KDD-2016 Tutorial, Aug 13, 2016, San Francisco, CA, USA

Lifelong Machine Learning and Computer Reading the Web

Zhiyuan (Brett) Chen, Google

Estevam Hruschka, UFSCar, CMU

Bing Liu, University of Illinois at Chicago

Introduction

- Classic Machine Learning (ML) paradigm (isolated single-task learning)
  - Given a dataset, run a ML algo. to build a model
  - Without considering the past learned knowledge
- Existing ML algorithms such as
  - SVM, NB, DT, Deep NN, CRF, and topic models
  - Have been very successful in practice
- Let’s call this: Machine Learning (ML) 1.0
  - Isolated learning has limitations.

Introduction: ML 1.0 limitation

- Learned knowledge is not cumulative
- No memory: Knowledge learned isn’t retained
  - ML cannot learn by leveraging the past knowledge
- Due to the lack of prior knowledge
  - ML needs a large number of training examples.
- Without knowledge accumulation and self-learning (with no supervision)
  - It is impossible to build a truly intelligent system
    - Cannot imagine that for every task a large number of training examples need to be labeled by humans

Introduction: human learning

- Humans never learn in isolation
- We learn effectively from a few examples with the help of the past knowledge.
    - Nobody has ever given me 1000 positive and 1000 negative docs, and asked me to build a classifier manually
- Whenever we see a new situation, a large part of it is known to us. Little is completely new!

Introduction: ML 2.0

(Thrun, 1996b; Silver et al 2013; Chen and Liu, 2014a)

- *Lifelong Machine Learning* (LML)
  - Learn as humans do
  - Retain learned knowledge from previous tasks & use it to help future learning
- Let us call this paradigm Machine Learning 2.0
  - LML may require a systems approach
  - Multiple tasks with multiple learning/mining algorithms

Introduction: LML with Big Data

- Big data provides a great opportunity for LML
  - Abundant information from the Web
  - Extensive sharing of concepts across tasks/domains
  - Example: natural language learning tasks on different sources are all related

Outline

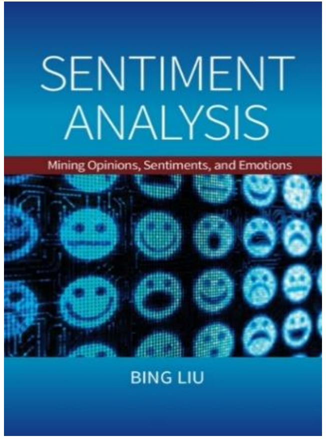
- A motivating example
- What is lifelong machine learning?
- Related learning tasks
- Lifelong supervised learning
- Semi-supervised never-ending learning
- Lifelong unsupervised learning
- Lifelong reinforcement learning
- Summary

Outline

- A motivating example
- What is lifelong machine learning?
- Related learning tasks
- Lifelong supervised learning
- Semi-supervised never-ending learning
- Lifelong unsupervised learning
- Lifelong reinforcement learning
- Summary

A Motivating Example

(Liu, 2012; 2015)



- Sentiment analysis or opinion mining
  - Computational study of opinion, sentiment, appraisal, evaluation, attitude, and emotion

- Active research area in NLP with unlimited applications
  - Useful to every organization and individual
  - Example: online shopping

A Motivating Example

(Liu, 2012; 2015)

- Sentiment analysis is suitable for LML
  - Extensive knowledge sharing across tasks/domains
  - Sentiment expressions, e.g., good, bad, expensive, great
  - Sentiment targets, e.g., "*The screen is great but the battery dies fast*."

(1) Sentiment Classification

- "*I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is great too. ....*"
- Goal: classify docs or sentences as + or -
  - Need to manually label a lot of training data for each domain, which is highly labor-intensive
  - Can we not label for every domain or at least not so many docs/sentences?

Exploiting the Past Information

- It is “well-known” that a sentiment classifier (SC) built for domain A will not work for domain B
    - E.g., SC built for “camera” will not work for “earphone”
- Classic solution: transfer learning
    - Using labeled data in the source domain (camera) to help learning in the target domain (earphone)
    - Two domains need to be very similar
- This may not be the best solution!

Lifelong Sentiment Classification

(Chen, Ma and Liu 2015)

Imagining - we have worked on a *large number of past domains/tasks* with their training data *D*

- Do we need any data from a new domain *T*?

- No in many cases – A naive "*LML*" method by polling all data together works wonders.
  - Can improve accuracy by as much as 19% (= 80%-61%)
  - Why? Sharing of sentiment expressions
- Yes in other cases: e.g., we build a SC using *D*, but it works poorly for toy reviews.
  - Why? Because of the word "toy"

(2) Lifelong Aspect Extraction

(Chen and Liu, 2014a, 2014b)

- "*The battery life is long, but pictures are poor*."
  - Aspects (opinion targets): battery life, picture
- Observation:
  - A fair amount of aspect overlapping across reviews of different products or domains
    - Every product review domain has the aspect *price*
    - Most electronic products share the aspect *battery*
    - Many also share the aspect of *screen*.
  - It is rather "silly" not to exploit such sharing in learning or extraction.

Outline

- A motivating example
- What is lifelong machine learning?
- Related learning tasks
- Lifelong supervised learning
- Semi-supervised never-ending learning
- Lifelong unsupervised learning
- Lifelong reinforcement learning
- Summary

Lifelong Machine Learning (LML)

(Thrun 1995, Chen and Liu 2014, 2015)

Definition: LML is a continuous learning process where the learner has performed a sequence of *N* learning tasks, T_1, T_2, …, T_N.

- When faced with the *N*th task T_{N+1} with its data D_{N+1}, the learner makes use of the prior knowledge *K* in its knowledge base (KB) to help learn T_{N+1}.
- KB contains all the knowledge accumulated in the past learning of the *N* tasks.
- After learning T_{N+1}, KB is updated with the learned (intermediate as well the final) results from T_{N+1}.

Key Characteristics of LML

- Continuous learning process
- Knowledge accumulation in KB
- Use of past knowledge to help future learning

Components of LML

- Knowledge Base (KB)
  - Past Information Store (PIS)
  - Knowledge Store (KS)
  - Knowledge Miner (KM)
  - Knowledge Reasoner (KR)

- Knowledge-Based Learner (KBL)

Past Information Store (PIS)

- It stores the information from the past learning. It may have sub-stores for storing information such as
  - The original data used in each past task
  - The intermediate results from the learning of each past task
  - The final model or patterns learned from each past task
  - etc.

