KDD2020

FreeDOM: A Transferable Neural Architecture for Structured Information Extraction on Web Documents

Bill Yuchen Lin*, Ying Sheng, Nguyen Vo, Sandeep Tata

USC University of Southern California

Google AI

* The work was done while BYL was a research intern at Google.

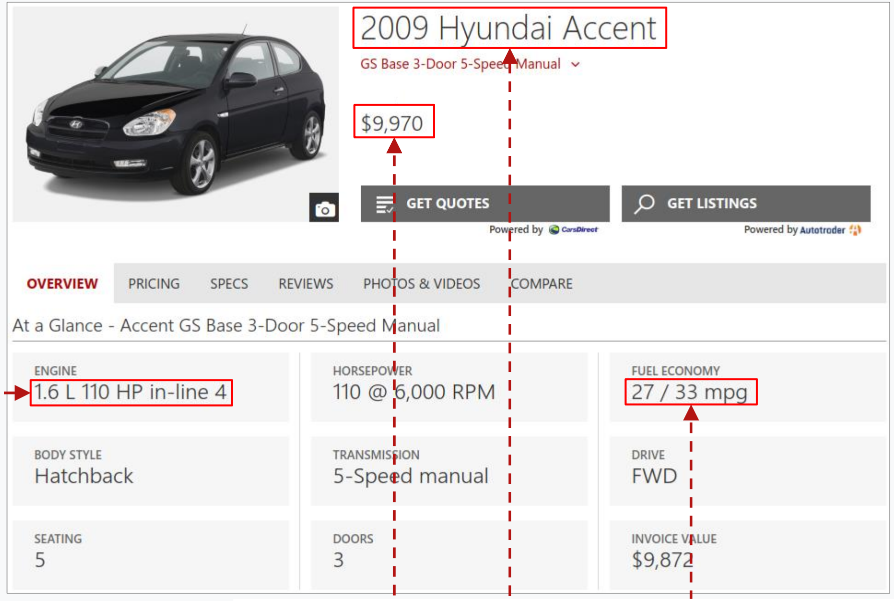
Task:
Information Extraction on Web Documents

Given a **domain** and a set of data **fields**.

- **Input**: Web pages
- **Output**: Structured data records

Downstream applications:

- Knowledge Graph Construction
- Question Answering
- Recommendation System
- etc.



Model	MSRP	Engine	Fuel Eco.
2009 H..	$9,970	1.6 L …	27/33 mpg …

Domain: **Auto**

Interested Fields:

- Model
- MSRP
- Engine
- Fuel Economy

Google Research

Key assumption: pages within a site have similar layout.

I have only A FEW websites of interest.

→ Develop and maintain rule-based matching programs (i.e. wrappers)!

→ Label some web pages, and train site-specific models via supervised learning (i.e., wrapper induction).

What if I have A LOT of unlabeled websites to process?

→ Building/training site-specific wrappers is **time-consuming and expensive!**

→ **RQ: Can we learn a transferrable IE model?**

Google Research

Problem formulation

A motivating example: building a course KB.

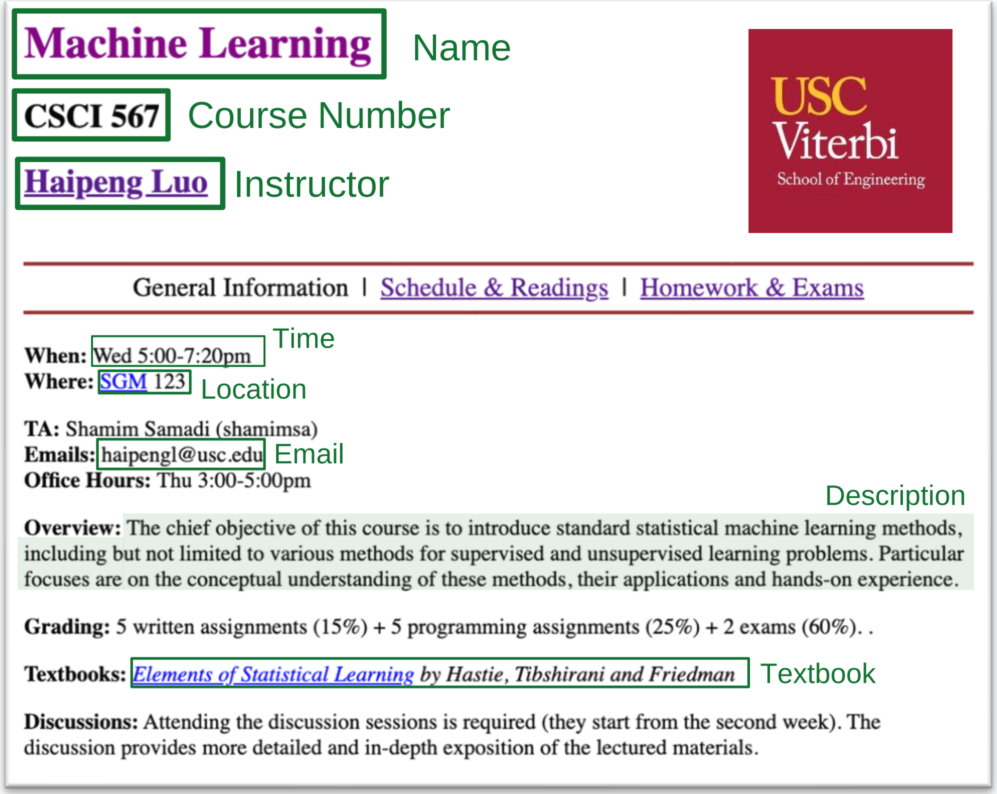
Domain: course
Fields: Name, Course Number, Instructor, Time, Location, Email, Textbook, Description

A few labeled **seed websites**.

MIT Harvard USC

.....

Detail Pages



A particular detail page w/ labels

Google Research

Problem formulation

A running example: building a course info KB.

Learning to **Generalize** for Unseen Websites

Many **unseen** websites, w/ different layouts

Univ. 1 Univ. 2 Univ. 3 Univ. 4

Univ. 5 Univ. 6 Univ. 7 .....

Test

A few labeled **seed websites.**

MIT Harvard USC

Training

A Transferable Model

Post-processing

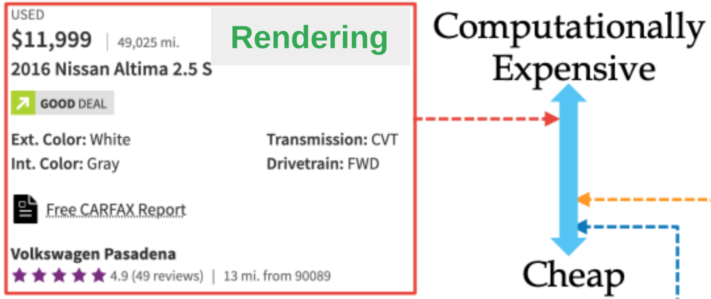
A Course Information Knowledge Base

Google Research

Problem formulation

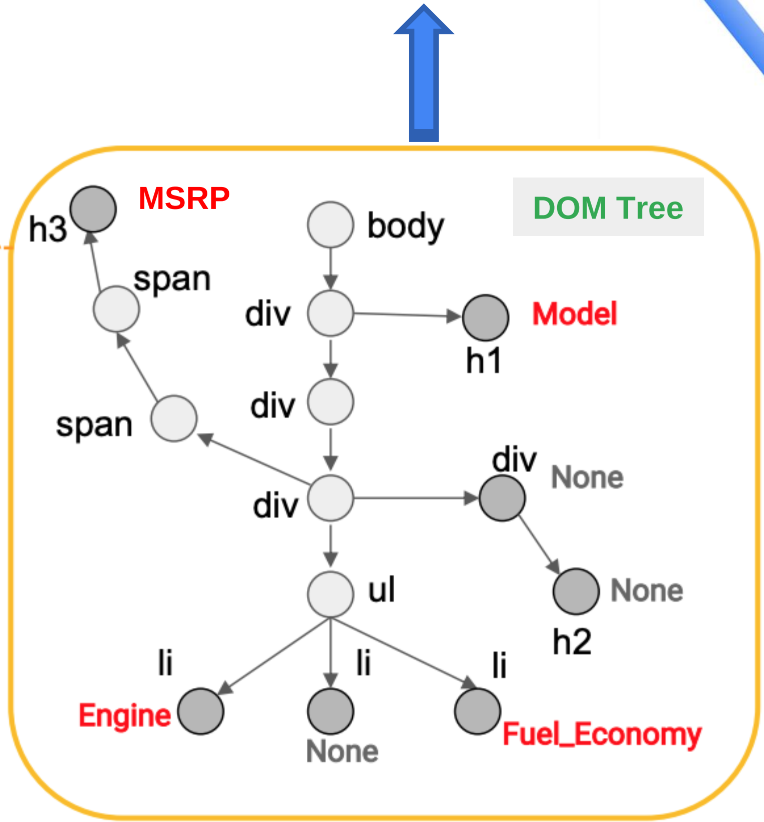
How to represent a web page

Information Extraction as DOM node classification



HTML Code

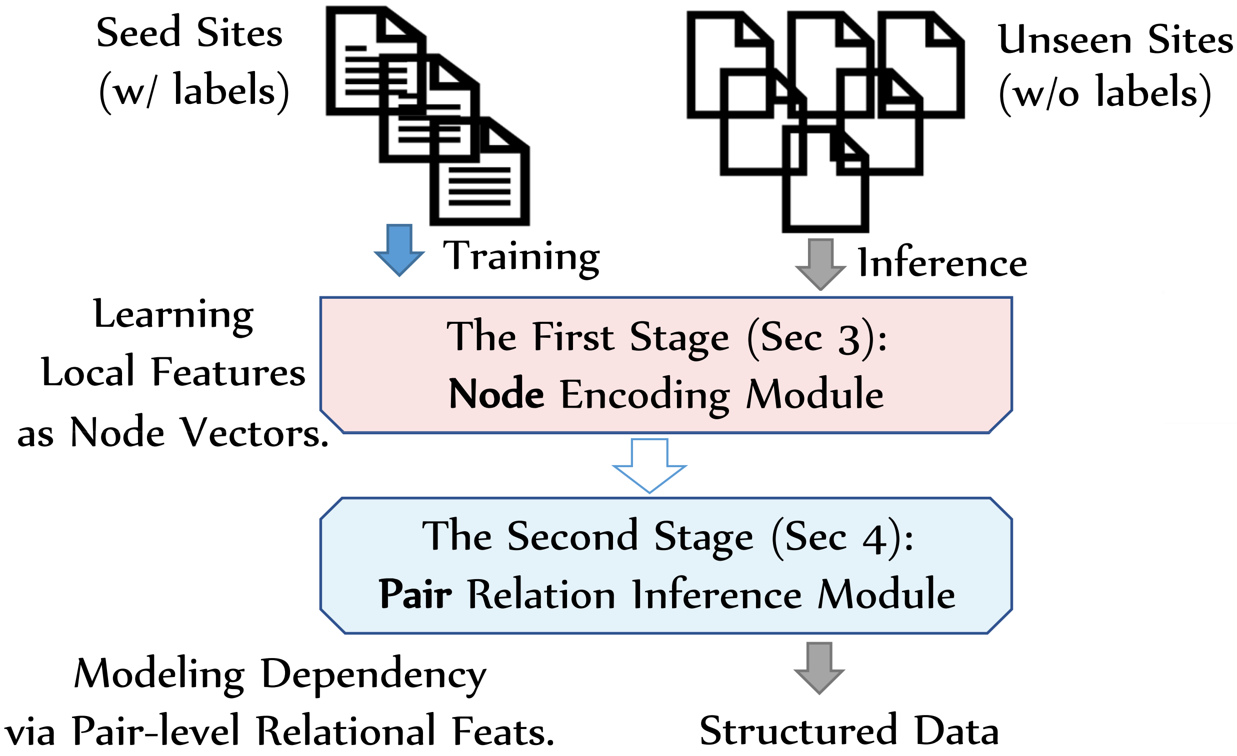
```
<div class="mod" id="yat-trim-engine">
    <div class="hd">
      <h2>Engine</h2>
    </div>
    <div class="bd">
      <div class="col col-left">
        <ul>
          <li>1.6L I4, 16 valves, 110 hp @ 6000 rpm</li>
          <li>5 speed manual transmission</li>
          <li>27 mpg city / 36 mpg hwy</li>
        </ul>
      </div>
      <div id="yat-green-rating" class="col">
  .......
```



Node labels:
- Model
- MSRP
- Engine
- Fuel_Eco.
- None

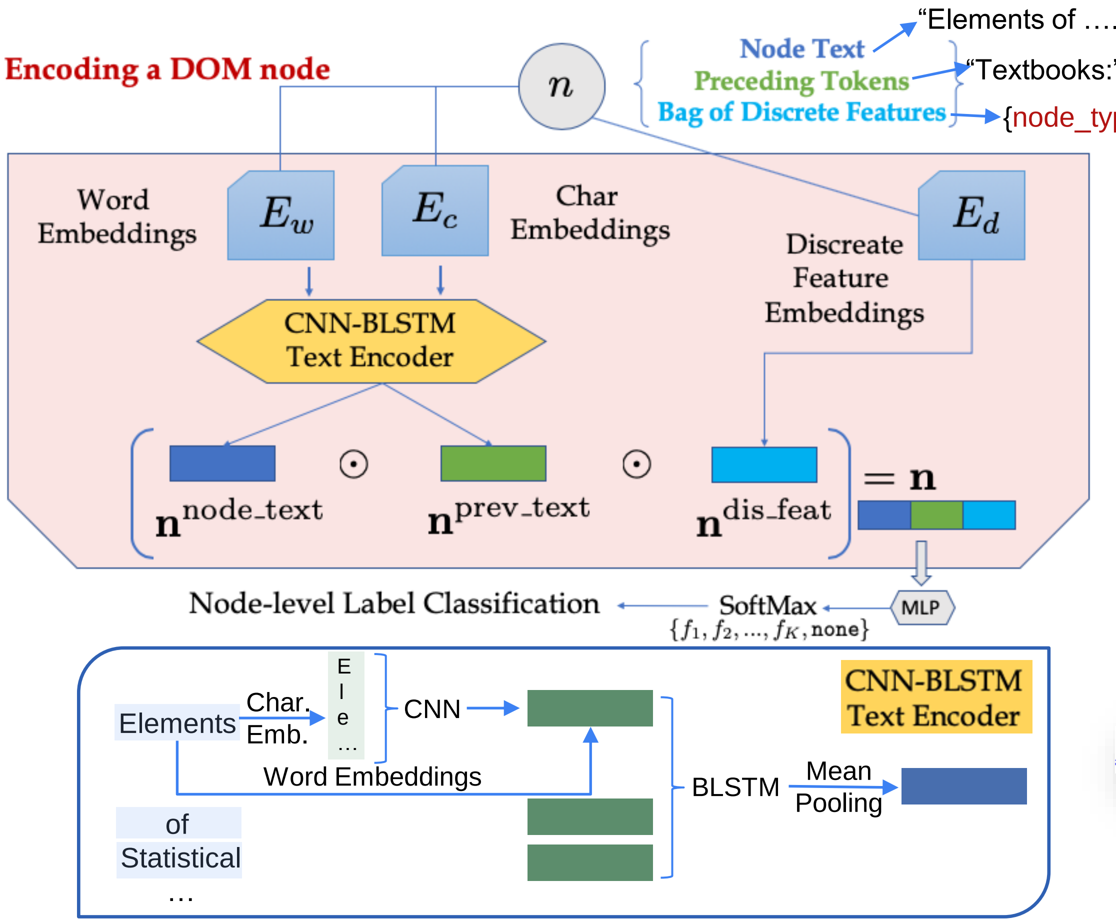
Google Research

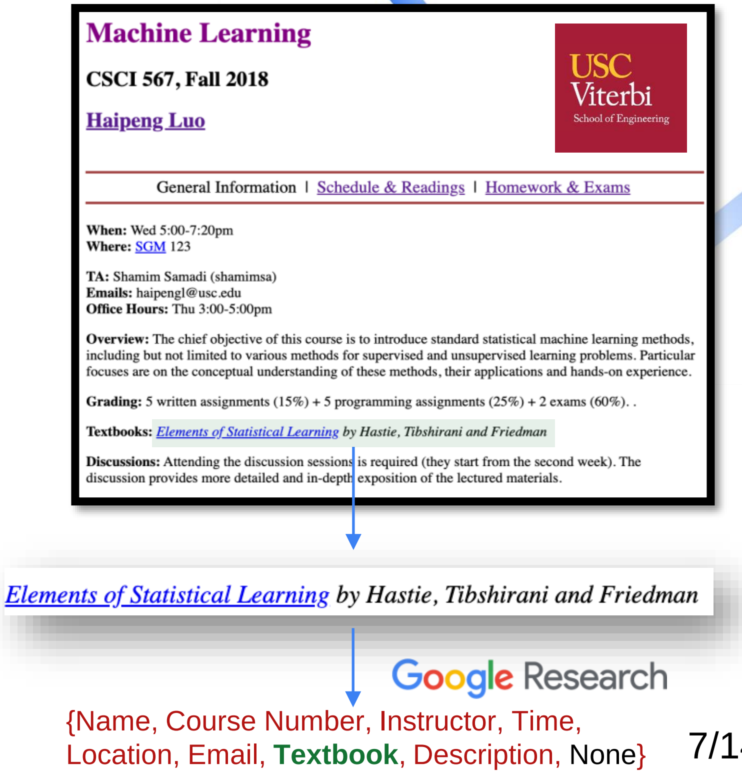
Overview of FreeDOM: A Two-Stage Framework



