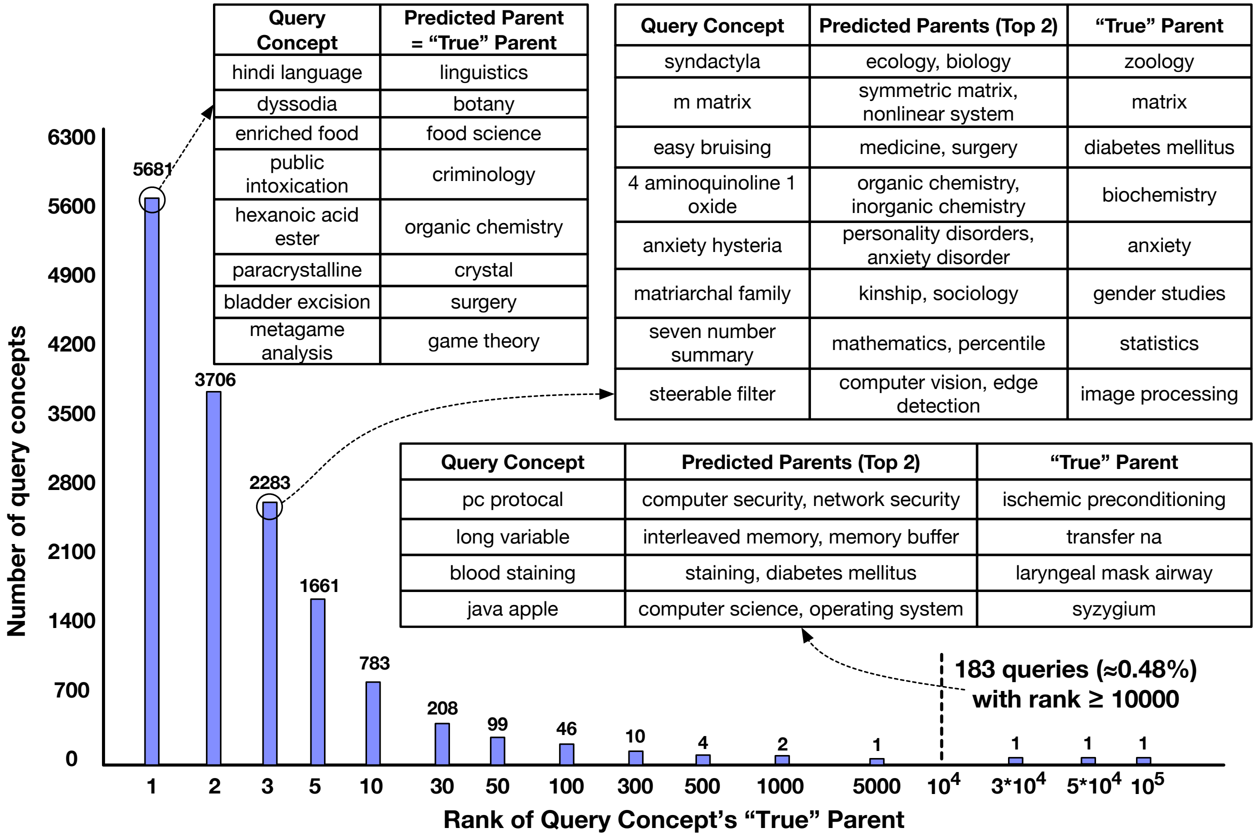


Query Concept	Predicted Parent = "True" Parent
archival science	library science
static library	programming language
halton sequence	hybrid monte carlo
digital learning	educational technology
real time web	world wide web
link farm	web search engine
skype security	computer security
ringer box	telecommunications

Query Concept	Predicted Parents (Top 2)	"True" Parent
email hacking	internet privacy, hacker	computer security
social graph	world wide web, the internet	social network
vigenere cipher	two square cipher, transposition cipher	cipher
file record	computer science, information retrieval	database
channel signaling	telecommunications, computer network	channel
solid state drive	computer data storage, operating system	flash memory
medline plus	world wide web, library science	the internet
captcha	artificial intelligence, computer security	internet privacy