Knowledge Store (KS)

- It stores the knowledge mined/consolidated from PIS (Past Information Store).
  - Meta-knowledge discovered from PIS, e.g., general/shared knowledge applicable to multiple domains/tasks
    - E.g., a list of words commonly used to represent positive or negative sentiment
  - This requires a general knowledge representation scheme suitable for a class of applications

Knowledge Miner (KM)

- It mines (meta) knowledge from PIS (Past Information Store)
- This mining is regarded as a meta-mining process because it learns knowledge from information resulted from learning of the past tasks
- The resulting knowledge is stored to KS (Knowledge Store)

Knowledge Reasoner (KR)

- It makes inference in the KB to generate additional knowledge.
- Most current LML systems do not have this capability.
- However, with the advance of LML, this component will become important.

Knowledge-Based Learner (KBL)

- Given the knowledge in KS, the LML learner can leverage the knowledge and possibly some information in PIS to learn from the new task, which should
  - Learn better even with a large amount of training data
  - Learn well with a small amount of data
  - …

LML: Flexible Learning

- It can use any past knowledge or information in any way to help the new task learning.
- It can focus on learning the (*N*+1)th task by using knowledge gained from the past *N* tasks.
- It can also improve any of the models from the past *N* tasks based on results from the other *N* tasks (including the (*N*+1)th task):
  - By treating that previous task as the "(*N*+1)th" task.

Outline

- A motivating example
- What is lifelong machine learning?
- Related learning tasks
- Lifelong supervised learning
- Semi-supervised never-ending learning
- Lifelong unsupervised learning
- Lifelong reinforcement learning
- Summary

Transfer learning

- Source domain(s): With labeled training data
- Target domain: With little/no labeled training data
- Goal: leverage the information from the source domain(s) to help learning in the target domain
    - Only optimize the target domain/task learning

A Large Body of Literature

- Transfer learning has been a popular research topic and researched in many fields, e.g.,
  - Machine learning
  - Data mining
  - Natural language processing
  - Computer vision
- Pan & Yang (2010) presented an excellent survey with extensive references.

One Transfer Learning Technique

- Structural correspondence learning (SCL) (Blitzer et al 2006)
- Pivot features
  - Have the same characteristics or behaviors in both domains
  - Non-pivot features which are correlated with many of the same pivot features are assumed to correspond

Choosing Pivot Features

- For different applications, pivot features may be chosen differently, for example,
  - For part-of-speech tagging, frequently-occurring words in both domains are good choices (Blitzer et al., 2006)
  - For sentiment classification, pivot features are words that frequently-occur in both domains and also have high mutual information with the source label (Blitzer et al., 2007).

Finding Feature Correspondence

- Compute the correlations of each pivot feature with non-pivot features in both domains by building binary pivot predictors

$$f_\ell(\mathbf{x}) = \mathrm{sgn}(\hat{\mathbf{w}}_\ell \cdot \mathbf{x}), \quad \ell = 1 \ldots m$$

  - Using unlabeled data (predicting whether the pivot feature *l* occurs in the instance)
  - The weight vector $\hat{\mathbf{w}}_\ell$ encodes the covariance of the non-pivot features with the pivot feature

Finding Feature Correspondence

- Positive values in $\hat{\mathbf{w}}_\ell$:
  - Indicate that those non-pivot features are positively correlated with the pivot feature *l* in the source or the target

- Produce a correlation matrix W

$$W = [\hat{\mathbf{w}}_1 | \ldots | \hat{\mathbf{w}}_m]$$

Computing Low Dim. Approximation

- SVD is employed to compute a low-dimensional linear approximation θ

$$W = UDV^T \quad \theta = U^T_{[1:h,:]}$$

- θ : mapping from original space to new space

- The final set of features used for training and for testing: original features **x** + θ**x**

Multi-task learning

- Problem statement: Co-learn multiple related tasks simultaneously:
  - All tasks have labeled data and are treated equally
  - Goal: optimize learning/performance across all tasks through shared knowledge
- Rationale: introduce inductive bias in the joint hypothesis space of all tasks (Caruana, 1997)
  - By exploiting the task relatedness structure, or shared knowledge

One multi-task model: GO-MTL

(Kumar et al., ICML 2012)

- GO-MTL: Grouping and Overlap in Multi-Task Learning
- Does not assume that all tasks are related
- Applicable to classification and regression

GO-MTL assumptions

- All task models share latent basic model components
- Each task model is a linear combination of shared latent components
- The linear weight is sparse, to use few latent components

Notations

- N tasks in total
- k (< N) latent basis model components
- Each basis task is represented by a $\boldsymbol{l}$ (a vector of size d)
- For all latent tasks, $\boldsymbol{L} = (\boldsymbol{l}_{1,} \boldsymbol{l}_{2,} \ldots, \boldsymbol{l}_k)$
- $\boldsymbol{L}$ is learned from N individual tasks.
  - E.g., weights/parameters of logistic regression or linear regression

The Approach

- **s**^{t} is a linear weight vector and is assumed to be sparse.

$$\boldsymbol{\theta}^t = \mathbf{L}\mathbf{s}^t$$

- Stacking **s**^{t} for all tasks, we get **S**. **S** captures the task grouping structure.

$$\underset{d\times N}{\boldsymbol{\theta}} = \underset{d\times k}{\mathbf{L}} \times \underset{k\times N}{\mathbf{S}}$$

Objective Function in GO-MTL

$$\sum_{t=1}^{N}\sum_{i=1}^{n_t}\mathcal{L}\left(f(\mathbf{x}_i^t;\mathbf{L}\mathbf{s}^t),y_i^t\right)+\mu\left\|\mathbf{S}\right\|_1+\lambda\left\|\mathbf{L}\right\|_F^2$$

Optimization Strategy

- Alternating optimization strategy to reach a local minimum.
- For a fixed ***L***, optimize *s*$_t$:

$$\mathbf{s}^t = \underset{\mathbf{s}}{\operatorname{argmin}} \sum_{i=1}^{n_t} \mathcal{L}\left(f(\mathbf{x}_i^t; \mathbf{Ls}), y_i^t\right) + \mu \left\|\mathbf{s}\right\|_1$$

- For a fixed ***S***, optimize ***L***:

$$\underset{\mathbf{L}}{\operatorname{argmin}} \sum_{t=1}^{N} \sum_{i=1}^{n_t} \mathcal{L}\left(f(\mathbf{x}_i^t; \mathbf{Ls}^t), y_i^t\right) + \lambda \left\|\mathbf{L}\right\|_F^2$$

A Large Body of Literature

- Two tutorials on MTL
  - Multi-Task Learning: Theory, Algorithms, and Applications. SDM-2012, by Jiayu Zhou, Jianhui Chen, Jieping Ye
  - Multi-Task Learning Primer. IJCNN'15, by Cong Li and Georgios C. Anagnostopoulos