Google Research

FreeDOM: (1) Learning to encode a DOM node

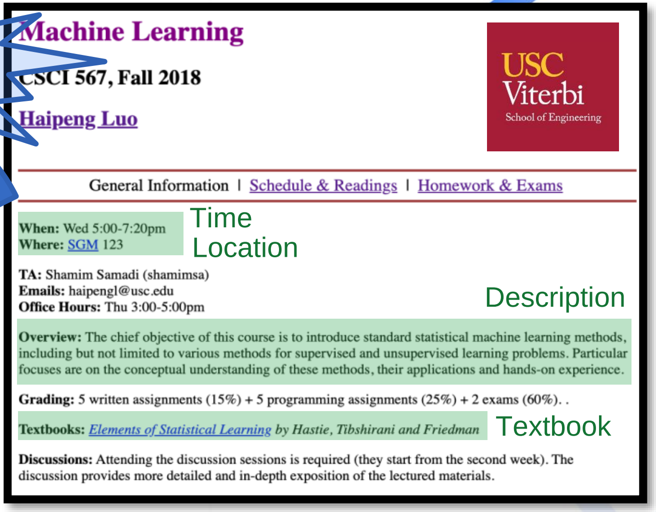




Google Research

Problems of Only Using Node Representations

Similar Dependency



Machine Learning

CSCI 567, Fall 2018

Haipeng Luo

USC Viterbi School of Engineering

General Information | Schedule & Readings | Homework & Exams

When: Wed 5:00-7:20pm
Where: SGM 123

TA: Shamim Samadi (shamimsa)
Emails: haipengl@usc.edu
Office Hours: Thu 3:00-5:00pm

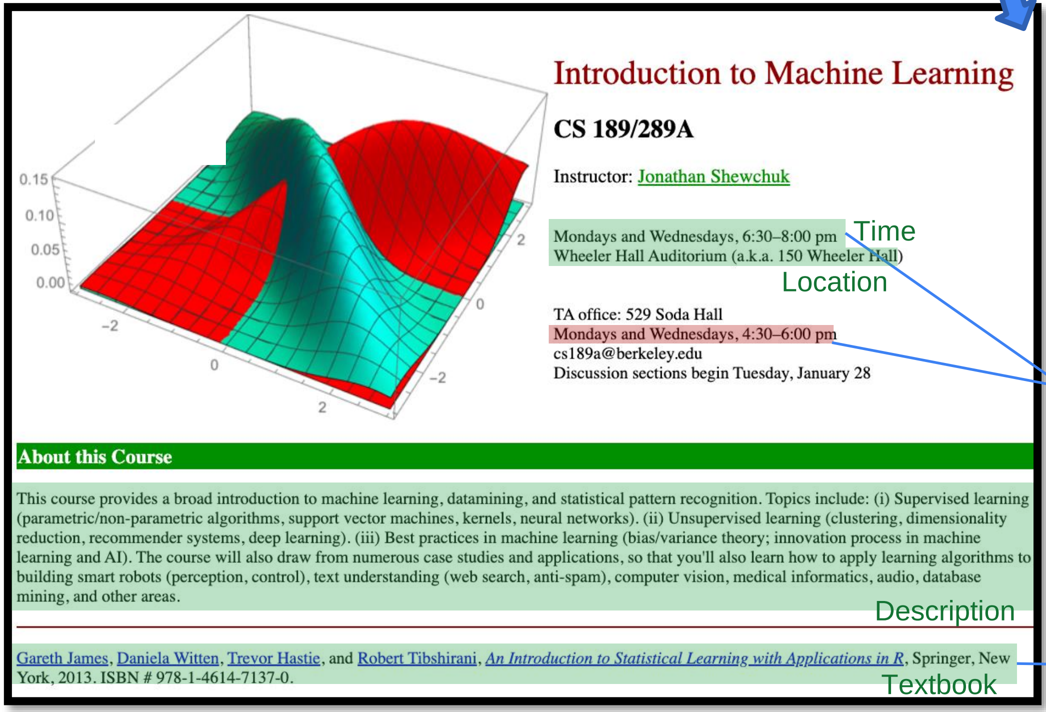
Overview: The chief objective of this course is to introduce standard statistical machine learning methods, including but not limited to various methods for supervised and unsupervised learning problems. Particular focuses are on the conceptual understanding of these methods, their applications and hands-on experience.

Grading: 5 written assignments (15%) + 5 programming assignments (25%) + 2 exams (60%). .

Textbooks: *Elements of Statistical Learning by Hastie, Tibshirani and Friedman*

Discussions: Attending the discussion sessions is required (they start from the second week). The discussion provides more detailed and in-depth exposition of the lectured materials.

A page in the training seed sites



Introduction to Machine Learning

CS 189/289A

Instructor: Jonathan Shewchuk

Mondays and Wednesdays, 6:30–8:00 pm
Wheeler Hall Auditorium (a.k.a. 150 Wheeler Hall)

TA office: 529 Soda Hall
Mondays and Wednesdays, 4:30–6:00 pm
cs189a@berkeley.edu
Discussion sections begin Tuesday, January 28

About this Course

This course provides a broad introduction to machine learning, datamining, and statistical pattern recognition. Topics include: (i) Supervised learning (parametric/non-parametric algorithms, support vector machines, kernels, neural networks). (ii) Unsupervised learning (clustering, dimensionality reduction, recommender systems, deep learning). (iii) Best practices in machine learning (bias/variance theory; innovation process in machine learning and AI). The course will also draw from numerous case studies and applications, so that you'll also learn how to apply learning algorithms to building smart robots (perception, control), text understanding (web search, anti-spam), computer vision, medical informatics, audio, database mining, and other areas.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Springer, New York, 2013. ISBN # 978-1-4614-7137-0.

A page from an unseen site at test time

Misleading Local Node Features

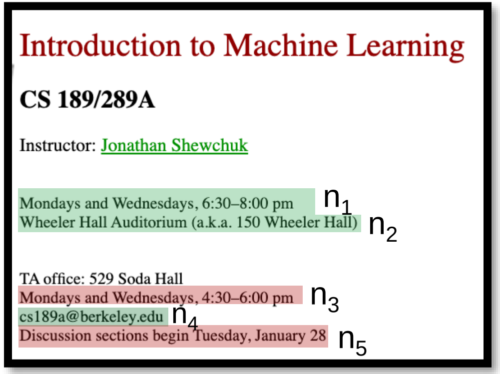
Weak Local features

Google Research

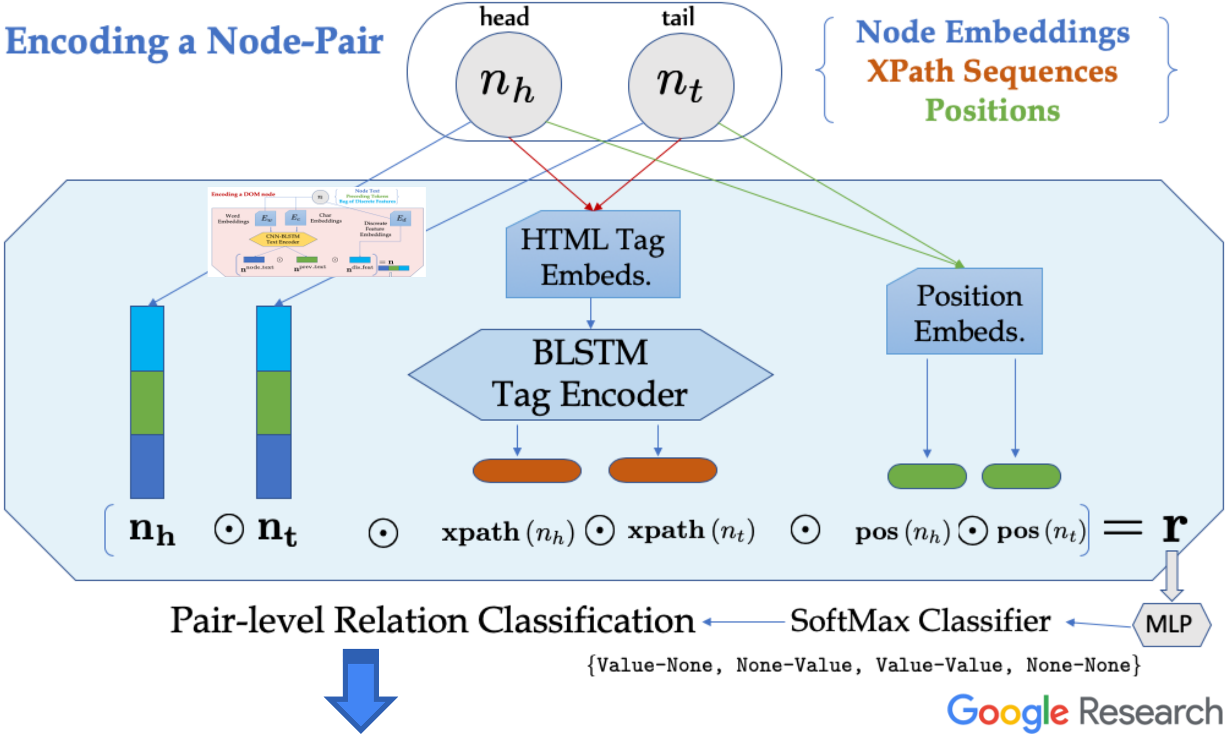
FreeDOM: (2) Learning to encode dependency via pair-wise modeling!

XPath (i.e., a sequence of html tags):
[“<html>”, “<body>”, “<div>”, “<ul>”, “<li>”].

Position embedding: integer2vec



Relation(n_1, n_2) = **Value-Value**
Relation(n_2, n_3) = **Value-None**
Relation(n_3, n_4) = **None-Value**
Relation(n_3, n_5) = **None-None**



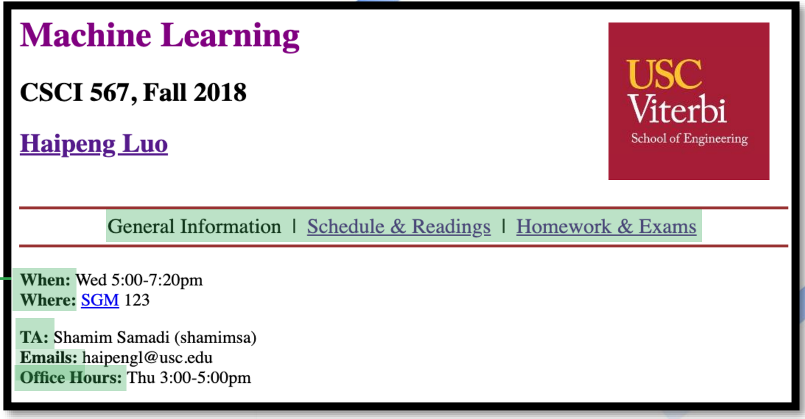
Aggregating scores for node labeling (based on Stage 1)

Google Research

Pre/Post-Processing Tricks

I. Too many nodes?

Variable nodes (with the same XPath) have different contents across different pages. Thus, we can ignore nodes that are common boilerplate, such as navigation bars, headers, footers, etc.



II. Too many node-pairs?

Uncertain fields. We can only look at the node pairs about the most plausible ***m*** nodes that are ranked top by the first-stage node classifier.

III. Site-level constraints?

Majority voting XPath-Fields patterns within each site, for avoiding outlier predictions.

Google Research

Experiment Set Up

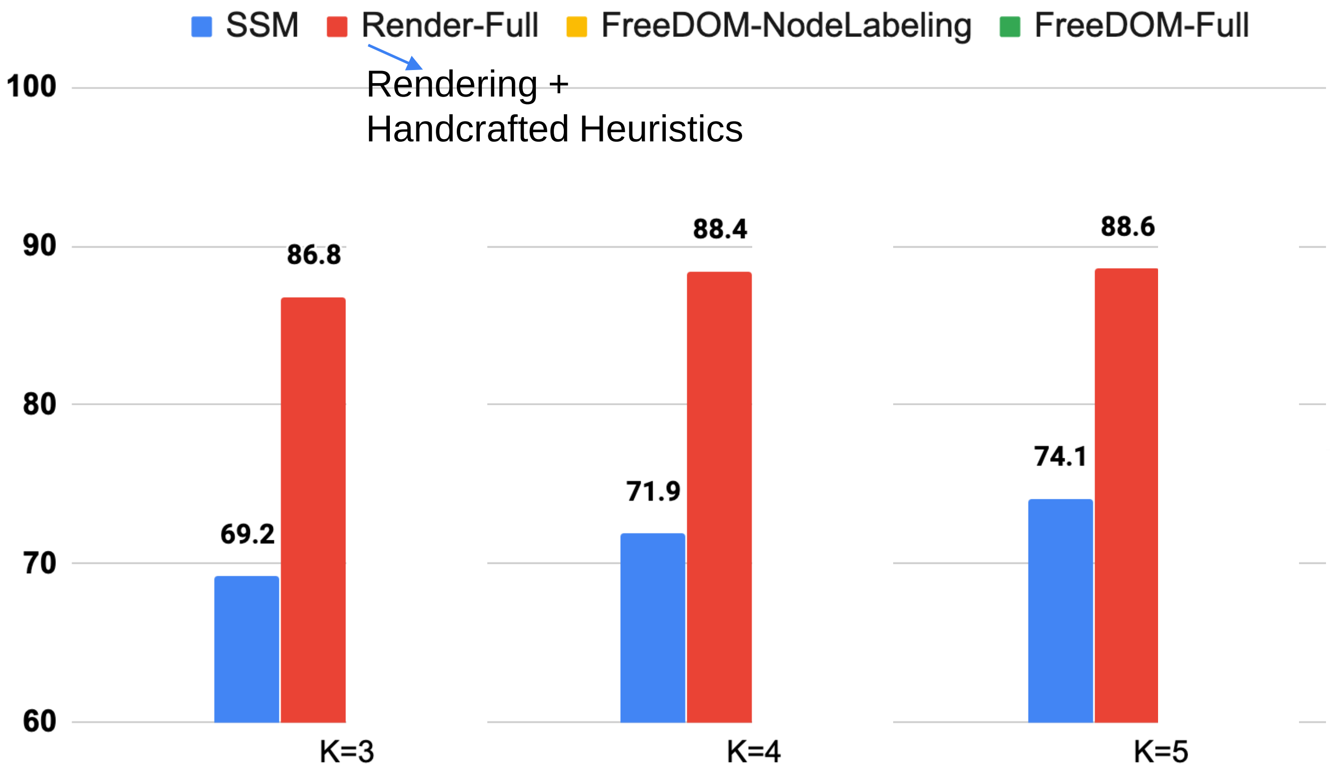
Vertical	#Sites	#Pages	#Var. Nodes	Fields
Auto	10	17,923	130.1	model, price, engine, fuel_economy
Book	10	20,000	476.8	title, author, isbn, pub, date
Camera	10	5,258	351.8	model, price, manufacturer
Job	10	20,000	374.7	title, company, location, date_posted
Movie	10	20,000	284.6	title, director, genre, mpaa_rating
NBA Player	10	4,405	321.5	name, team, height, weight
Restaurant	10	20,000	267.4	name, address, phone, cuisine
University	10	16,705	186.2	name, phone, website, type

The statistics of the SWDE dataset (Hao et al. in Proc. of SIGIR 2011).