Query Concept	Predicted Parents (Top 2)	"True" Parent
z order curve	data structure, computer science	skip list
hardware obfuscation	embedded system, hardware	reverse engineering
boils and carbuncles	risk assessment, medical poisoning	dataset
resnet	poly glycerol sebacate, hemp fibre	deep learning

(a) MAG-CS Dataset (totally 2450 query concepts)



Query Concept	Predicted Parent = "True" Parent
hindi language	linguistics
dyssodia	botany
enriched food	food science
public intoxication	criminology
hexanoic acid ester	organic chemistry
paracrystalline	crystal
bladder excision	surgery
metagame analysis	game theory

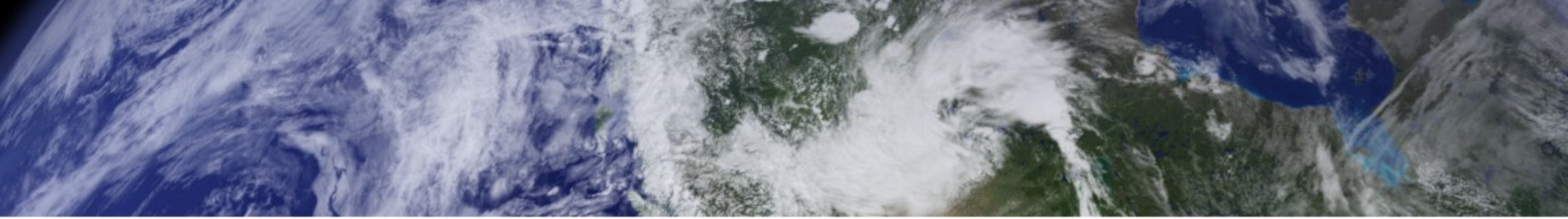
Query Concept	Predicted Parents (Top 2)	"True" Parent
syndactyla	ecology, biology	zoology
m matrix	symmetric matrix, nonlinear system	matrix
easy bruising	medicine, surgery	diabetes mellitus
4 aminoquinoline 1 oxide	organic chemistry, inorganic chemistry	biochemistry
anxiety hysteria	personality disorders, anxiety disorder	anxiety
matriarchal family	kinship, sociology	gender studies
seven number summary	mathematics, percentile	statistics
steerable filter	computer vision, edge detection	image processing

Query Concept	Predicted Parents (Top 2)	"True" Parent
pc protocal	computer security, network security	ischemic preconditioning
long variable	interleaved memory, memory buffer	transfer na
blood staining	staining, diabetes mellitus	laryngeal mask airway
java apple	computer science, operating system	syzygium

(b) MAG-Full Dataset (totally 37804 query concepts)

References

- Xiaotao Gu , Zihan Wang , Zhenyu Bi , Yu Meng, Liyuan Liu, Jiawei Han, Jingbo Shang. “UCPhrase: Unsupervised Context-aware Quality Phrase Tagging.” (KDD’21)
- Ahmed El-Kishky, Yanglei Song, Chi Wang, Clare R Voss, and Jiawei Han. 2014. Scalable Topical Phrase Mining from Text Corpora. Proceedings of the VLDB Endowment 8, 3 (2014).
- Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. IEEE Transactions on Knowledge and Data Engineering 30, 10 (2018), 1825–1837.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.
- Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. 55–60.
- Jiaxin Huang, Yiqing Xie, Yu Meng, Yunyi Zhang and Jiawei Han, “CoRel: Seed-Guided Topical Taxonomy Construction by Concept Learning and Relation Transferring”, KDD (2020)
- Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang and Jiawei Han ”TaxoExpan: Self-supervised Taxonomy Expansion with Position-Enhanced Graph Neural Network”, (WWW’20)



Q&A