Transfer, Multitask vs. Lifelong

- Transfer learning vs. LML
  - Transfer learning is not continuous
  - The source must be very similar to the target
  - No retention or accumulation of knowledge
  - Only one directional: help target domain
- Multitask learning vs. LML
  - Multitask learning retains no knowledge except data
  - Hard to re-learn all when tasks are numerous
- Incremental (online) multi-task learning is LML

Online Learning

- The training data points come in a sequential order (online setting)
  - Computationally infeasible to train over the entire dataset
- Different from LML
  - Still performs the same learning task over time
  - LML aims to learn from a sequence of different tasks, retain and accumulate knowledge

Outline

- A motivating example
- What is lifelong machine learning?
- Related learning tasks
- Lifelong supervised learning
- Semi-supervised never-ending learning
- Lifelong unsupervised learning
- Lifelong reinforcement learning
- Summary

Lifelong Supervised Learning (LSL)

- The learner has performed learning on a sequence of supervised learning tasks, from 1 to *N*.
- When faced with the (*N*+1)th task, it uses the relevant knowledge and labeled training data of the (*N*+1)th task to help learning for the (*N*+1)th task.

Early Work on Lifelong Learning (Thrun, 1996b)

- Concept learning tasks: The functions are learned over the lifetime of the learner, f_1, f_2, f_3, … $\in F$.
- Each task: learn the function $f: I \rightarrow \{0, 1\}$. $f(x)=1$ means x is a particular concept.
  - For example, $f_{dog}(x)=1$ means x is a dog.
- For nth task, we have its training data X
  - Also the training data X_k of k =1 , 2, …, n-1 tasks.

Intuition

- The paper proposed a few approaches based on two learning algorithms,
  - Memory-based, e.g., kNN or shepard's method
  - Neural networks,
- Intuition: when we learn $f_{dog}(x)$, we can use functions or knowledge learned from previous tasks, such as $f_{cat}(x)$, $f_{bird}(x)$, $f_{tree}(x)$, etc.
  - Data for $\mathrm{f}_{\mathrm{cat}}(\mathrm{X})$, $\mathrm{f}_{\mathrm{bird}}(\mathrm{X})$, $\mathrm{f}_{\mathrm{tree}}(\mathrm{X})$… are support sets.

Memory based Lifelong Learning

- First method: use the support sets to learn a new representation, or function

  g: *I* → *I'*

  - which maps input vectors to a new space. The new space is the input space for the final *k*NN.
  - Adjust *g* to minimize the energy function.

$$E := \sum_{k=1}^{n-1} \sum_{\langle x,y=1\rangle \in X_k} \left(\sum_{\langle x',y'=1\rangle \in X_k} ||g(x)-g(x')|| - \sum_{\langle x',y'=0\rangle \in X_k} ||g(x)-g(x')|| \right)$$

  - g is a neural network, trained with Back-Prop. kNN is then applied for the *n*th (new) task

Second Method

- It learns a distance function using support sets

  d: $I \times I \rightarrow [0, 1]$

  - It takes two input vectors x and x' from a pair of examples <x, y>, <x', y'> of the same support set X_k (k = 1, 2, , …, n-1)
  - d is trained with neural network using back-prop, and used as a general distance function
  - Training examples are:

$$\langle (x, x'), 1 \rangle \quad \text{if } y{=}y'{=}1$$

$$\langle (x, x'), 0 \rangle \quad \text{if } (y{=}1 \wedge y'{=}0) \text{ or } (y{=}0 \wedge y'{=}1)$$

Making Decision

- Given the new task training set X_n and a test vector x, for each +ve example, (x', y'=1)$\in X_n$,
  - d(x, x') is the probability that x is a member of the target concept.
- Decision is made by using votes from positive examples, <x_1, 1>, <x_2, 1>, … $\in X_n$ combined with Bayes' rule

$$P(f_n(x)=1)=1-\left(1+\prod_{\langle x',y'=1\rangle\in X_n}\frac{d(x,x')}{1-d(x,x')}\right)^{-1}$$

LML Components in this case

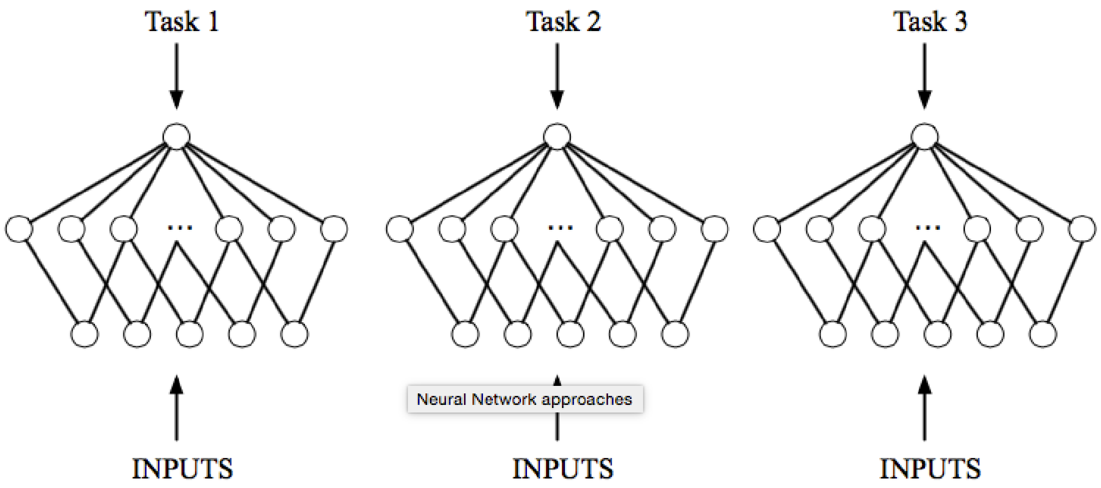
- **KB**
  - **PIS**: store all the support sets.
  - **KS:** Distance function *d*(x, x’): the probability of example x and x’ being the same concept.
    - Past knowledge is re-learned whenever a new task arrives.
  - **KM**: Neural network with Back-Propagation.
- **KBL**: The decision making procedure in the last slide.