- K for training (i.e., seed source sites)
- 10-K for test (i.e., target sites)
- 10 cyclic permutations → Average Performance

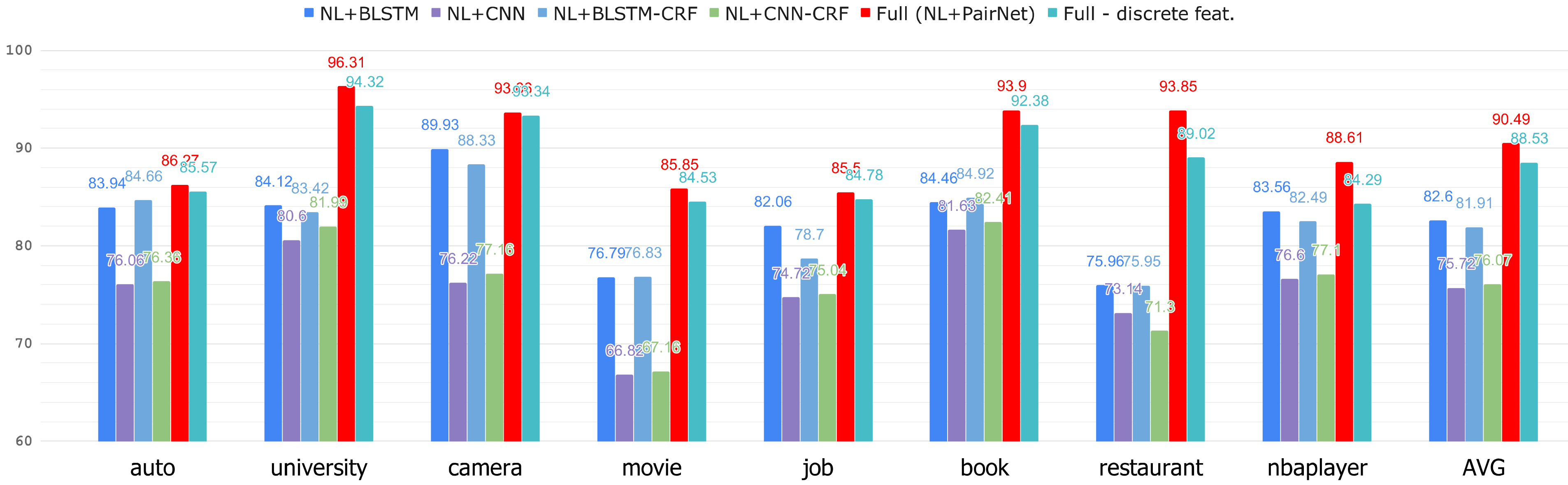
Google Research

Experimental Results on SWDE dataset



Google Research

Ablation Study:
First Stage+ Different Node Tagging Models



Google Research

Conclusion

- We present a novel neural architecture, FreeDOM, for transferrable information extraction on web docs.

- Expensive rendering is not necessary, as FreeDOM can encode the node dependency via pairwise modeling.

- FreeDOM achieves a new state-of-the-art on the SWDE dataset while not using any hand-crafted features or complex heuristic algorithms.

Future Directions based on FreeDOM

- Open Information Extraction?
- Self-supervised pre-training for HTML documents?

Google Research

KDD2020

Thank You!

Google Research

COMPOSE: Cross-Modal Pseudo-Siamese Network for Patient Trial Matching

Junyi Gao[1], Cao Xiao[1], Lucas M. Glass[12], Jimeng Sun[3]

[1]Analytics Center of Excellence, IQVIA

[2]Department of Statistics, Temple University

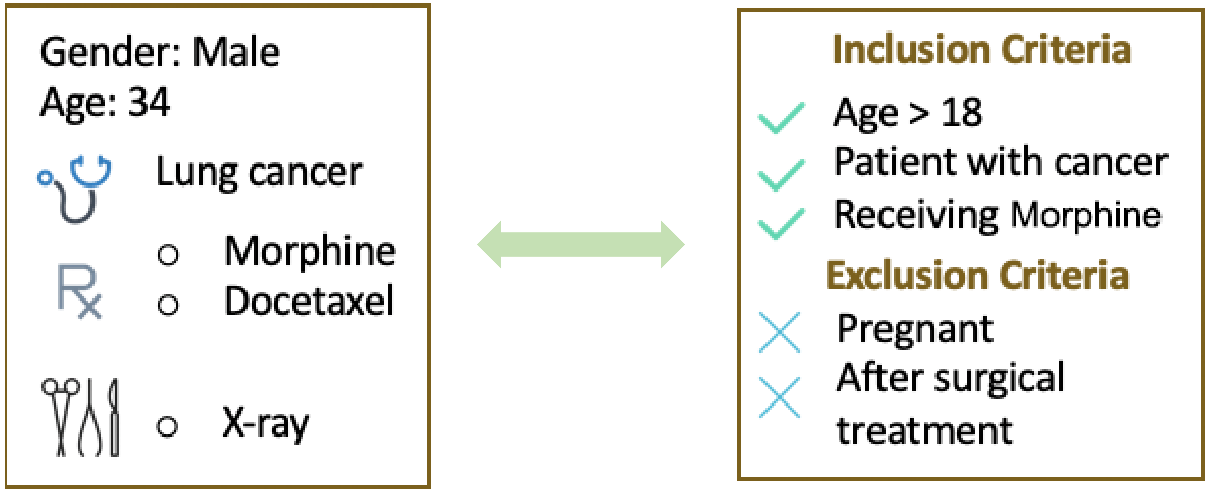
[3]Department of Computer Science, University of Illinois Urbana-Champaign

Content

- Clinical Background
- Challenges
- Method
- Experiment Results

Clinical Background 1: What is patient trial matching?

- Electronic Health Records (EHR): A type of high-dimensional sequence data
  - Procedures
  - Diagnosis
  - Drugs
- Clinical trials: Unstructured text data
  - Inclusion Criteria
  - Exclusion Criteria



Clinical Background 2: Why automated patient trial matching is important?

Essential	Annual market over $46 billion
Time Consuming	50% of trials delayed, 25% of cancer trials failed due to enrollment.
High Costs	High recruitment cost: $6000 to $7500 per patient.

Clinical Background 2: Why automated patient trial matching is important?

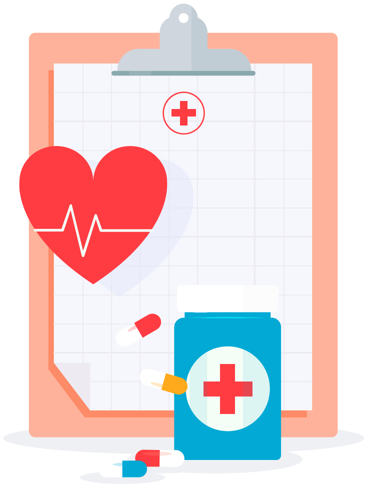
For clinicians	Require huge amount of labor work and expertise knowledge.
For patients	Difficult to find appropriate trials
For recruiters	Need to design criteria carefully

Content

- **Clinical Background**
- **Challenges**
  - Multi-granularity medical concept
  - Many-to-many relationship between patient and trials
  - Explicit inclusion/exclusion criteria handling
- **Method**
- **Experiment Results**

Challenge 1: Multi-granularity medical concept

- Eligibility criteria encode more general disease
- EHRs use more specific medical codes

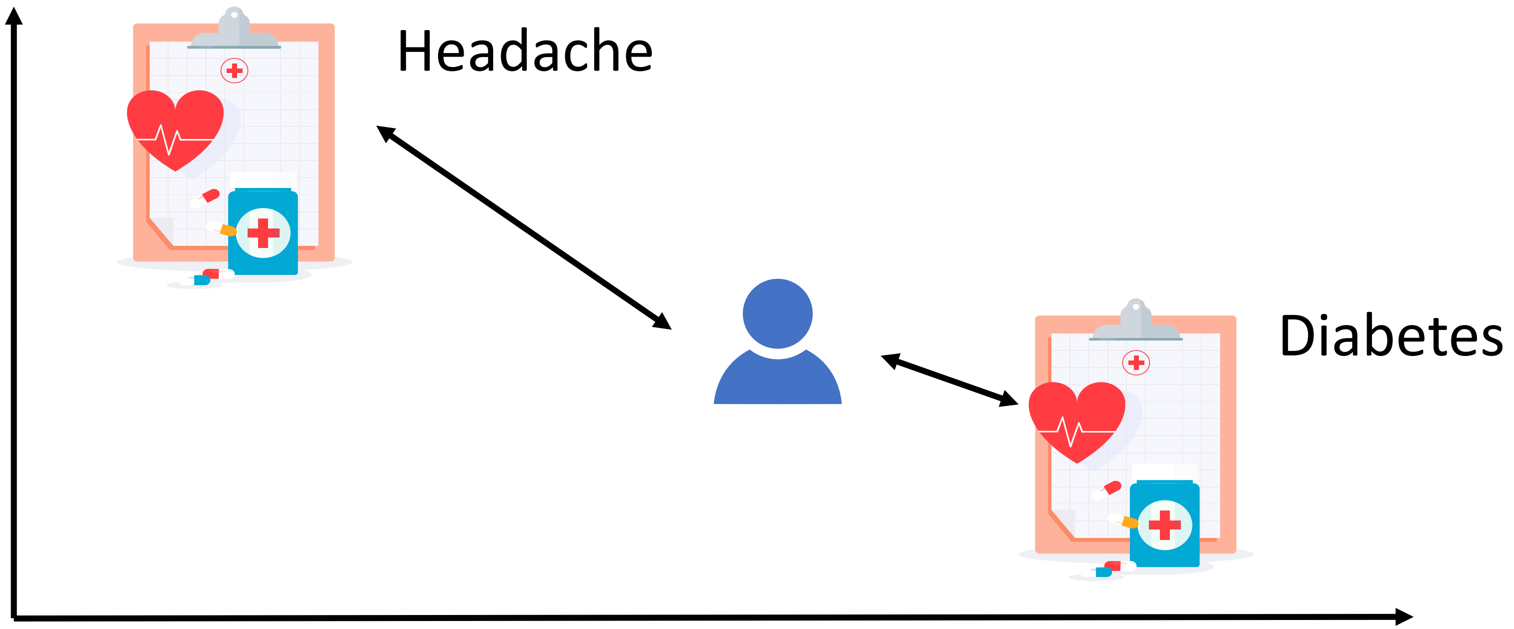


Trial of Cardiovascular Disesases

- ✓ Pleuropericardial adhesion
- ✓ Myocardial infraction
- ✓ Inflammatory cardiomyopathy

Challenge 2: Many-to-many relationship between patients and trials

- Each patient may enroll in more than one trial and vice versa



- Align the patient embedding to different trial embeddings may confuse the embed function

Challenge 3: Explicit inclusion/exclusion criteria handling

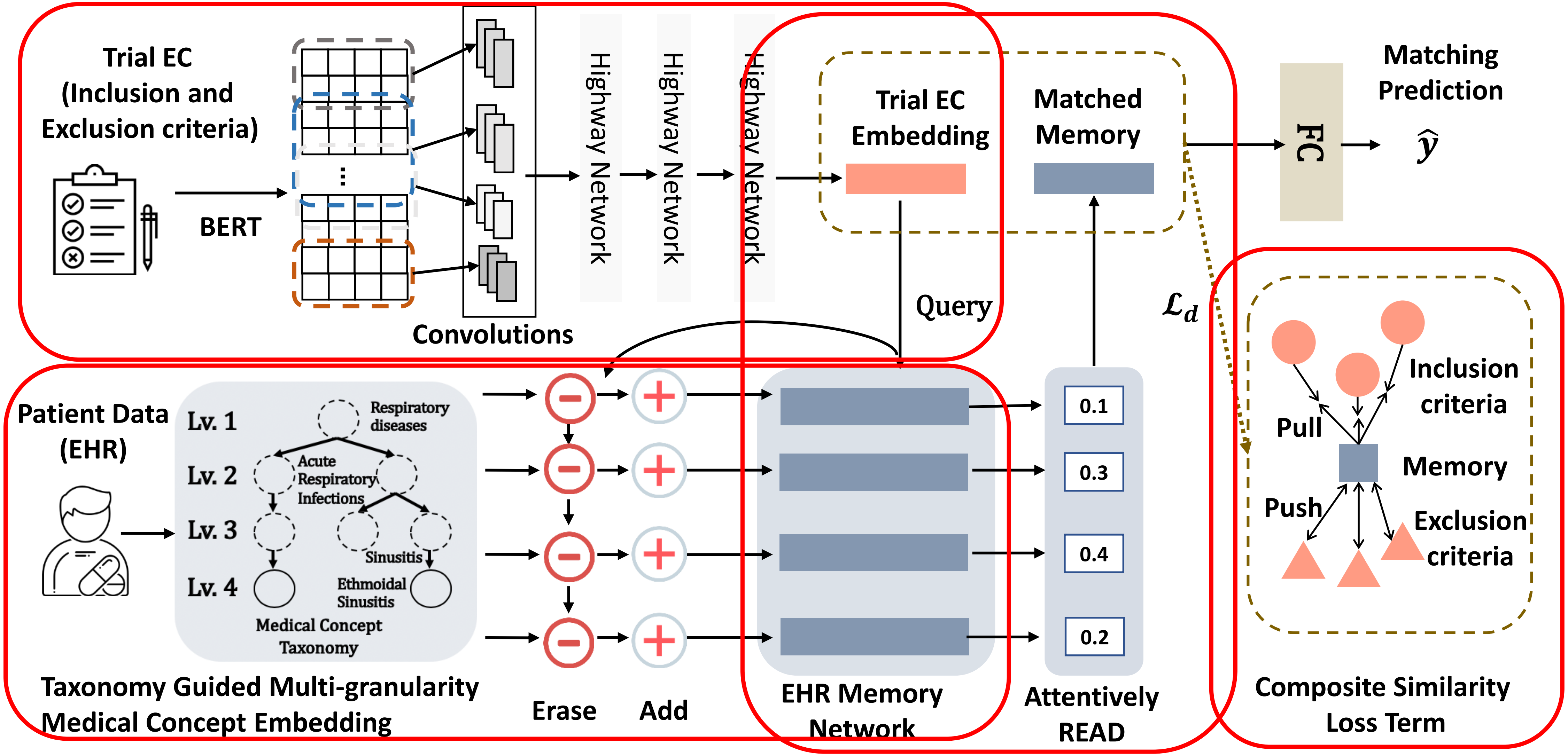
- Inclusion and Exclusion criteria describe desired and unwanted from the targeted patients

Inclusion criteria ⟵ Age > 18 ⟶ Exclusion criteria

Content

- **Clinical Background**
- **Challenges**
- **Method**
  - Trial eligibility criteria embedding
  - Taxonomy guided patient embedding
  - Attentional record alignment and dynamic matching
  - Explicit inclusion/exclusion criteria handling
- **Experiment Results**

Method Overview: COMPOSE



Method: Trial eligibility criteria embedding

- Use BERT to learn contextual embeddings for EC sentence $[w_1, \dots, w_N]$

$$\widetilde{\boldsymbol{c}} = [\widetilde{\boldsymbol{w}}_1, \dots, \widetilde{\boldsymbol{w}}_N] = BERT([\boldsymbol{w}_1, \dots, \boldsymbol{w}_N])$$

- Use different kernel sizes to capture different granularity semantics

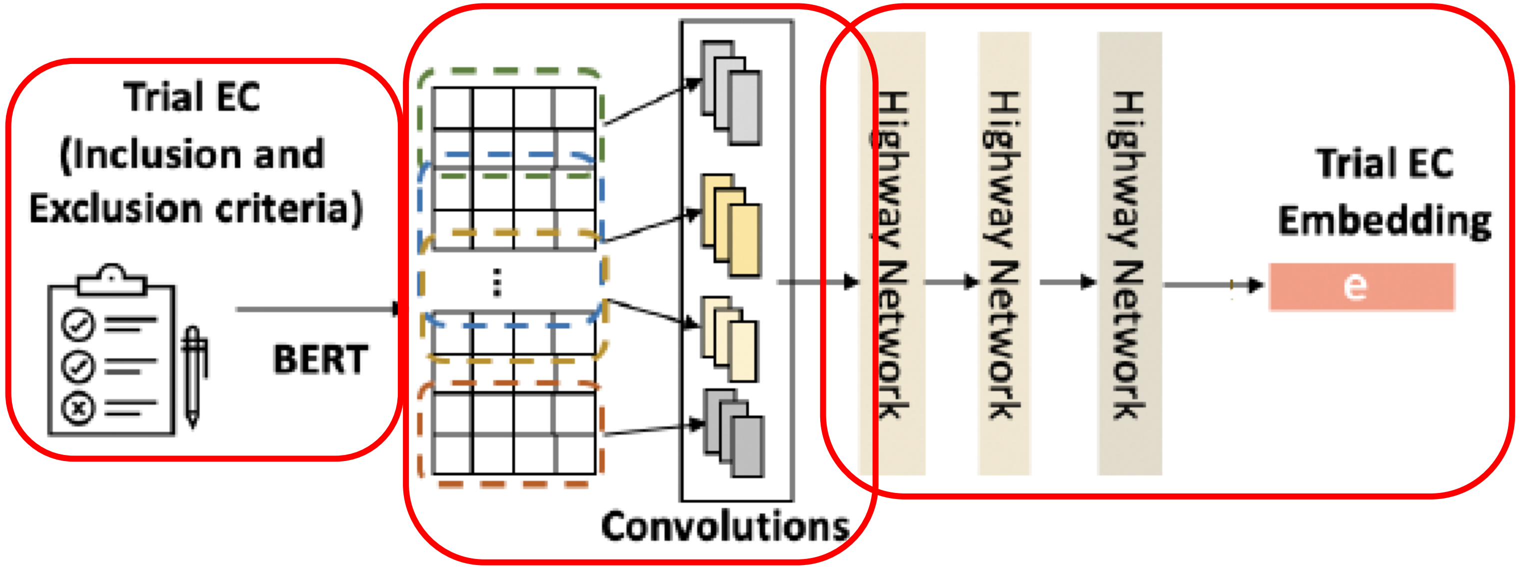
$$\boldsymbol{x} = [Conv(\widetilde{\boldsymbol{c}}, k_1), Conv(\widetilde{\boldsymbol{c}}, k_2), Conv(\widetilde{\boldsymbol{c}}, k_3), Conv(\widetilde{\boldsymbol{c}}, k_4)]$$

- Use highway network and max pooling to obtain the final EC embedidng

$$\boldsymbol{u} = \sigma(Conv(\boldsymbol{x}, k))$$

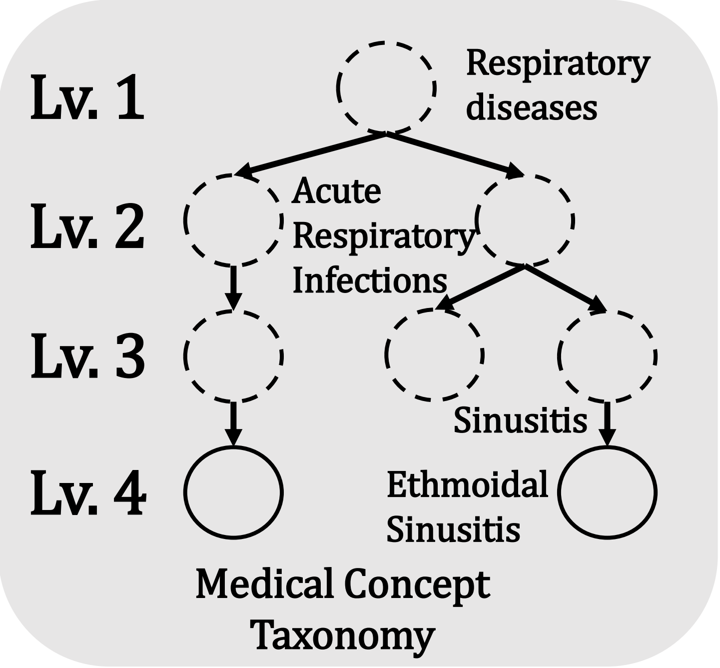
$$\boldsymbol{v} = \boldsymbol{u} \cdot Conv(\boldsymbol{x}, k) + \boldsymbol{x} \cdot (1 - \boldsymbol{u})$$

$$\boldsymbol{e} = MaxPool(\boldsymbol{v})$$

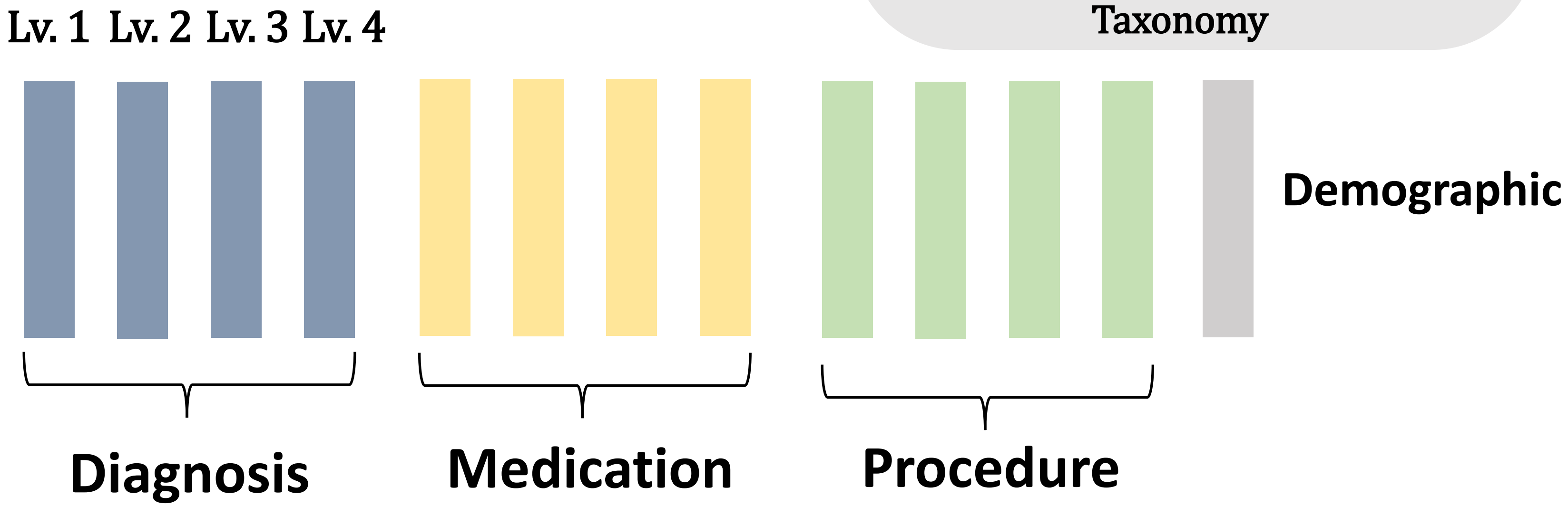


Method: Taxonomy guided patient embedding

- Use medical concept taxonomy to divide each concept into four levels
  - the Uniform System of Classification (USC)
- Three memory networks to store diagnosis, medications and procedures



$$
\begin{aligned}
\mathrm{m} &= [\mathrm{m}_{\mathcal{D}}, \mathrm{m}_{\mathcal{O}}, \mathrm{m}_{\mathcal{P}}] \\
&= [\mathrm{m}_{\mathcal{D}}^{1}, \ldots, \mathrm{m}_{\mathcal{D}}^{4}, \mathrm{m}_{\mathcal{O}}^{1}, \ldots, \mathrm{m}_{\mathcal{O}}^{4}, \mathrm{m}_{\mathcal{P}}^{1}, \ldots, \mathrm{m}_{\mathcal{P}}^{4}]
\end{aligned}
$$



Method: Taxonomy guided patient embedding

- Augment medical codes with textual description:
  - Code 692.9 -> “Contact dermatitis and other eczema”

$$\tilde{g}_t = MaxPool(BERT([w_1, ..., w_L]))$$

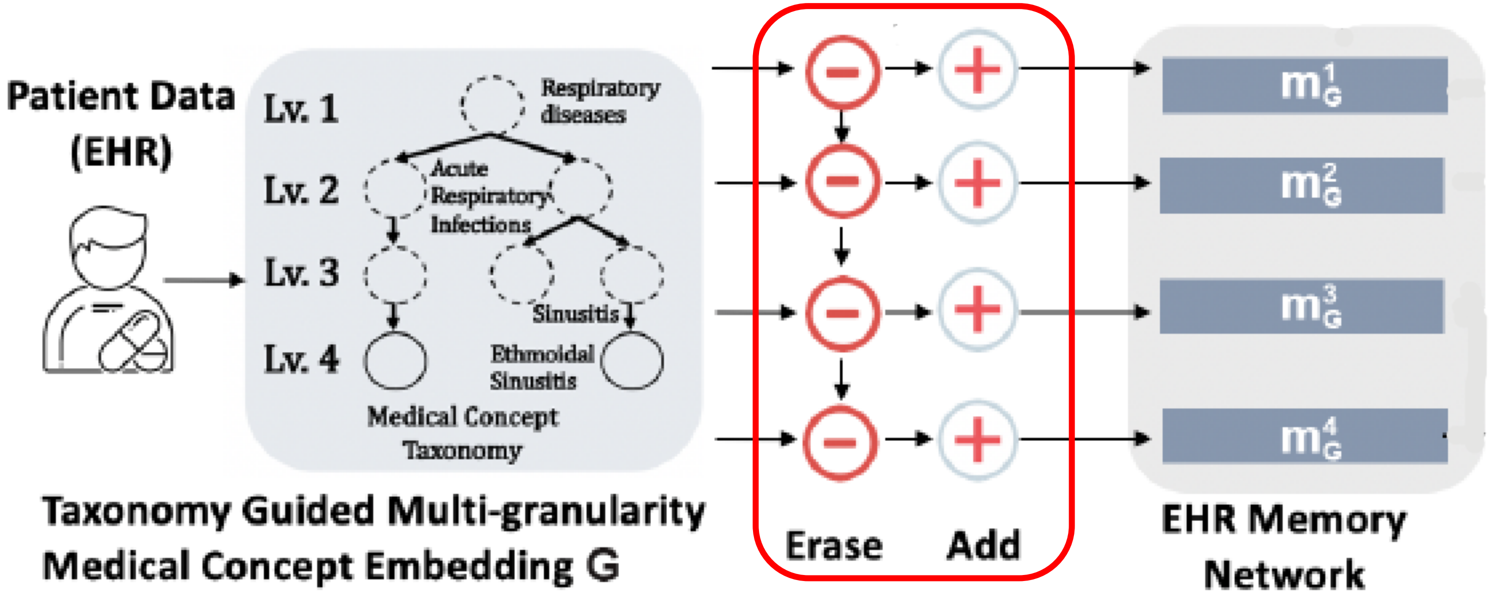
- Update memories at each visit
  - Erase-followed-by-add:

$$\mathbf{erase}_t = \sigma(\mathbf{W}_e \tilde{g}_t^k + \mathbf{b}_e),$$

$$\mathbf{add}_t = tanh(\mathbf{W}_a \tilde{g}_t^k + \mathbf{b}_a)$$

  - Update slot:

$$m_G^k \leftarrow m_G^k \odot (1 - \mathbf{erase}_t) + \mathbf{add}_t$$



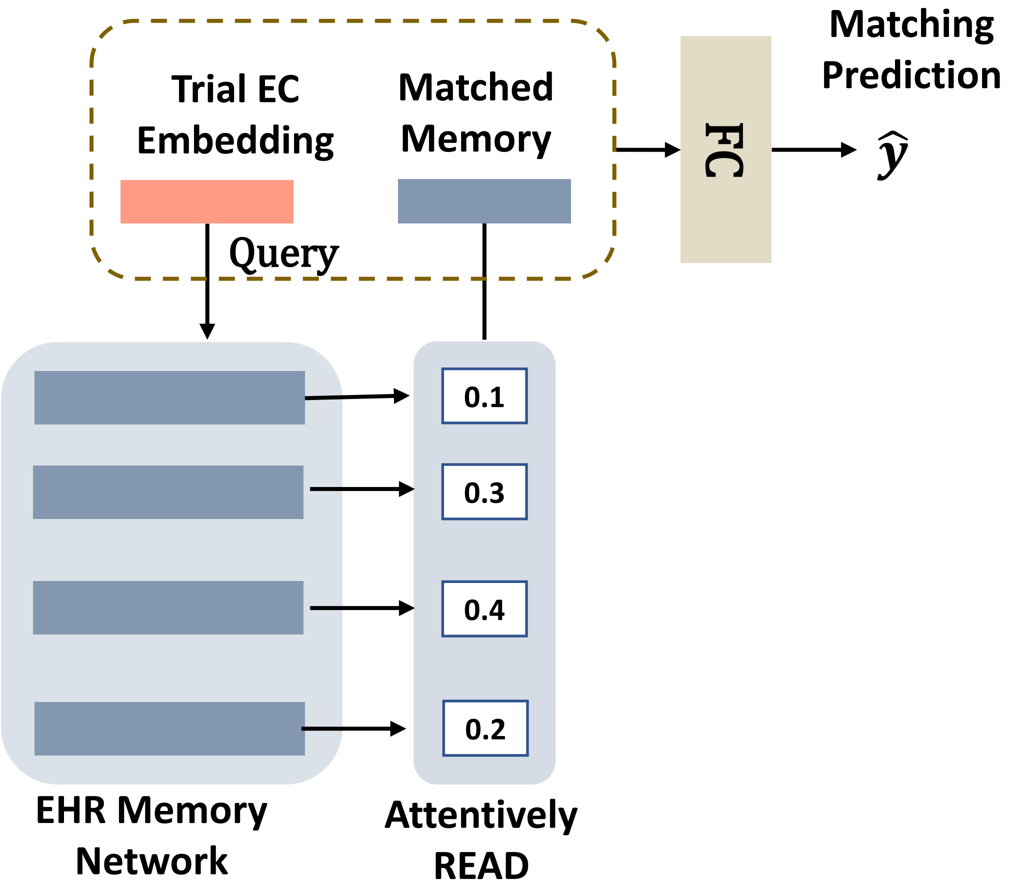
Method: Attentional record alignment and dynamic matching

- Let each EC correspond to the sub-memories

- Attentional matching
  - Trial EC embedding -> Query
  - Matched memory -> Response

$$a_{k,G} = \frac{exp(\boldsymbol{m}_G^{k\,\mathrm{T}} MLP(\boldsymbol{e}))}{\sum_{x\in\{\mathcal{D},\mathcal{O},\mathcal{P}\}} \sum_{i=1}^{4} exp(\boldsymbol{m}_x^{i\,\mathrm{T}} MLP(\boldsymbol{e}))}$$

$$\widetilde{\boldsymbol{m}} = \sum_{x\in\{\mathcal{D},\mathcal{O},\mathcal{P}\}} \sum_{i=1}^{4} a_{i,x} \boldsymbol{m}_x^i$$



Method: Explicit inclusion/exclusion criteria handling

- Classification loss:

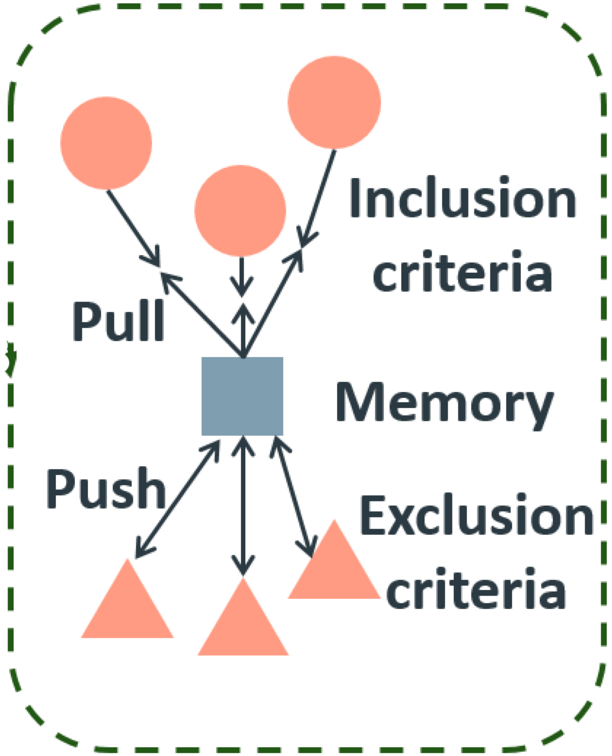
$$\mathcal{L}_c = -(\boldsymbol{y}^{\mathrm{T}} log(\hat{\boldsymbol{y}}) + (1-\boldsymbol{y})^{\mathrm{T}} log(1-\hat{\boldsymbol{y}}))$$

- Inclusion/Exclusion loss:

$$\mathcal{L}_d = \begin{cases} \underbrace{1 - d(\boldsymbol{e}, \widetilde{\boldsymbol{m}}_I)}_{\to 0}, & if\ \boldsymbol{e}\ is\ \boldsymbol{e}_I \\ \underbrace{max(0, d(\boldsymbol{e}, \widetilde{\boldsymbol{m}}_E) - \alpha)}_{>= \alpha}, & if\ \boldsymbol{e}\ is\ \boldsymbol{e}_E \end{cases}$$

- Final loss:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_d$$



Composite Similarity Loss Term

Content

- **Background & Motivation**
- **Problem Formulation**
- **Insights**
- **Solution**
- **Experiment**
  - Patient trial matching
  - Discussions
  - Case studies

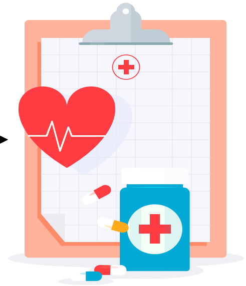
Experiment

- **Dataset**
  - Clinical trial data
    - 590 trials from publicly available data source (clinicaltrials.gov)
    - 12,445 criteria-level EC statements
  - Patient EHR data
    - 83,371 patients from 2002 to 2018

Experiment

- **Label definition**
  - 397,321 labelled pairs
  - The patient matches a trial only if all $(P, IC) = Match$ and $(P, EC) = Mismatch$

P



$Trial: [IC_1, IC_2, \ldots, EC_1, EC_2, \ldots]$

$$(P, IC_1) = Match$$
$$(P, IC_2) = Match$$
$$\ldots$$
$$(P, EC_1) = Mismatch$$
$$(P, EC_2) = Mismatch$$
$$\ldots$$

$$(P, IC_r) = Unknown$$
$$(P, EC_r) = Unknown$$
$$\ldots$$

Experiment: Patient trial matching

- Outperforms all baseline models across both trial level and criteria level matching in all evaluation metrics.