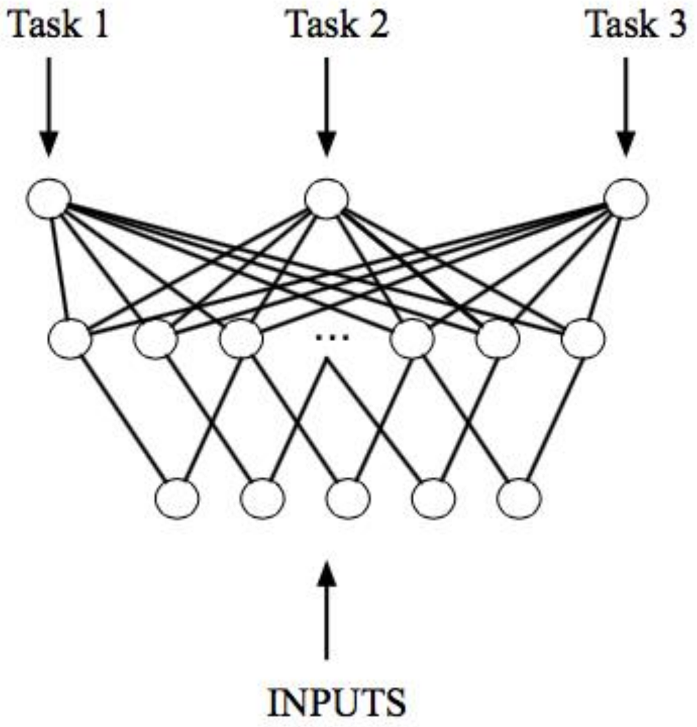
Neural Network approaches

- Approach 1: based on that in (Caruana, 1993, 1997), which is actually a batch multitask learning approach.
  - simultaneously minimize the error on both the support sets $\{X_k\}$ and the training set X_n

- Approach 2: an *explanation-based neural network (EBNN)*

Neural Network approaches





Results

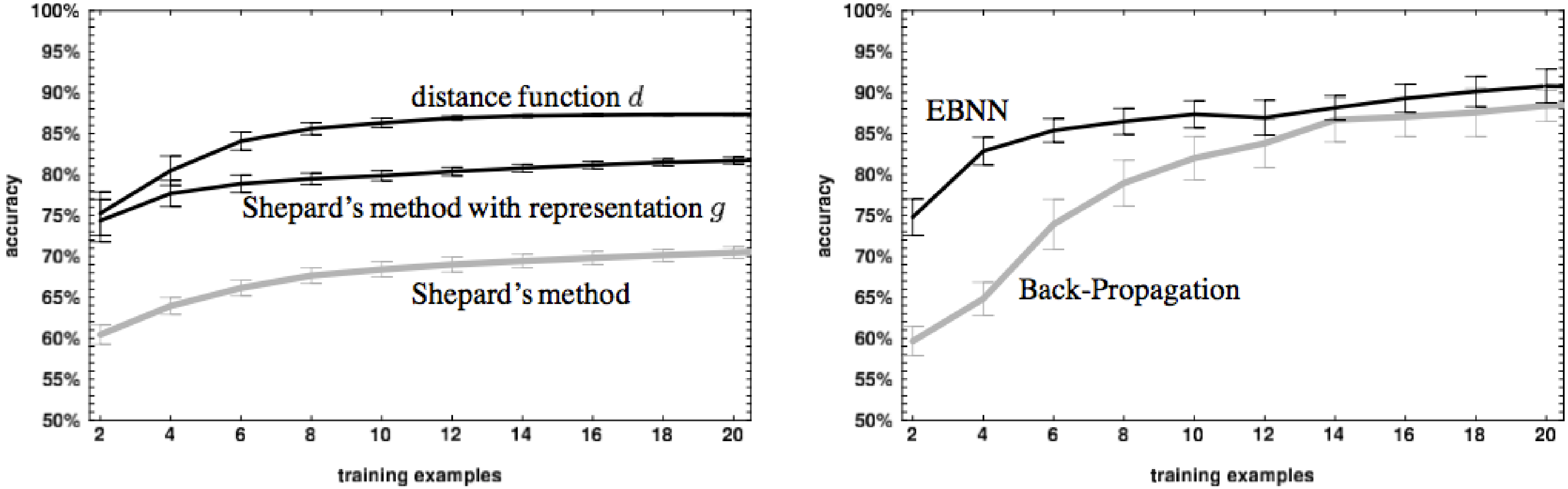


Figure 2: Generalization accuracy as a function of training examples, measured on an independent test set and averaged over 100 experiments. 95%-confidence bars are also displayed.

Task Clustering (TC)

(Thrun and O'Sullivan, 1996)

- In general, not all previous *N*-1 tasks are similar to the *N*th (new) task
- Based on a similar idea to the lifelong memory-based methods in (Thrun, 1996b)
  - It clusters previous tasks into groups or clusters
- When the (new) *N*th task arrives, it first
  - selects the most similar cluster and then
  - uses the distance function of the cluster for classification in the *N*th task

Some Other Early works on LML

- Constructive inductive learning to deal with learning problem when the original representation space is inadequate for the problem at hand (Michalski, 1993)
- Incremental learning primed on a small, incomplete set of primitive concepts (Solomonoff, 1989)
- Explanation-based neural networks MTL (Thrun, 1996a)
- MTL method of functional (parallel) transfer (Silver & Mercer, 1996)
- Lifelong reinforcement learning (Tanaka & Yamamura, 1997)
- Collaborative interface agents (Metral & Maes, 1998)

ELLA

(Ruvolo & Eaton, 2013a)

- ELLA: Efficient Lifelong Learning Algorithm
- It is based on GO-MTL (Kumar et al., 2012)
  - A batch multitask learning method
- ELLA is online multitask learning method
  - ELLA is more efficient and can handle a large number of tasks
  - Becomes a lifelong learning method
    - The model for a new task can be added efficiently.
    - The model for each past task can be updated rapidly.

Inefficiency of GO-MTL

- Since GO-MTL is a batch multitask learning method, the optimization goes through all tasks and their training instances (Kumar et al., 2012).

$$\sum_{t=1}^{T} \sum_{i=1}^{n_t} \mathcal{L}\left(f(\boldsymbol{x}_i^{(t)}; \boldsymbol{L}\boldsymbol{s}^{(t)}), y_i^{(t)} \right) + \mu \|\boldsymbol{S}\|_1 + \lambda \|\boldsymbol{L}\|_F^2$$

- Very inefficient and impractical for a large number of tasks.
  - It cannot incrementally add a new task efficiently

Initial Objective Function of ELLA

- Objective Function (**Average** rather than sum)

$$e_T\left(\mathbf{L}\right) = \frac{1}{T}\sum_{t=1}^{T}\min_{\mathbf{s}^{(t)}}\left\{\frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}\left(f\left(\mathbf{x}_i^{(t)};\mathbf{L}\mathbf{s}^{(t)}\right),y_i^{(t)}\right)\right.$$
$$\left. + \mu\|\mathbf{s}^{(t)}\|_1\right\} + \lambda\|\mathbf{L}\|_{\mathsf{F}}^2 \ , \qquad (1)$$

Approximate Equation (1)

- Eliminate the dependence on all of the past training data through inner summation
  - By using the second-order Taylor expansion of around $\theta = \theta^{(t)}$ where
  - $\theta^{(t)}$ is an optimal predictor learned on only the training data on task *t*.