- 24.3% higher accuracy for trial level matching
- 8.8% higher accuracy and 4.7% higher AUROC for criteria level matching

	Model	Accuracy
Baselines	LSTM+GloVe	0.4294±0.010
	LSTM+BERT	0.5460±0.008
	Criteria2Query	0.6147±-
	DeepEnroll	0.6737±0.021
Reduced	COMPOSE-MN	0.7833±0.011
	COMPOSE-Highway	0.8102±0.009
	COMPOSE-$\mathcal{L}_1$	0.8212±0.010
Proposed	COMPOSE	**0.8373±0.012**

	Model	Accuracy	AUROC	AUPRC
Baselines	LSTM+GloVe	0.722±0.010	0.789±0.009	0.784±0.009
	LSTM+BERT	0.834±0.008	0.845±0.007	0.840±0.007
	DeepEnroll	0.869±0.012	0.936±0.013	0.947±0.011
Reduced	COMPOSE-MN	0.899±0.012	0.955±0.013	0.960±0.010
	COMPOSE-Highway	0.912±0.007	0.965±0.007	0.967±0.009
	COMPOSE-$\mathcal{L}_d$	0.939±0.010	0.976±0.009	0.973±0.007
Proposed	COMPOSE	**0.945±0.008**	**0.980±0.007**	**0.979±0.008**

Discussion: Varying length of patient record

- How COMPOSE performs in matching trials with patients who have short or long records?
  - Short (1 visit), Medium (2-3 visits), Long (≥ 4 visits)

- COMPOSE have robust performance

Model	Short	Medium	Long
LSTM+GloVe	0.4906	0.4328	0.0000
LSTM+BERT	0.5484	0.5512	0.5338
Criteria2Query	0.6833	0.5989	0.5172
DeepEnroll	0.6779	0.6797	0.6443
COMPOSE	**0.8420**	**0.8389**	**0.8350**

Discussion: Varying disease types

- How COMPOSE performs on different types of diseases?
  - Chronic, Oncology, Rare diseases

- Achieves 77.3% higher accuracy for chronic diseases
- Most baseline models fail to match correct patients for oncology and rare diseases

Model	Chronic Diseases	Oncology	Rare Diseases
LSTM+GloVe	0.1793	0.0000	0.0000
LSTM+BERT	0.2062	0.0000	0.0000
Criteria2Query	0.5103	0.2722	0.2292
DeepEnroll	0.3345	0.0000	0.0000
COMPOSE	**0.5931**	**0.6370**	**0.6875**

Discussion: Varying trial phases

- How COMPOSE performs on different phases?
  - Phase I, II, III

- 155% higher accuracy for phase I trials
- 19% higher accuracy for phase II trials
- 27% higher accuracy for phase III trials

Model	Phase I	Phase II	Phase III
LSTM+GloVe	0.0008	0.5865	0.3743
LSTM+BERT	0.0025	0.6045	0.4862
Criteria2Query	0.3025	0.6433	0.5870
DeepEnroll	0.2034	0.7493	0.6329
COMPOSE	**0.5189**	**0.8939**	**0.8005**

Discussion: Varying threshold of matching

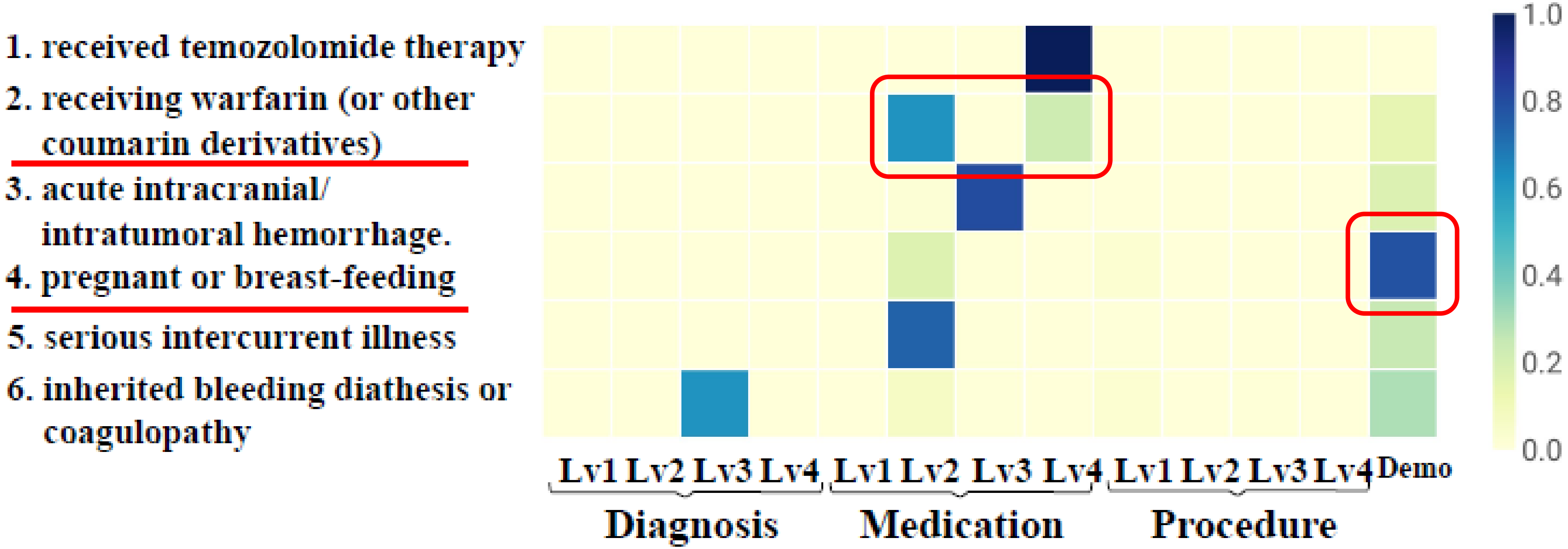
- Some inclusion or exclusion criteria can be too strict to prevent finding patients
- How COMPOSE performs on varying thresholds?
  - 70%, 80%, 90%

- COMPOSE have robust performance under all thresholds

Model	70% Matching	80% Matching	90% Matching
LSTM+GloVe	0.6218	0.5862	0.5057
LSTM+BERT	0.7231	0.6861	0.6238
DeepEnroll	0.8225	0.7963	0.7422
COMPOSE	**0.9334**	**0.9193**	**0.8915**

Case study: Attention weights on memory slots

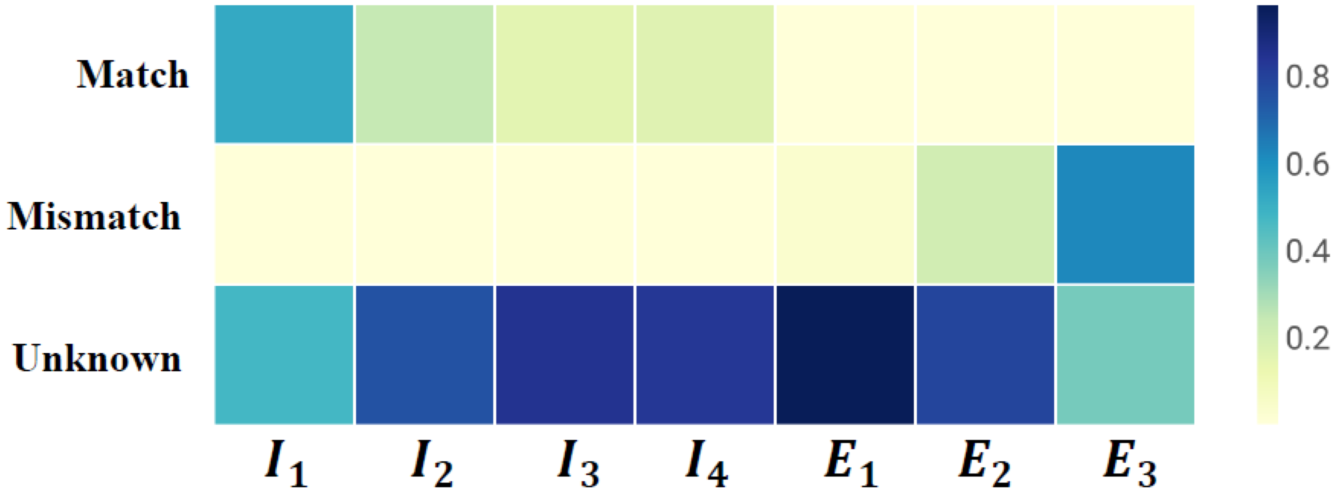
- A trial on Cabozantinib which treats grade IV astrocytic tumors



Case study: Failed case

- A trial for *Early Stage Non-Small Cell Lung Cancer*

- I2: Lung function capacity capable of tolerating the proposed lung surgery
- I3: Eastern Cooperative Oncology Group (ECOG) Performance Status of 0-1
- I4: Available tissue of primary lung tumor



Thank you!

COMPOSE: Cross-Modal Pseudo-Siamese Network for Patient Trial Matching

Personal Homepage

http://aboutme.vixerunt.org/

Paper Link

https://arxiv.org/abs/2006.08765

Source code

https://github.com/v1xerunt/COMPOSE

Probabilistic Metric Learning with Adaptive Margin for Top-K Recommendation

Chen Ma[1], Liheng Ma[1], Yingxue Zhang[2], Ruiming Tang[3], Xue Liu[1] and Mark Coates[1]

[1]McGill University, Montreal, Canada

[2]Huawei Noah's Ark Lab, Montreal, Canada

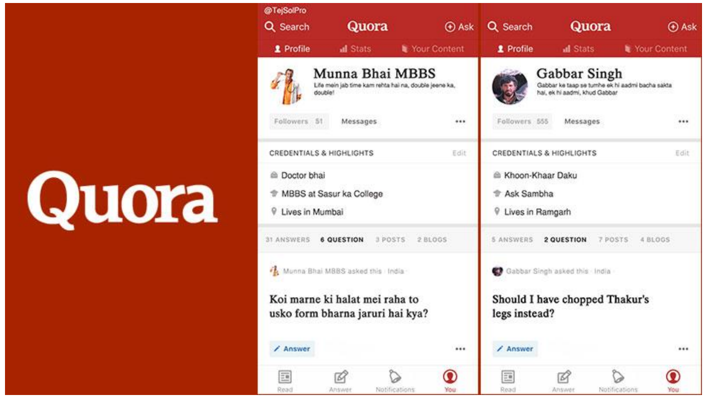
[3]Huawei Noah's Ark Lab, Shenzhen, China

SIGKDD 2020

Background

- The rapid growth of Internet services allows users to access millions of online products, such as movies, articles.

- The large amount of user-item data facilitates a promising and practical service – the **personalized recommendation**.



Background

- Typically, the recommendation problem focuses on the user-item interaction/rating matrix.

user \ movie	1	2	3	4	5
1			✓		✓
2	✓				
3		✓			✓
4			✓		
5				✓	✓

Recommendation: based on observed user preference on items, recommending some **new** K items that users are **interested in**.

Background

- Typically, the recommendation problem focuses on the user-item interaction/rating matrix.

user \ movie	1	2	3	4	5
1			✓		✓
2	✓				
3		✓			✓
4			✓		
5				✓	✓

Pointwise: learn the absolute value of each entry, e.g., $\hat{r}_{1,3} \approx 1$

Pairwise: learn the pairwise item relation, e.g., $\hat{r}_{1,3} > \hat{r}_{1,4}$

Background

- Pairwise and Pointwise methods both can achieve promising performance in Top-K recommendation
  - **Pairwise** methods are computation-efficient

- The **inner product** and **distance calculation** both can capture the pairwise relation between items
  - Distance has a major benefit: it guarantees the triangle inequality $d(j,k) \leq d(j,i) + d(i,k)$
  - Applying the distance as the scoring function becomes popular

Hsieh et al., "Collaborative Metric Learning", WWW 2017

Background

- Distance learning for recommendation
  - **Distance calculation**:

    $$d(\mathbf{u}_i, \mathbf{v}_j) = ||\mathbf{u}_i - \mathbf{v}_j||$$

    $\mathbf{u}_i, \mathbf{v}_j$: learnable embeddings of users and items

  - **Loss function**:

    $$\mathcal{L}^{hinge} = \sum_{j \in \mathcal{S}_i} \sum_{k \notin \mathcal{S}_i} [m + d(\mathbf{u}_i, \mathbf{v}_j) - d(\mathbf{u}_i, \mathbf{v}_k)]_+ \qquad [z]_+ = \max(z, 0)$$

    $\mathcal{S}_i$: the item set user *i* has interacted (*j* is the **positive** item and *k* is the **negative**)

    m : the safe margin (a hyper-parameter with a fixed value)

Drawbacks in Distance Learning Methods

- D1: Learning deterministic embeddings without handling uncertainty.

- D2: The margin in the loss function is fixed during training.

- D3: The user-user and item-item relations are neglected.

Probabilistic Distance Learning for D1

- Represent users and items as Gaussian distributions
  - $\mathbf{u}_i \sim \mathcal{N}(\mu_i^{(U)}, \mathbf{\Sigma}_i^{(U)})$, $\mathbf{v}_j \sim \mathcal{N}(\mu_j^{(I)}, \mathbf{\Sigma}_j^{(I)})$
  - $\mu \in \mathbb{R}^h$, $\mathbf{\Sigma} \in \mathbb{R}^h$ (diagonal matrix) are parameters to be learned.
  - The uncertainty can be captured by the covariance matrix

- The distance between Gaussian distributions
  - **Wasserstein distance** has a neat form between two Gaussian distributions
  - $\mathcal{W}_2(i,j)^2 = ||\mu_i^{(U)} - \mu_j^{(I)}||_2^2 + ||(\mathbf{\Sigma}_i^{(U)})^{\frac{1}{2}} - (\mathbf{\Sigma}_j^{(I)})^{\frac{1}{2}}||_2^2$

Adaptive Margin for D2

- We apply an adaptive margin in the loss function:

$$\mathcal{L}_{Fix}(i,j,k;\Theta) = [d(i,j;\Theta)^2 - d(i,k;\Theta)^2 + \boxed{m}]_+$$

$$\mathcal{L}_{Ada}(i,j,k;\Theta,\Phi) = [d(i,j;\Theta)^2 - d(i,k;\Theta)^2 + \boxed{f(i,j,k;\Phi)}]_+$$

- We formulate the margin learning and model learning as:

$$\min_{\Phi} \mathcal{J}_{outer}\left(\Theta^*(\Phi)\right) := \sum_{i} \sum_{j \in \mathcal{S}_i} \sum_{k \notin \mathcal{S}_i} \mathcal{L}_{Fix}\left(i,j,k;\Theta^*(\Phi)\right)$$

$$\text{s.t. } \Theta^*(\Phi) = \operatorname*{argmin}_{\Theta} \mathcal{J}_{inner}(\Theta,\Phi) := \sum_{i} \sum_{j \in \mathcal{S}_i} \sum_{k \notin \mathcal{S}_i} \mathcal{L}_{Ada}(i,j,k;\Theta,\Phi)$$