Taylor Expansion

- One variable function

$$g(x) \approx g(a) + g'(a)(x-a) + \frac{1}{2}g''(a)(x-a)^2$$

- Multivariate function

$$g(\mathbf{x}) \approx g(\mathbf{a}) + \nabla g(\mathbf{a})(\mathbf{x}-\mathbf{a}) + \frac{1}{2}\|(\mathbf{x}-\mathbf{a})\|^2_{\boldsymbol{H}(\mathbf{a})}$$

Removing inner summation

$$\frac{1}{N}\sum_{t=1}^{N}\min_{\mathbf{s}^t}\left\{\|\hat{\boldsymbol{\theta}}^t - \mathbf{L}\mathbf{s}^t\|^2_{\boldsymbol{H}^t} + \mu\|\mathbf{s}^t\|_1\right\} + \lambda\|\mathbf{L}\|_F^2$$

$$\boldsymbol{H}^t = \frac{1}{2}\nabla^2_{\boldsymbol{\theta}^t,\boldsymbol{\theta}^t}\frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}\left(f(\boldsymbol{x}_i^t;\boldsymbol{\theta}^t), y_i^t\right)\Big|_{\boldsymbol{\theta}^t=\hat{\boldsymbol{\theta}}^t}$$

$$\hat{\boldsymbol{\theta}}^t = \operatorname*{argmin}_{\boldsymbol{\theta}^t}\frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}\left(f(\boldsymbol{x}_i^t;\boldsymbol{\theta}^t), y_i^t\right)$$

Simplify optimization

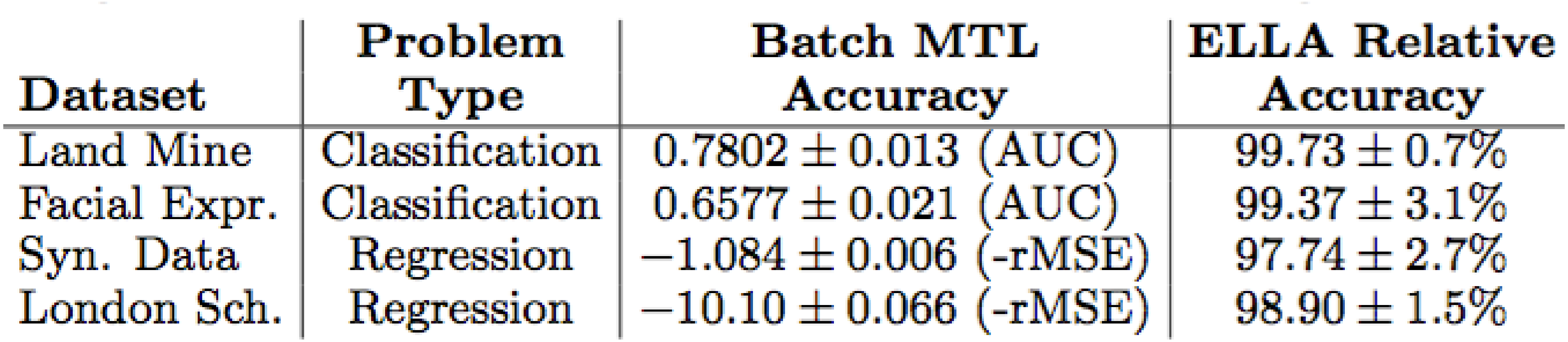
- GO-MTL: when computing a single candidate ***L***, an optimization problem must be solved to re-compute the value of each $s^{(t)}$.

- ELLA: after $s^{(t)}$ is computed given the training data for task *t*, it will not be updated when training on other tasks. Only ***L*** will be changed.

- Note: (Ruvolo and Eaton, 2013b) added the mechanism to actively select the next task to learn.

ELLA Accuracy Result

- ELLA vs. GO-MTL

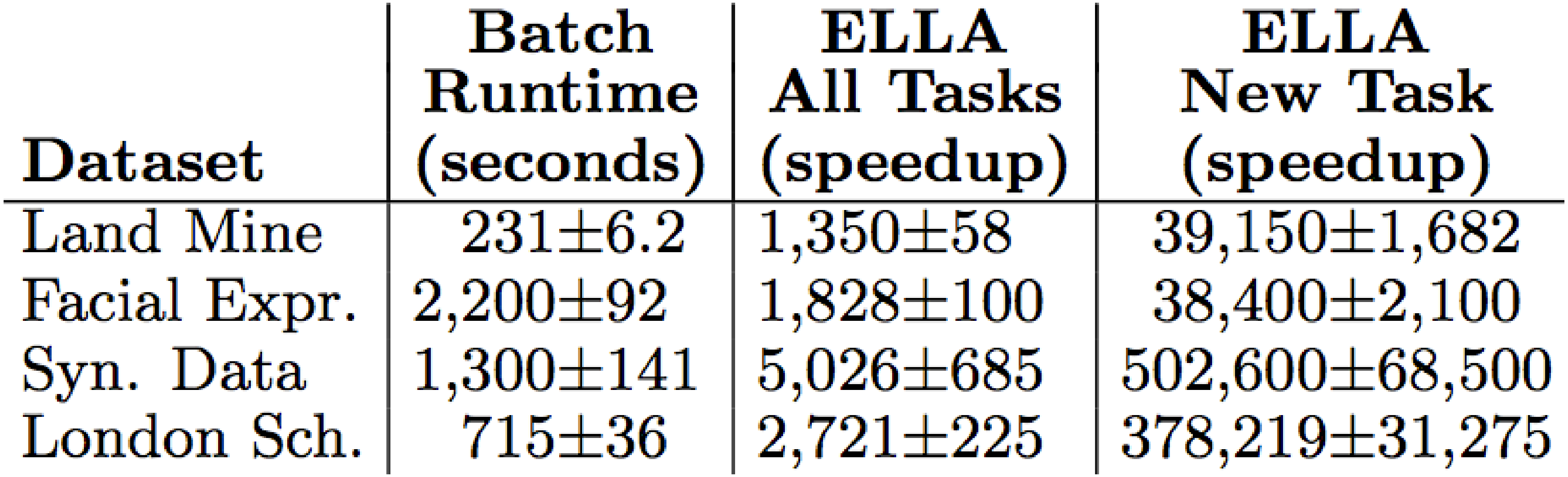


Dataset	Problem Type	Batch MTL Accuracy	ELLA Relative Accuracy
Land Mine	Classification	0.7802 ± 0.013 (AUC)	$99.73 \pm 0.7\%$
Facial Expr.	Classification	0.6577 ± 0.021 (AUC)	$99.37 \pm 3.1\%$
Syn. Data	Regression	-1.084 ± 0.006 (-rMSE)	$97.74 \pm 2.7\%$
London Sch.	Regression	-10.10 ± 0.066 (-rMSE)	$98.90 \pm 1.5\%$