Θ: the model parameters (μ, Σ) Φ: the parameters related to margin generation

Adaptive Margin for D2

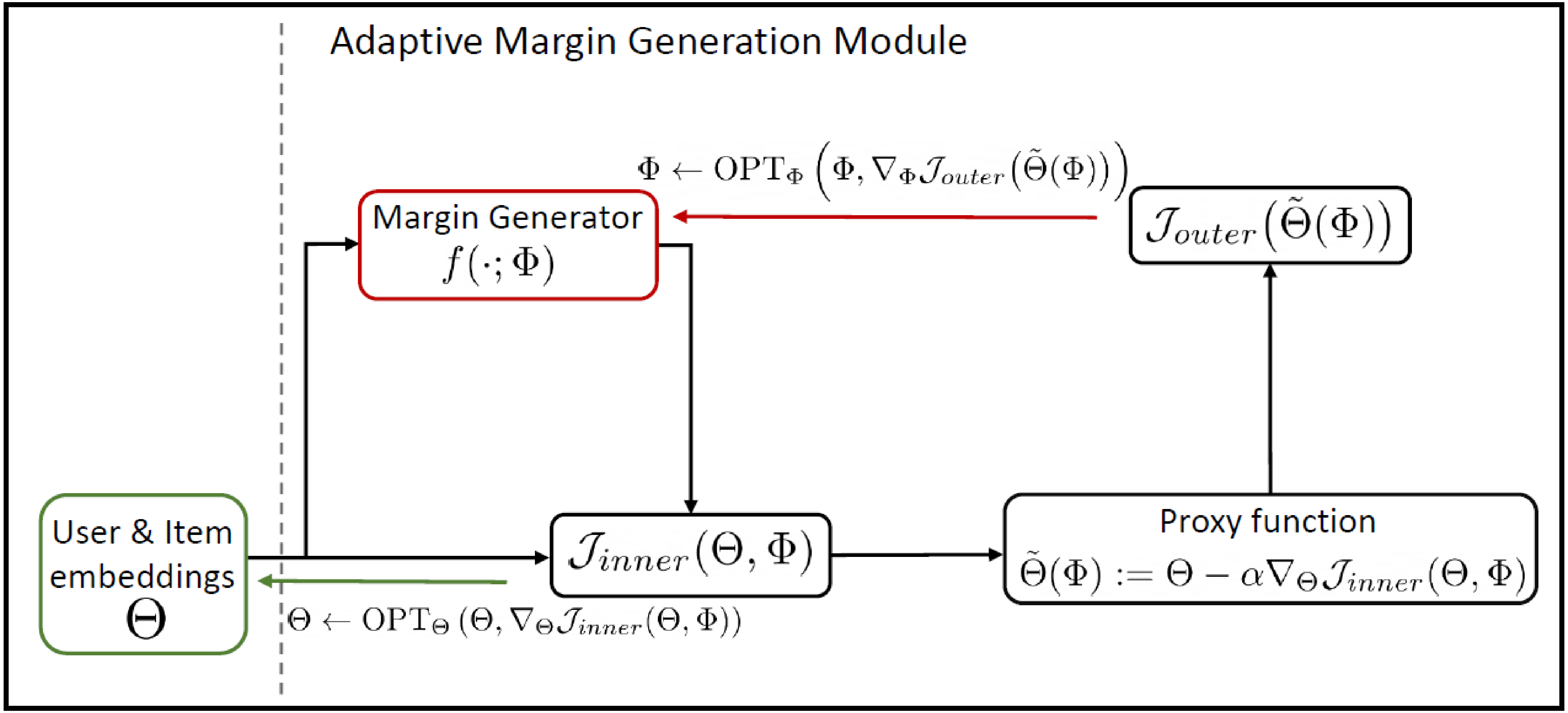
- Training strategy:
  - Θ *update phase* (Inner Optimization): Fix Φ and optimize Θ.
  - Φ *update phase* (Outer Optimization): Fix Θ and optimize Φ.

- The update of Φ :
  - We build a proxy function to link the update of Φ with the outer optimization

$$\Theta^*(\Phi) \approx \tilde{\Theta}(\Phi) := \Theta - \alpha \boxed{\nabla_\Theta \mathcal{J}_{inner}(\Theta, \Phi)}$$

  - By optimizing the outer loss, the gradient w.r.t to Φ can be passed through $\nabla_\Theta \mathcal{J}_{inner}(\Theta, \Phi)$

Adaptive Margin for D2



Adaptive Margin for D2

- Training procedure:

Algorithm 1: Iterative Optimization Procedure

Initialize optimizers OPT_Θ and OPT_Φ ;
while *not converged* **do**
  Θ Update (fix Φ^t):
    $\Theta^{t+1} \longleftarrow \mathrm{OPT}_\Theta\left(\Theta^t, \nabla_{\Theta^t}\mathcal{J}_{inner}(\Theta^t, \Phi^t)\right)$;
  Proxy:
    $\tilde{\Theta}^{t+1}(\Phi^t) := \Theta^t - \alpha\nabla_{\Theta^t}\mathcal{J}_{inner}(\Theta^t, \Phi^t)$;
  Φ Update (fix Θ^t):
    $\Phi^{t+1} \longleftarrow \mathrm{OPT}_\Phi\left(\Phi^t, \nabla_{\Phi^t}\mathcal{J}_{outer}(\tilde{\Theta}^{t+1}(\Phi^t))\right)$;
end

- The design of *f*():

$$\mathbf{z}_{ijk} = \tanh(\mathbf{W}_1 \cdot \mathbf{s}_{ijk} + \mathbf{b}_1)$$
$$m_{ijk} = \mathrm{softplus}(\mathbf{W}_2 \cdot \mathbf{z}_{ijk} + \mathbf{b}_2)$$

$\mathbf{s}_{ijk}$: the input of the two-layer MLP

$\mathrm{softplus}$: make the generated margin positive

User-user and Item-item Relations for D3

- User-user and item-item relations can regularize the model
  - Similar users or items should not be mapped too far in the latent space
  - We apply the hinge loss with adaptive margin mechanism to regularize similar users and items

$$\begin{cases} \mathcal{J}_{outer}^{U-U} & := \sum_i \sum_{p \in \mathcal{N}_i^U} \sum_{q \notin \mathcal{N}_i^U} \mathcal{L}_{Fix}(i, p, q; \tilde{\Theta}_{U-U}^{t+1}), \\ \mathcal{J}_{inner}^{U-U} & := \sum_i \sum_{p \in \mathcal{N}_i^U} \sum_{q \notin \mathcal{N}_i^U} \mathcal{L}_{Ada}(i, p, q; \Theta^t, \Phi_{U-U}^t) \end{cases}$$

$$\begin{cases} \mathcal{J}_{outer}^{I-I} & := \sum_j \sum_{p \in \mathcal{N}_j^I} \sum_{q \notin \mathcal{N}_j^I} \mathcal{L}_{Fix}(j, p, q; \tilde{\Theta}_{I-I}^{t+1}), \\ \mathcal{J}_{inner}^{I-I} & := \sum_j \sum_{p \in \mathcal{N}_j^I} \sum_{q \notin \mathcal{N}_j^I} \mathcal{L}_{Ada}(j, p, q; \Theta^t, \Phi_{I-I}^t), \end{cases}$$

Evaluation

- Five datasets

Dataset	#Users	#Items	#Interactions	Density
Books	77,754	66,963	2,517,343	0.048%
Electronics	40,358	28,147	524,906	0.046%
CDs	24,934	24,634	478,048	0.079%
Comics	37,633	39,623	2,504,498	0.168%
Gowalla	64,404	72,871	1,237,869	0.034%

We employ the five-fold cross-validation to evaluate our model.

- Evaluation Metrics
  - Recall@5, 10, 15, 20
  - NDCG@5, 10, 15, 20 (normalized discounted cumulative gain)

Evaluation Baselines

BPR: Bayesian personalized ranking, UAI' 2009 → **Classical CF methods**

NCF: Neural Collaborative Filtering, WWW' 2017

DeepAE: Deep Autoencoder, CIKM' 2018

DL-based Recommendation

CML: Collaborative Metric Learning, WWW' 2017

LRML: Latent Relational Metric Learning, WWW' 2018

TransCF: Collaborative Translational Metric Learning, ICDM' 2018

SML: Symmetric Metric Learning with adaptive margin, AAAI' 2020

Distance-based Recommendation

Evaluation Results

	BPRMF	NCF	DeepAE	CML	LRML	TransCF	SML	PMLAM	Improv.
Recall@10									
Books	0.0553	0.0568	<u>0.0817</u>	0.0730	0.0565	0.0754	0.0581	**0.0885****	8.32%
Electronics	0.0243	0.0277	0.0253	<u>0.0395</u>	0.0299	0.0353	0.0279	**0.0469*****	18.73%
CDs	0.0730	0.0759	0.0736	<u>0.0922</u>	0.0822	0.0851	0.0793	**0.1129*****	22.45%
Comics	0.1966	0.2092	<u>0.2324</u>	0.1934	0.1795	0.1967	0.1713	**0.2417**	4.00%
Gowalla	0.0888	0.0895	<u>0.1113</u>	0.0840	0.0935	0.0824	0.0894	**0.1331*****	19.58%
NDCG@10									
Books	0.0391	0.0404	<u>0.0590</u>	0.0519	0.0383	0.0542	0.0415	**0.0671****	13.72%
Electronics	0.0111	0.0125	0.0134	<u>0.0178</u>	0.0117	0.0148	0.0105	**0.0234*****	31.46%
CDs	0.0383	0.0402	0.0411	<u>0.0502</u>	0.0420	0.0461	0.0423	**0.0619*****	23.30%
Comics	0.2247	0.2395	<u>0.2595</u>	0.2239	0.1922	0.2341	0.1834	**0.2753***	6.08%
Gowalla	0.0806	0.0822	<u>0.0944</u>	0.0611	0.0670	0.0611	0.0823	**0.0984***	4.23%

*: $p <= 0.05$, ** $p < 0.01$, ***: $p < 0.001$

Our model outperforms other methods significantly on most of the datasets

Evaluation Results

- Ablation study

Architecture	*CDs*		*Electronics*	
	R@10	N@10	R@10	N@10
(1) Fix^{U-I} + Deter_Emb	0.0721	0.0371	0.0241	0.0090
(2) Fix^{U-I} + Gauss_Emb	0.0815	0.0434	0.0296	0.0110
(3) Ada^{U-I} + Deter_Emb	0.0777	0.0415	0.0338	0.0125
(4) Ada^{U-I}-*cat* + Deter_Emb	0.0408	0.0204	0.0139	0.0055
(5) Ada^{U-I}-*add* + Deter_Emb	0.0311	0.0158	0.0050	0.0018
(6) Ada^{U-I} + Gauss_Emb	0.0856	0.0454	0.0365	0.0155
(7) Ada^{U-I} + Fix^{U-U} + Fix^{I-I}	0.0966	0.0526	0.0429	0.0189
(8) PMLAM	**0.1129**	**0.0619**	**0.0469**	**0.0234**

- Probabilistic embeddings improve the performance
- Adaptive margin scheme works
- User-user/item-item relations are important

Evaluation Results

- Case study

User	Positive	Sampled Movie	Margin
405	*Scream* (Thriller)	*Four Rooms* (Thriller)	**1.2752**
		Toy Story (Animation)	12.8004
	French Kiss (Comedy)	*Addicted to Love* (Comedy)	**2.6448**
		Batman (Action)	12.4607
66	*Air Force One* (Action)	*GoldenEye* (Action)	**0.3216**
		Crumb (Documentary)	5.0010
	The Godfather (Crime)	*The Godfather II* (Crime)	**0.0067**
		Terminator (Sci-Fi)	3.6335

Conclusion

- Each user and item in our model are represented by **Gaussian distributions** with learnable parameters to handle the uncertainties.
- By incorporating an adaptive margin scheme, our model can generate **fine-grained margins** for the training triples during the training procedure.
- Explicitly model the **user-user/item-item** relations.
- Experimental results show that the proposed method outperforms the state-of-the-art methods significantly.

Thank you!

Q & A

Email: chen.ma2@mail.mcgill.ca
allenjack.github.io

Robust Spammer Detection by Nash Reinforcement Learning

Yingtong Dou (UIC) **Guixiang Ma (Intel Labs)**

Philip S. Yu (UIC) **Sihong Xie (Lehigh)**

ydou5@uic.edu

Paper: http://arxiv.org/abs/2006.06069
Slides: http://ytongdou.com/files/kdd20slides.pdf
Code: https://github.com/YingtongDou/Nash-Detect

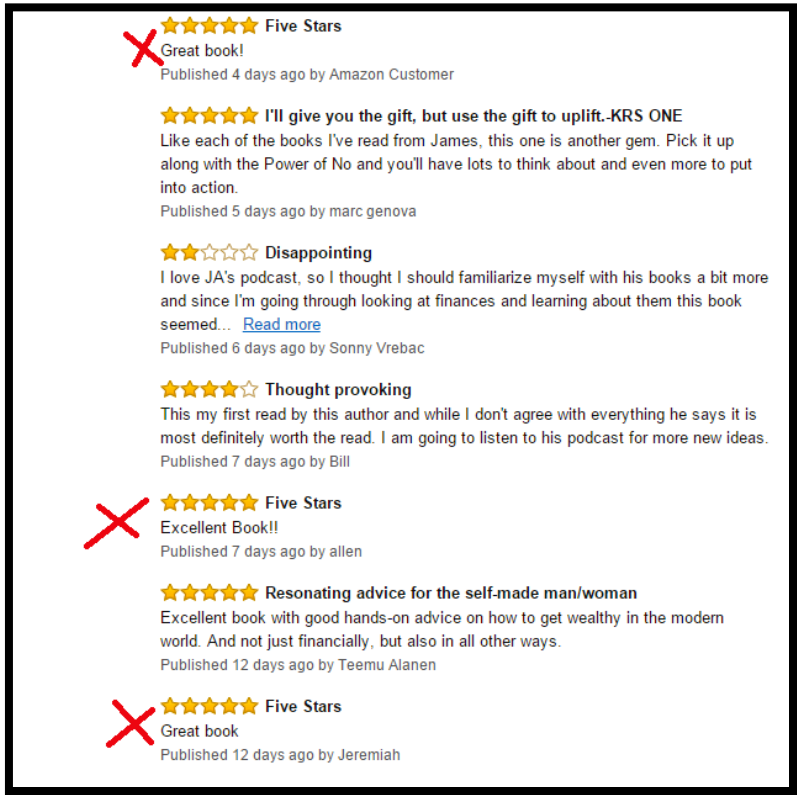
ACM SIGKDD' 20, August 23-27th, Virtual Event, CA, USA

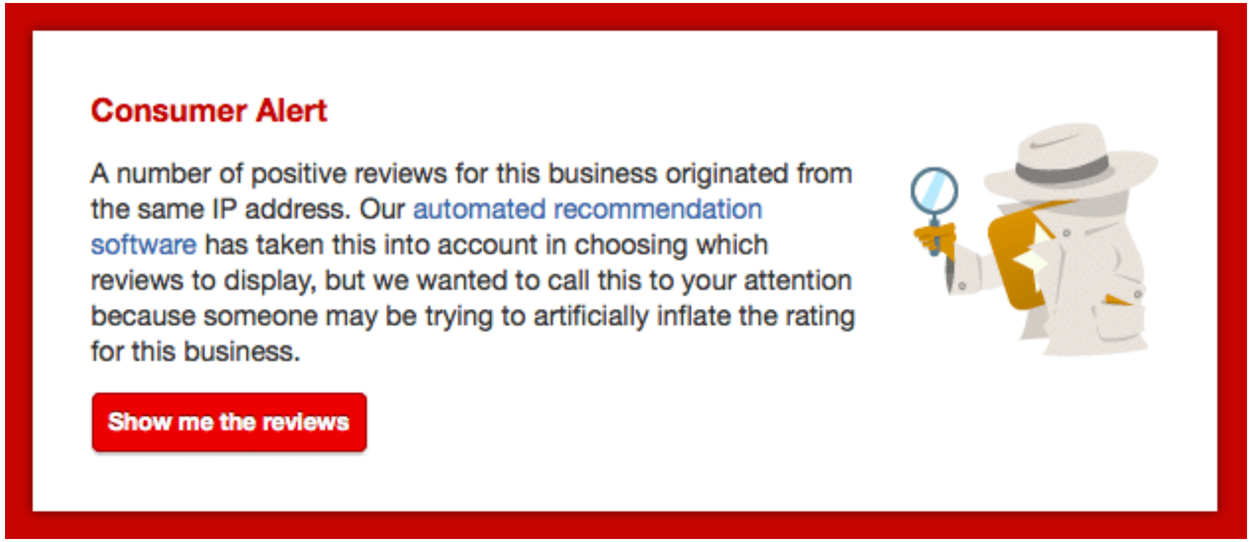
Outline

- **Background:** review spam and spamming campaign
- **Highlight:** previous works vs. our works
- **Methodology I:** practical goals of spammers and defenders
- **Methodology II:** robust training of spam detectors (Nash-Detect)
- **Experiments:** the training and deployment performance of Nash-Detect
- **Conclusion & Future Works**

Fake Reviews are Prevalent

- Near **40%** reviews in Amazon are fake[1]
- Yelp hide suspicious reviews and alert consumers