Batch MTL is GO-MTL

ELLA Speed Result

- ELLA vs. GO-MTL



Dataset	Batch Runtime (seconds)	ELLA All Tasks (speedup)	ELLA New Task (speedup)
Land Mine	231±6.2	1,350±58	39,150±1,682
Facial Expr.	2,200±92	1,828±100	38,400±2,100
Syn. Data	1,300±141	5,026±685	502,600±68,500
London Sch.	715±36	2,721±225	378,219±31,275

ELLA is 1K times faster than GO-MTL on all tasks, 30K times on a new task

ELLA in LML

- **KB**
  - **PIS**: Stores all the task data
  - **KS**: matrix $\boldsymbol{L}$ for K basis tasks and S
    - Past knowledge is again re-learned whenever a new task arrives.
  - **KM**: optimization (e.g. alternating optimization strategy)
- **KBL**: Each task parameter vector is a linear combination of **KS**, i.e., $\boldsymbol{\theta}^{(t)} = \boldsymbol{L}s^{(t)}$

Lifelong Sentiment Classification

(Chen, Ma, and Liu 2015)

- "*I bought a cellphone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is great too. ....*"
- Goal: classify docs or sentences as + or -.
  - Need to manually label a lot of training data for each domain, which is highly labor-intensive
- Can we not label for every domain or at least not label so many docs/sentences?

A Simple Lifelong Learning Method

Assuming we have worked on a *large number of past domains* with all their training data *D*

- Build a classifier using *D*, test on new domain
  - Note - using only one past/source domain as in ***transfer learning*** is not good.
- In many cases – improve accuracy by as much as 19% (= 80%-61%). Why?
- In some others cases – not so good, e.g., it works poorly for toy reviews. Why? “toy”

Lifelong Sentiment Classification

(Chen, Ma and Liu, 2015)

- It adopts a Bayesian optimization framework for LML using stochastic gradient decent

- Lifelong learning uses
  - Word counts from the past data as priors.
  - Penalty terms to deal with domain dependent sentiment words and reliability of knowledge.

Naïve Bayesian Text Classification

- Key parameter

$$P\left(w|c_j\right) = \frac{\lambda + N_{c_j,w}}{\lambda|V| + \sum_{v=1}^{|V|} N_{c_j,v}}$$

- Only depends on the count of words in each class

LML Component: PIS

- Probabilities of a word appearing in positive or negative

  $P^{\hat{t}}(w|+)$ and $P^{\hat{t}}(w|-)$

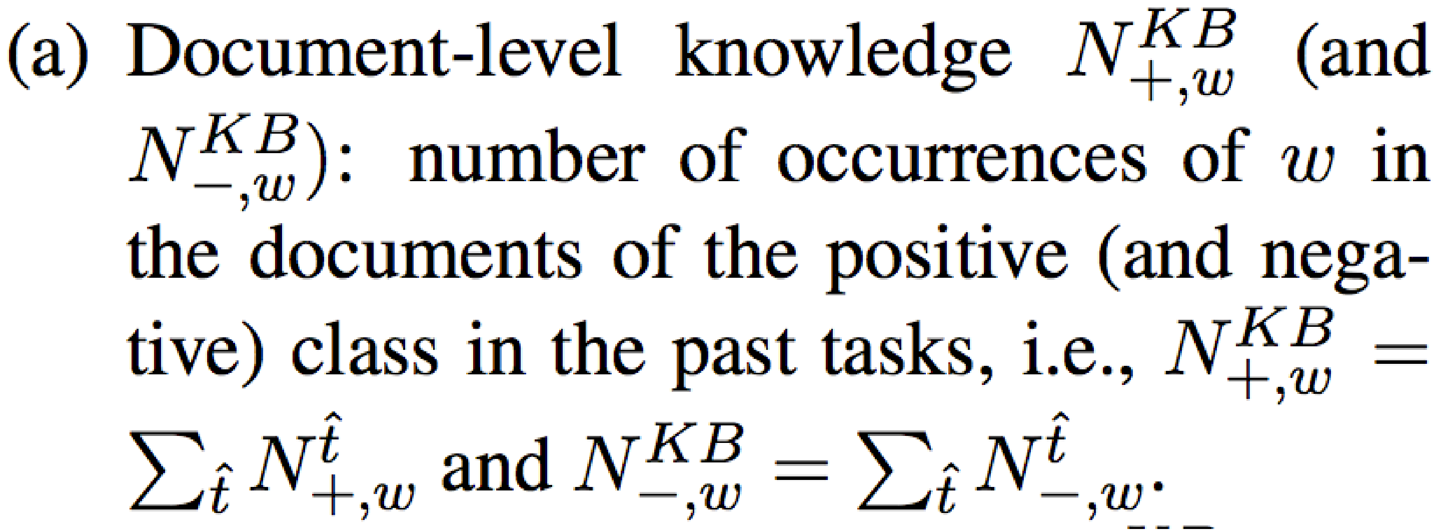
- Word counts
  - Number of times that a word appears in positive class: $N^{\hat{t}}_{+,w}$
  - Number of times that a word appears in negative class: $N^{\hat{t}}_{-,w}$

LML Component: KB

- Two types of knowledge
  - Document-level knowledge
  - Domain-level knowledge

LML Component: KB

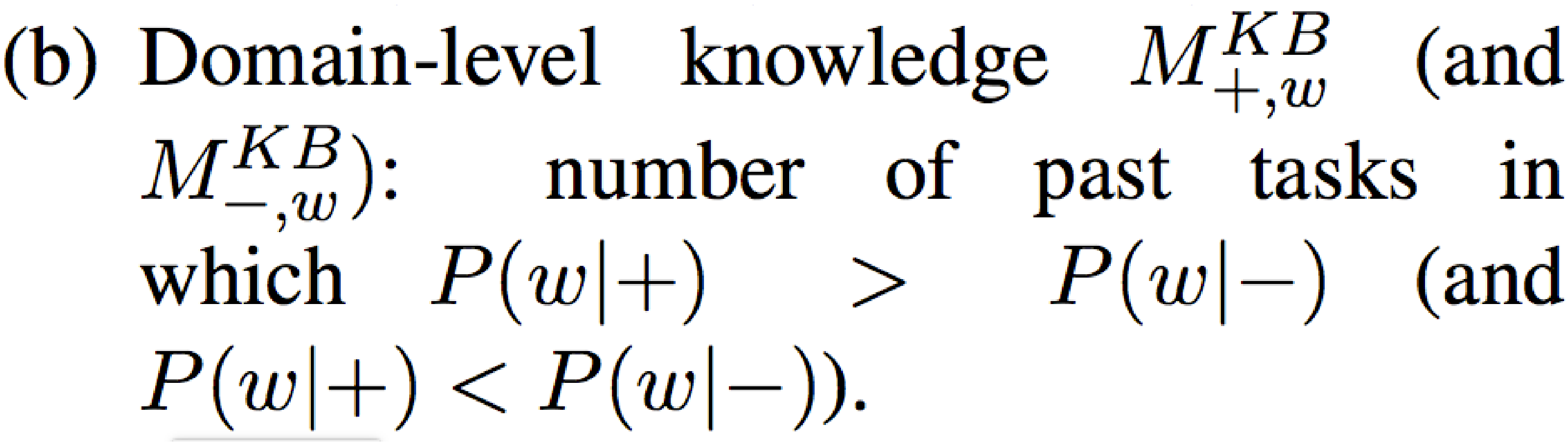
- Two types of knowledge
  - Document-level knowledge
  - Domain-level knowledge



(a) Document-level knowledge $N^{KB}_{+,w}$ (and $N^{KB}_{-,w}$): number of occurrences of w in the documents of the positive (and negative) class in the past tasks, i.e., $N^{KB}_{+,w} = \sum_{\hat{t}} N^{\hat{t}}_{+,w}$ and $N^{KB}_{-,w} = \sum_{\hat{t}} N^{\hat{t}}_{-,w}$.

LML Component: KB

- Two types of knowledge
  - Document-level knowledge
  - Domain-level knowledge



(b) Domain-level knowledge $M_{+,w}^{KB}$ (and $M_{-,w}^{KB}$): number of past tasks in which $P(w|+) > P(w|-)$ (and $P(w|+) < P(w|-)$).

LML Component: KM & KBL

- KM: performs counting and aggregation
- KBL: incorporates knowledge using regularization as penalty terms

Exploiting Knowledge via Penalties

- Penalty terms for two types of knowledge
  - Document-level knowledge
  - Domain-level knowledge

Exploiting Knowledge via Penalties

- Penalty terms for two types of knowledge
  - Document-level knowledge
  - Domain-level knowledge

$$\frac{1}{2}\alpha \sum_{w\in V_T} \left(\left(X_{+,w} - N^t_{+,w}\right)^2 + \left(X_{-,w} - N^t_{-,w}\right)^2 \right)$$

- - *t* is the new task

Exploiting Knowledge via Penalties

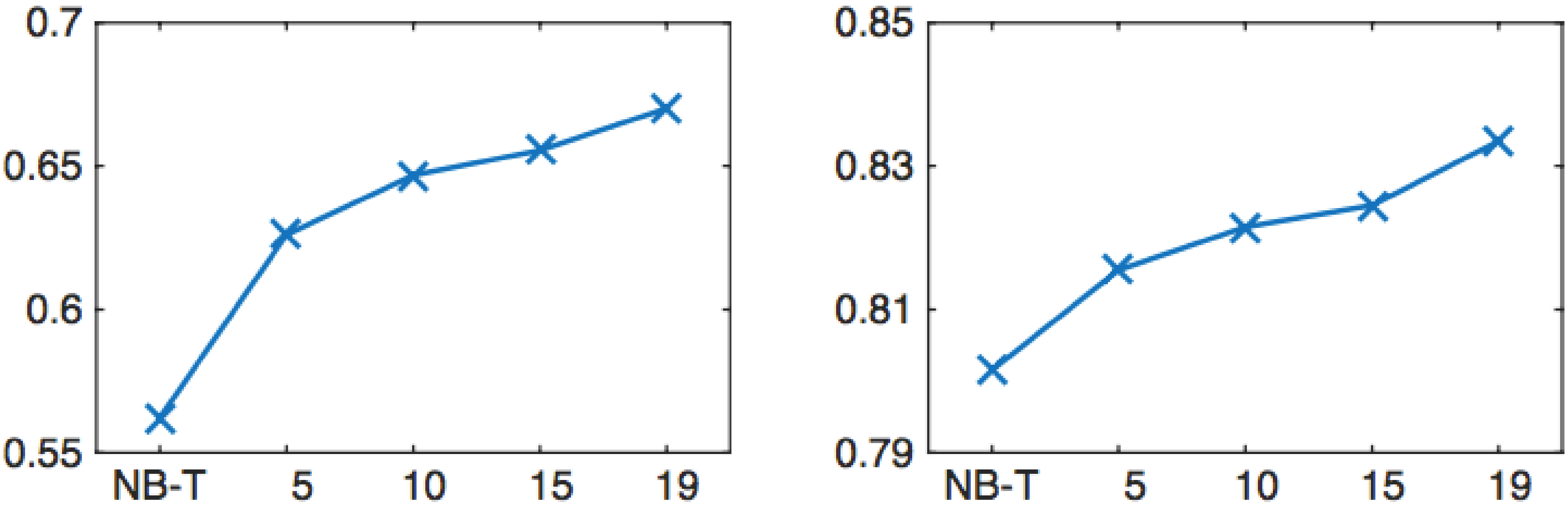
- Penalty terms for two types of knowledge
  - Document-level knowledge
  - Domain-level knowledge

$$\frac{1}{2}\alpha \sum_{w \in V_S} \left(X_{+,w} - R_w \times X^0_{+,w}\right)^2$$
$$+\frac{1}{2}\alpha \sum_{w \in V_S} \left(X_{-,w} - (1 - R_w) \times X^0_{-,w}\right)^2$$

  - R_W: ratio of #tasks where *w* is positive / #all tasks
  - $X^0_{+,w} = N^t_{+,w} + N^{KB}_{+,w}$ and $X^0_{-,w} = N^t_{-,w} + N^{KB}_{-,w}$

One Result of LSC model

- Better F1-score (left) and accuracy (right) with more past tasks



Cumulative Learning

- Cumulative learning (Fei et al., KDD-2016)
  - Open (World) Classification or Learning
    - Detecting unseen classes in testing

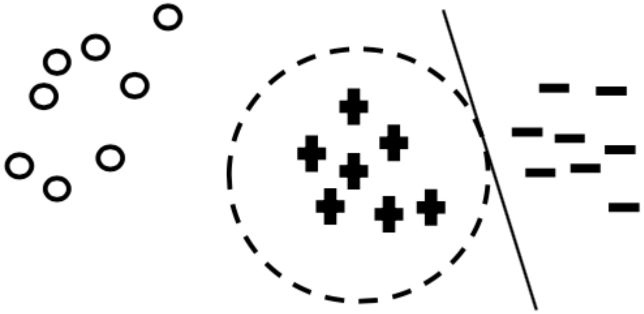
Toward self-learning

- Cumulative learning (Fei et al., KDD-2016)
  - Open (World) Classification or Learning
    - Detecting unseen classes in testing
- Incrementally adding new classes without re-training the whole model from scratch
  - At each time point, a new class is introduced.
  - The new task is the combination of all classes
- Self-learning: realizing something is new and learning it makes self-learning possible.