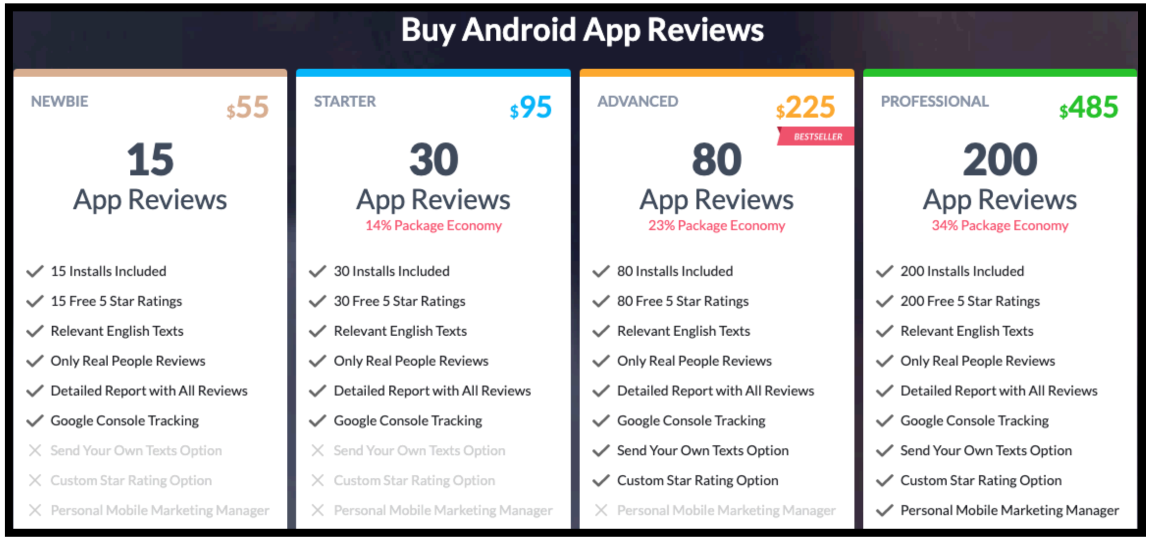


[1] J. Swearingen. 2017. Amazon Is Filled With Sketchy Reviews. Here's How to Spot Them. https://slct.al/2TBXDpT

Images from https://upserve.com/restaurant-insider/five-key-reasons-shouldnt-buy-yelp-reviews/ http://greyenlightenment.com/detecting-fake-amazon-reviews/

Spamming Campaign

- Dishonest merchants can **easily** buy high-quality fake reviews online
- Machine-generated fake reviews are very **authentic-like**[1]



Generated Reviews (Yelp)
I love this place ! I ’ve been here several times and I ’ve never been disappointed . The food is always fresh and delicious . The service is always friendly and attentive . I ’ve been here several times and have never been disappointed .
I ’ve been to this location twice now and both times I ’ve been very impressed . I ’ve tried their specialty pizzas and they ’re all really good . The only problem is that they ’re not open on sundays . They ’re not open on sundays .
I have been coming to this place for years and have always had great food and service . They have a great lunch buffet . They have a great selection of food for the price . They do have a lot of seating and I would recommend reservations .
I ’ve eaten here about 8 times . I ’ve been introduced to this place . Its always busy and their food is consistently great . I LOVE their food , hence the name . It is so clean , the staff is so friendly , and the food is great . I especially like the chicken pad thai , volcano roll , and the yellow curry .
this is strictly to go . Love , love , love the food ! we usually usually get brisket (oh my) , sandwich (pastrami , or pork , just so good) and now these are my two favorites . It ’s great . This is gone (according to our waitress) .

[1] P. Kaghazgaran, M. Alfifi, and J. Caverlee. 2019. Wide-Ranging Review Manipulation Attacks: Model, Empirical Study, and Countermeasures. In CIKM.

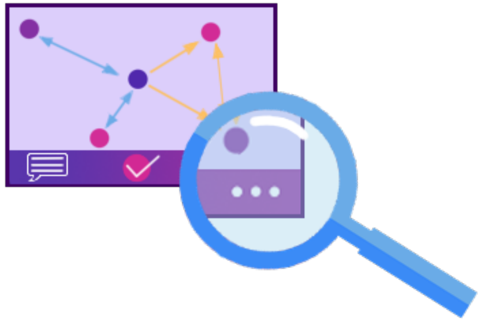
Images from https://mopeak.com/buy-android-reviews/
http://faculty.cs.tamu.edu/caverlee/pubs/kaghazgaran19cikm.pdf

Review Spam Detection

- To detect fake reviews, three major types of spam detectors have been proposed

Text-based Detectors

Behavior-based Detectors

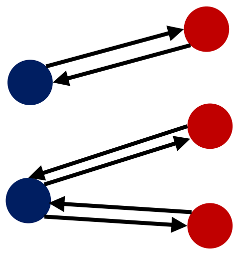


Graph-based Detectors

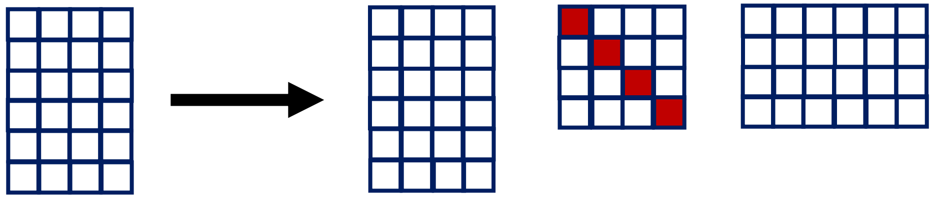
Base Spam Detectors

- **GANG**
- **SpEagle**

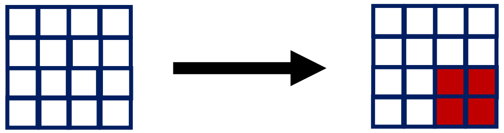
MRF-based detector



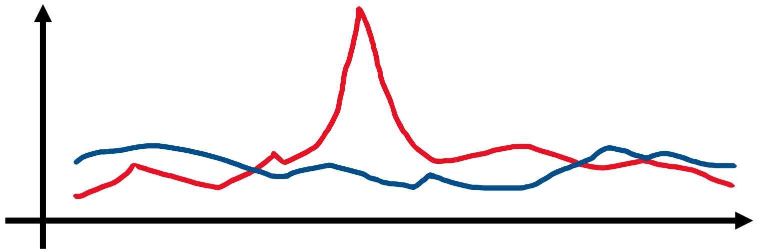
- **fBox** SVD-based detector



- **Fraudar** Dense-block-based detector



- **Prior** Behavior-based detector



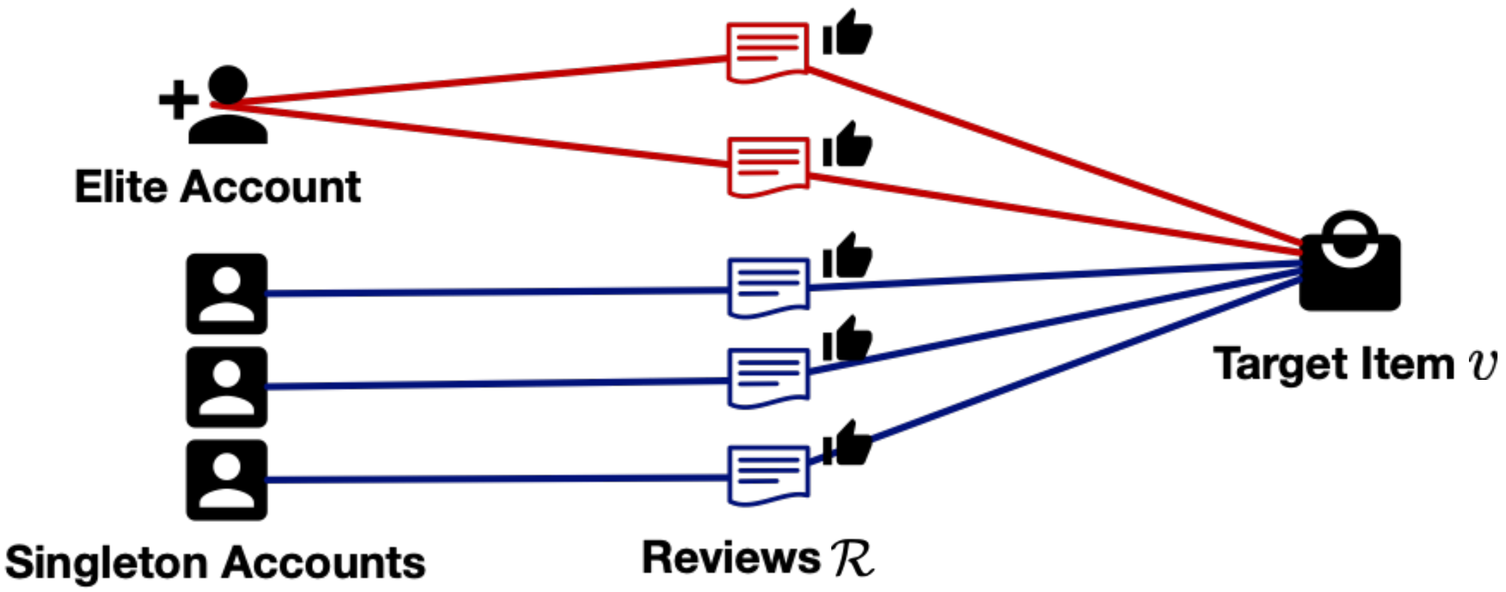
Previous Works vs. Our Work

- **Previous works:**
  - Static dataset
  - Accuracy-based evaluation metric
  - Fixed spamming pattern
  - Single detector
- **Our work:**
  - Dynamic game between spammer and defender
  - Practical evaluation metric
  - Evolving spamming strategies
  - Multiple detectors ensemble

Turning Reviews into Business Revenues

- In Yelp, product's rating is correlated to its revenue[1]

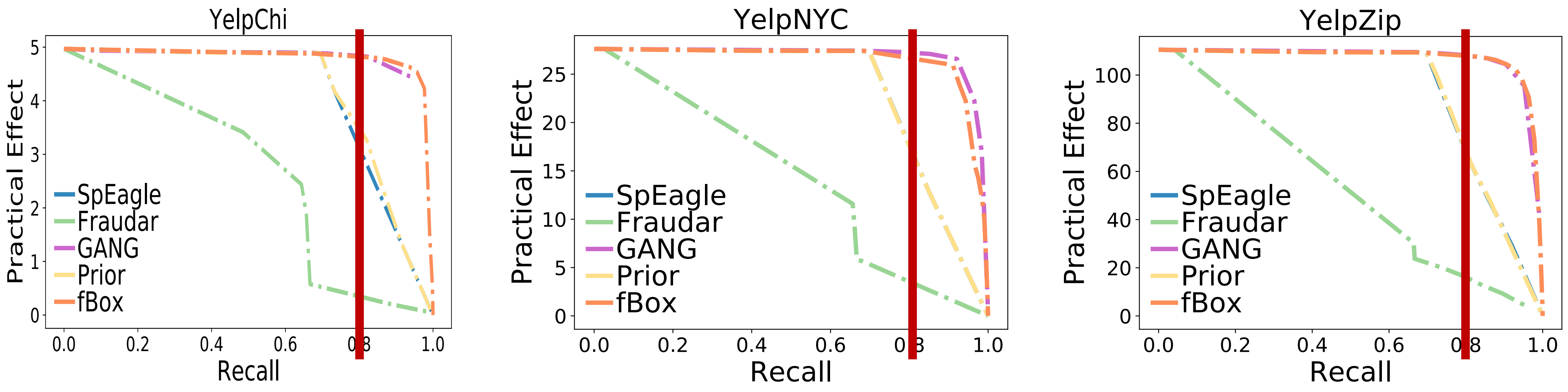
Revenue Estimation & Practical Effect: $f(v;\mathcal{R}) = \beta_0 \times \boxed{\mathrm{RI}(v;\mathcal{R})} + \beta_1 \times \boxed{\mathrm{ERI}(v;\mathcal{R}_E(v))} + \alpha$



[1] M. Luca. 2016. Reviews, reputation, and revenue: The case of Yelp. com. HBS Working Paper (2016).

Practical Effect is Better than Recall

- We run five detectors individually against five attacks
- When detector recalls are **high (>0.7)**, the practical effects are **not reduced**



Spammer’s Practical Goal

Spamming Practical Effect: $\mathrm{PE}(v;\mathcal{R},\mathsf{p},\mathsf{q}) = \underbrace{\boxed{f(v;\mathcal{R}(\mathsf{p},\mathsf{q}))}}_{\text{Revenue after attacks}} - \underbrace{\boxed{f(v;\mathcal{R})}}_{\text{Revenue before attacks}}$

Revenue after attacks

Revenue before attacks

- To promote a product, the practical goal of the spammer is to **maximize** the PE.

Spammer’s Goal: $\max_{\boxed{\mathsf{p}}} \quad \max\{0, \mathrm{PE}(v;\mathcal{R},\mathsf{p},\mathsf{q})\}$

Spamming strategy weights

Defender’s Practical Goal

- The defender needs to **minimize** the practical effect
- We combine detector prediction results with the practical effect to formulate a **cost-sensitive loss**

The cost of false negatives

Defender’s Goal: $$\min_{\boxed{\mathsf{q}}} \mathcal{L}_{\mathsf{q}} = \frac{1}{|\mathcal{R}(\mathsf{p},\mathsf{q})|} \sum_{r \text{ is FN}} \boxed{-C_{\mathrm{FN}}(v,r)} \boxed{\log P(y=1|r;\mathsf{q})}$$

Detector weights

The prediction results of detectors

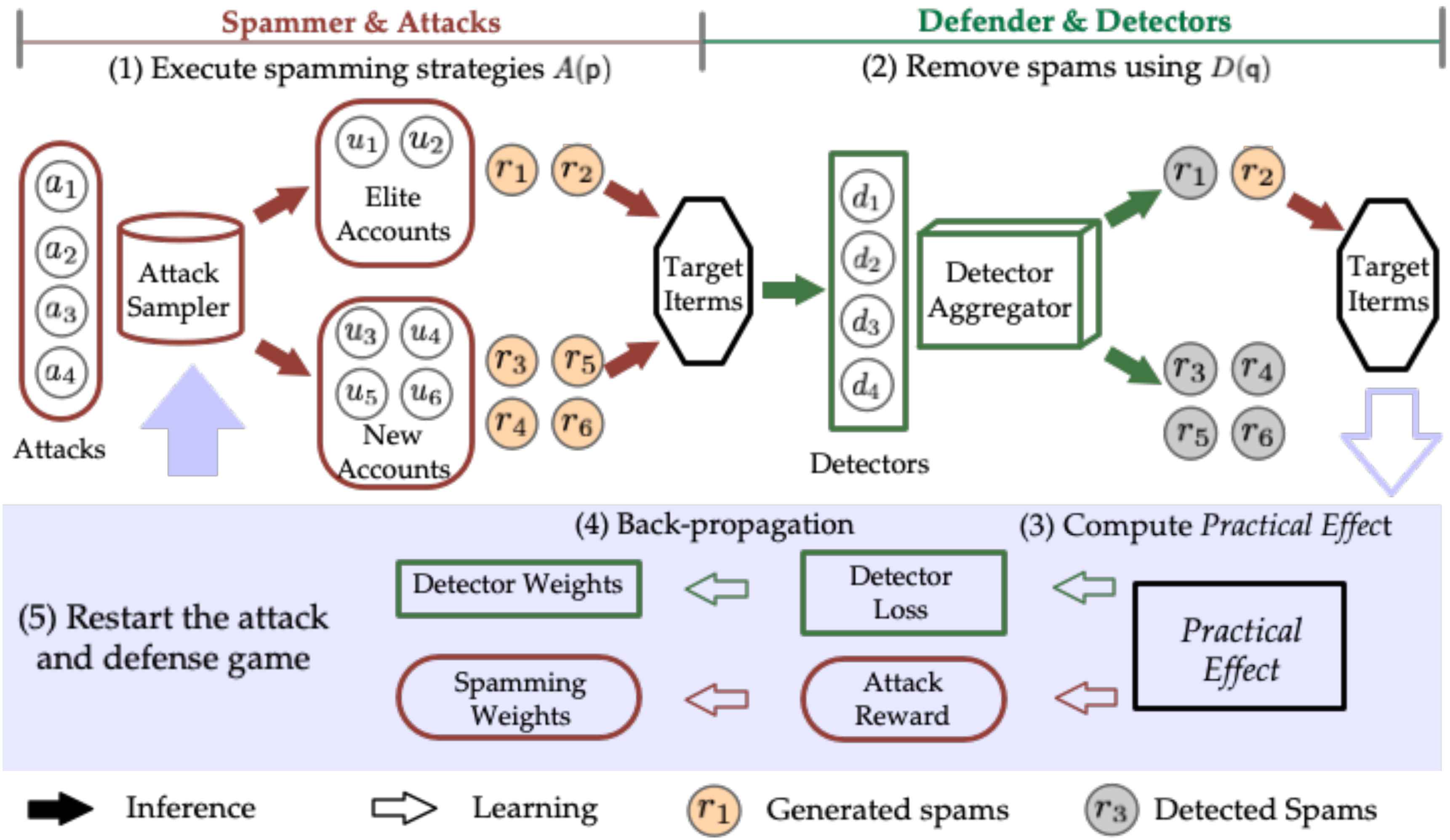
A Minimax-Game Formulation

Minimax Game Objective: $\min_{\mathsf{q}} \max_{\mathsf{p}} \sum_{v \in \mathcal{V}_T} \max\{0, \mathrm{PE}(v; \mathcal{R}, \mathsf{p}, \mathsf{q})\}$

- The objective function is not differentiable

- Our solution: **multi-agent non-cooperative reinforcement learning** and **SGD optimization**

Train a Robust Detector - Nash-Detect



Base Spamming Strategies

- **IncBP:** add reviews with minimum suspiciousness based on belief propagation on MRF
- **IncDS:** add reviews with minimum densities on graph composed of accounts, reviews, and products
- **IncPR:** add reviews with minimum prior suspicious scores computed by behavior features
- **Random:** randomly add reviews
- **Singleton:** add reviews with new accounts

Experimental Settings

- Dataset statistics and spamming attack settings

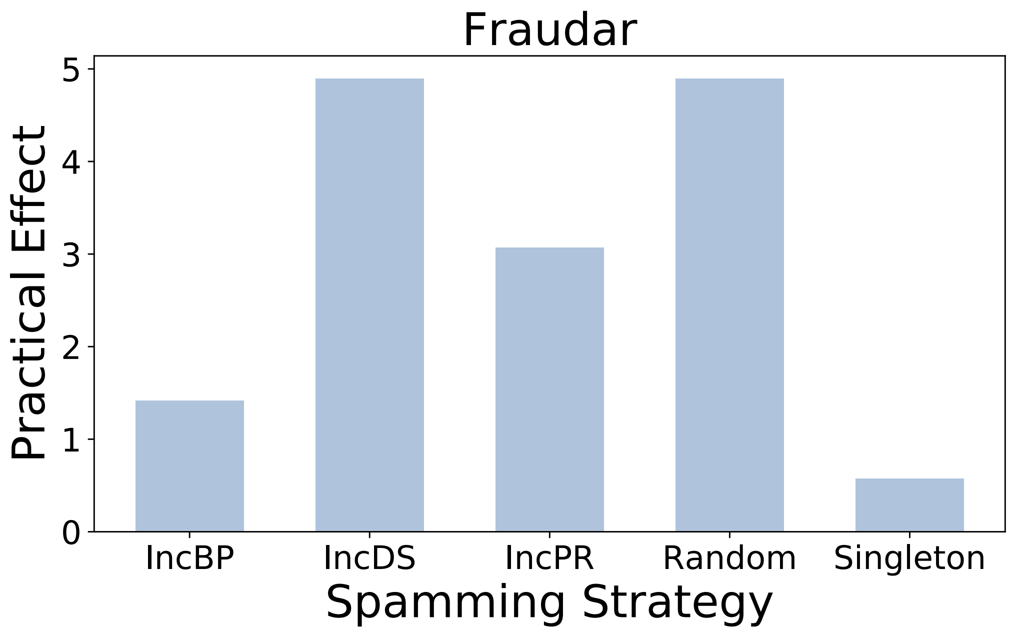
Dataset	# Accounts	# Products	# Reviews	# Controlled elite accounts	# Target products	# Posted fake reviews
YelpChi	38063	201	67395	100	30	450
YelpNYC	160225	923	359052	400	120	1800
YelpZip	260277	5044	608598	700	600	9000

- The spammer controls **elite and new accounts**

- The defender removes **top k** suspicious reviews

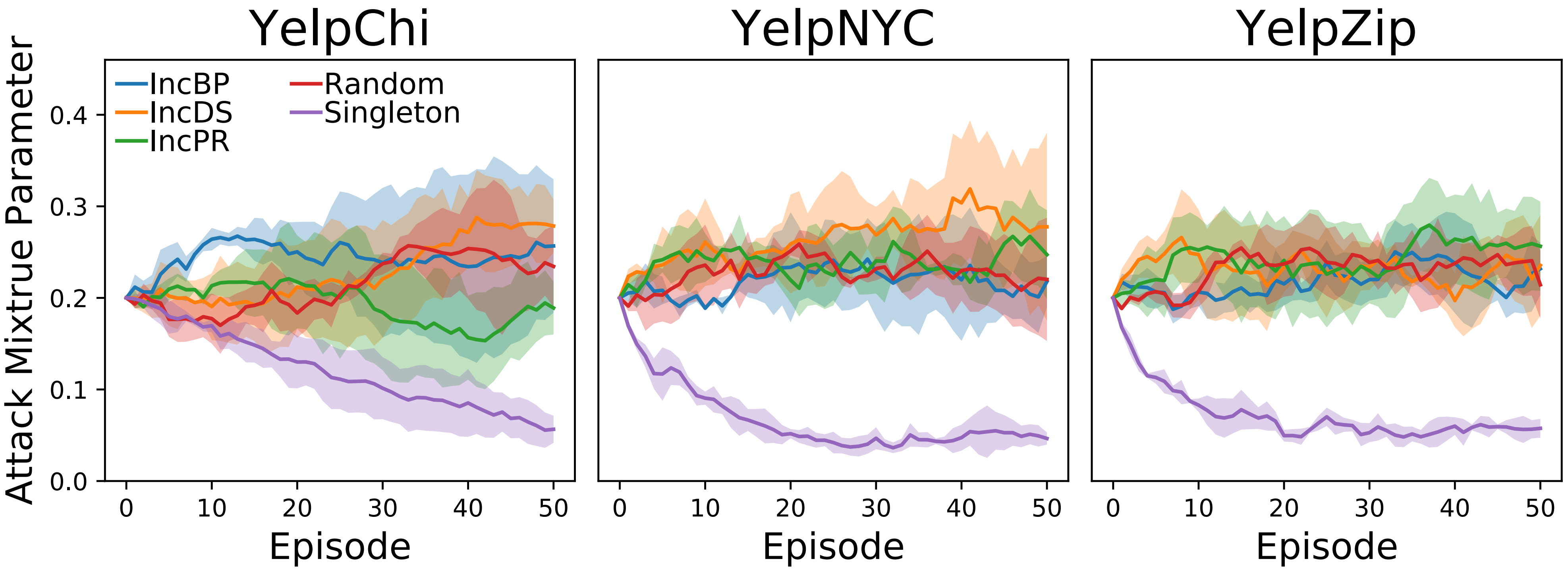
Fixed Detector's Vulnerability

- For a fixed detector (**Fraudar**), the spammer can switch to the spamming strategy with the max practical effect (**IncDS**)



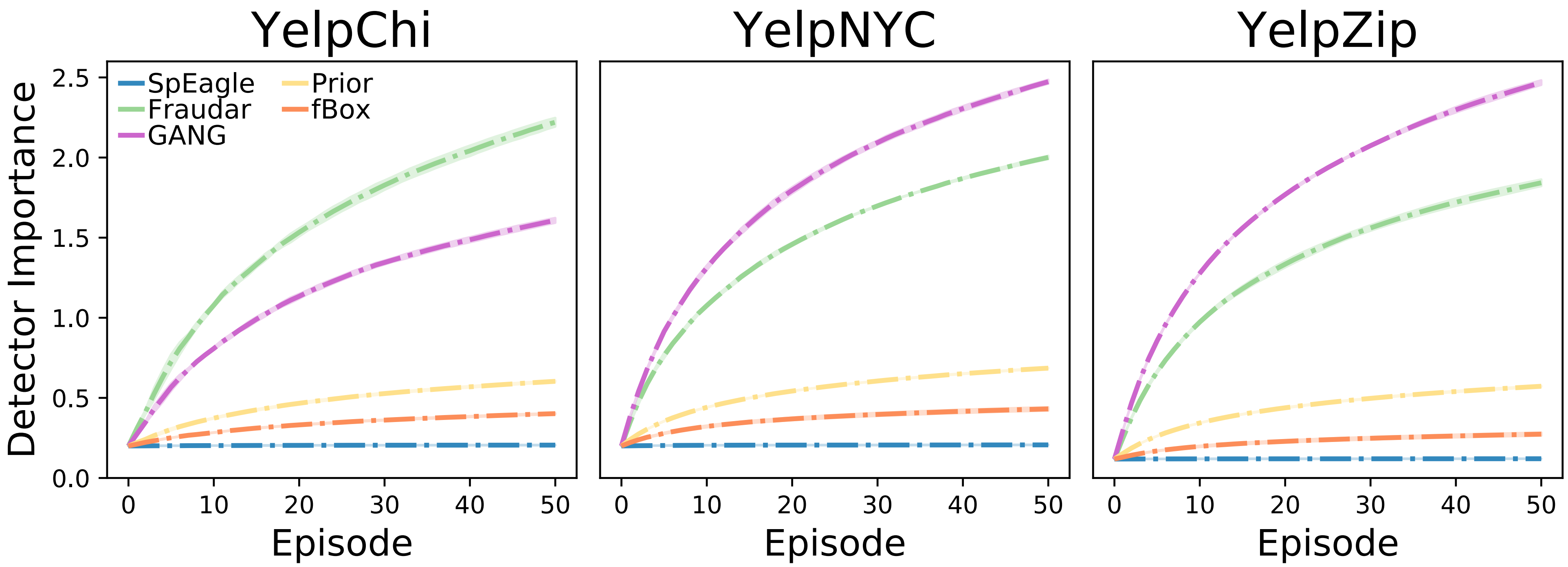
Nash-Detect Training Process

- **Singleton** attack is less effective than other four attacks.



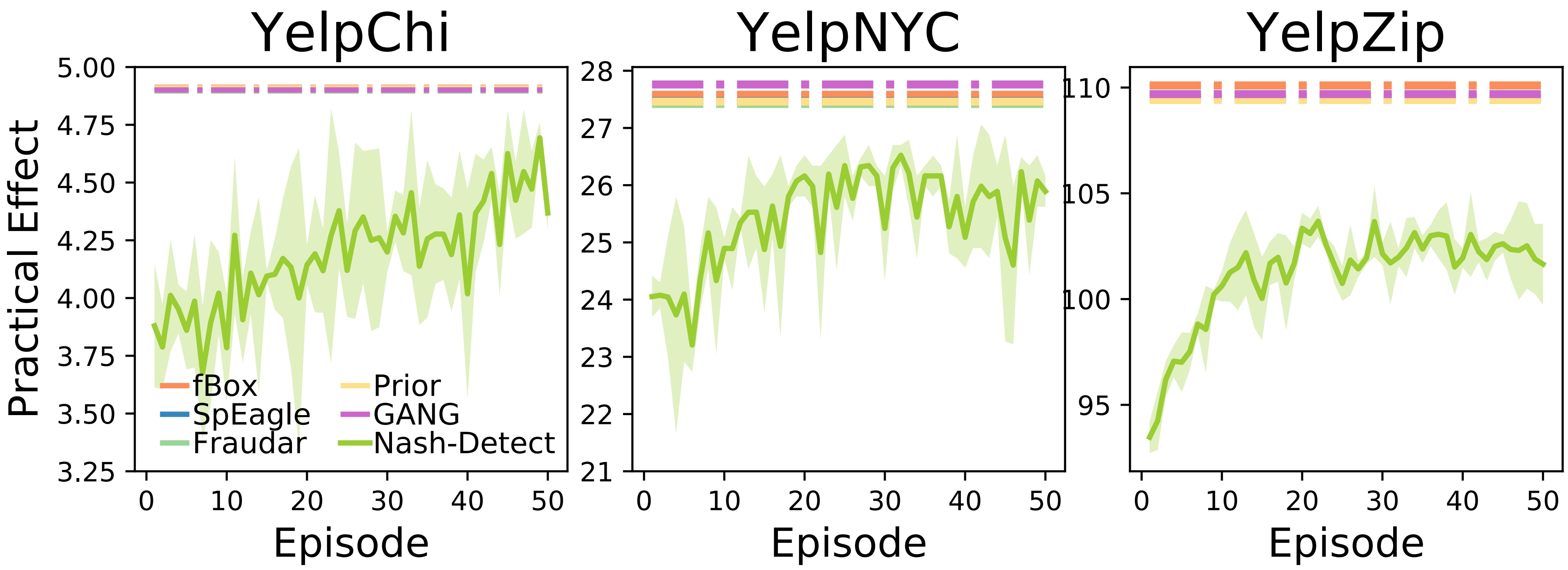
Nash-Detect Training Process

- Nash-Detect can find the optimal detector importance smoothly

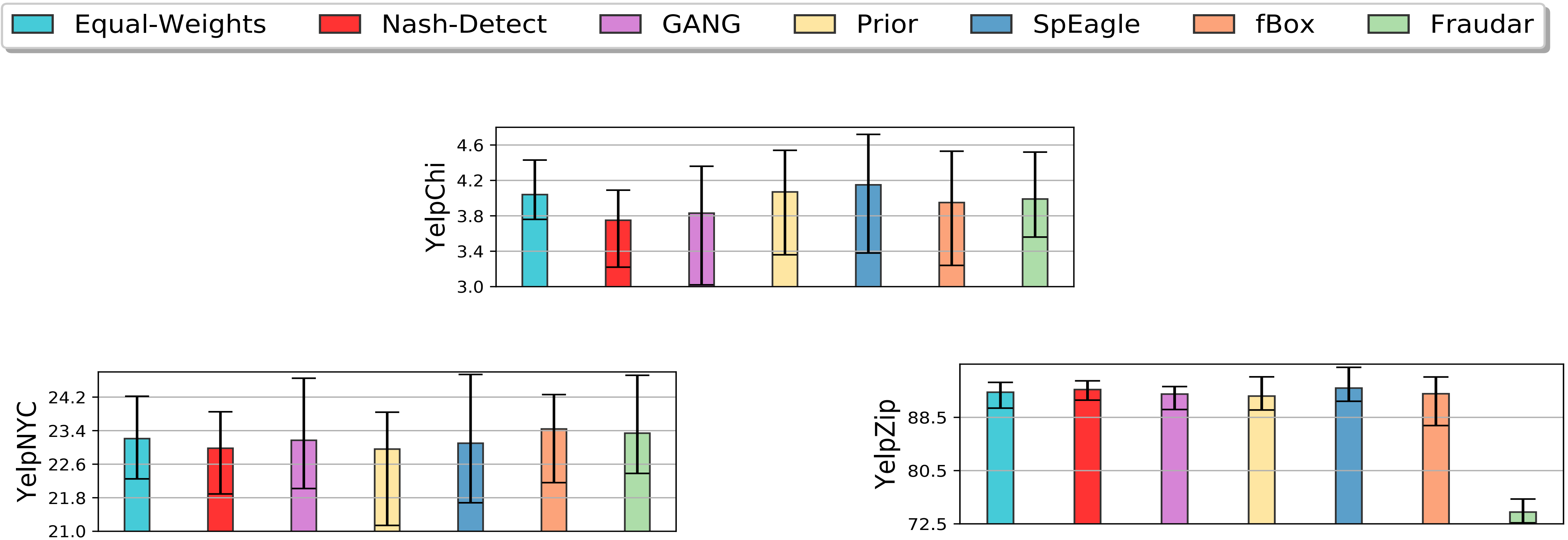


Nash-Detect Training Process

- The practical effect of detectors configured by Nash-Detect are always **less than** the worst-case performances



Nash-Detect Performance in Deployment



Key Takeaways

- New metric
- New spamming strategies
- New adversarial training algorithm

Future Works

- Investigate the attack and defenses of deep learning spam detection methods
- Apply the Nash-Detect framework on other review systems and applications
- Develop advanced attack generation techniques aware of the states of review system

SafeGraph (https://github.com/safe-graph)

- **DGFraud**: a GNN-based fraud detection toolbox
  - 178 stars, ten GNN models
- **UGFraud**: an unsupervised graph-based fraud detection toolbox
  - Just released, six classic models, deployed on Pypi
- Graph-based Fraud Detection Paper List
  - 177 stars, more than 40 papers listed
- Graph Adversarial Learning Paper List
  - 238 stars, more than 110 papers listed

Robust Spammer Detection by Nash Reinforcement Learning

Yingtong Dou (UIC)   Guixiang Ma (Intel Labs)

Philip S. Yu (UIC)   Sihong Xie (Lehigh)

ydou5@uic.edu

Paper: http://arxiv.org/abs/2006.06069
Slides: http://ytongdou.com/files/kdd20slides.pdf
Code: https://github.com/YingtongDou/Nash-Detect

ACM SIGKDD' 20, August 23-27th, Virtual Event, CA, USA