Based on space transformation

- Based on center-based similarity space (CBS) learning

- Each class has a center point and a circle range
  - Instances fall into it are more likely to belong to this class.



Main steps

- Search for a set of classes *SC* that are similar to the new (*N* + 1) class
- Learn to separate the new class and the classes in *SC*
- Build a new model for the new class, update the models for classes in *SC*

Outline

- A motivating example
- What is lifelong machine learning?
- Related learning tasks
- Lifelong supervised learning
- Semi-supervised never-ending learning
- Lifelong unsupervised learning
- Lifelong reinforcement learning
- Summary

Humans learn many things, for years, and become better learners over time

Why not machines?

Never-Ending Learning

We'll never really understand learning until we build machines that

- learn many different things,
- over years,
- and become better learners over time.

Never-Ending Learning

We'll never produce natural language understanding systems until we have systems that react to arbitrary sentences by saying one of:

- I understand, and already knew that
- I understand, and didn't know, but accept it
- I understand, and disagree because …

Never-Ending Learning

Mitchell et al., 2015

- Main Task: acquire a growing competence without asymptote
  - over years
  - multiple functions
  - where learning one thing improves ability to learn the next
  - acquiring data from humans, environment
- Many candidate domains:
  - Robots
  - Softbots
  - Game players

NELL: Never-Ending Language Learner

Inputs:

- initial ontology
- handful of examples of each predicate in ontology
- the web
- occasional interaction with human trainers

The task:

- run 24x7, forever
- each day:
  1. extract more facts from the web to populate the initial ontology
  2. learn to read (perform #1) better than yesterday

NELL: Never-Ending Language Learner

Goal:

- run 24x7, forever
- each day:
  1. extract more facts from the web to populate given ontology
  2. learn to read better than yesterday

Today...

Running 24 x 7, since January, 2010

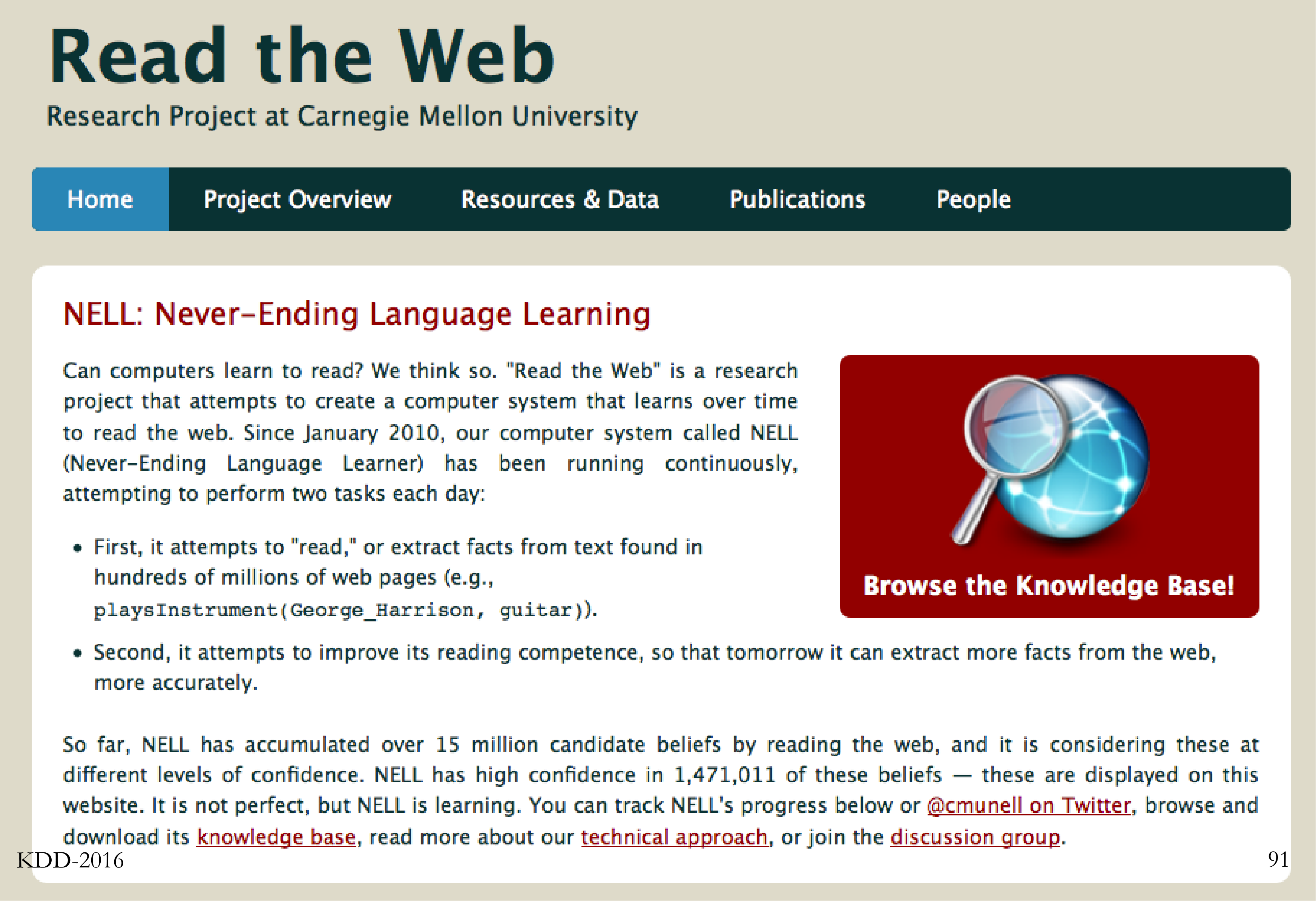
Input:

- ontology defining ~800 categories and relations
- 10-20 seed examples of each
- 1 billion web pages (ClueWeb – Jamie Callan)

Result:

- continuously growing KB with +90,000,000 extracted beliefs

http://rtw.ml.cmu.edu



Read the Web

Research Project at Carnegie Mellon University

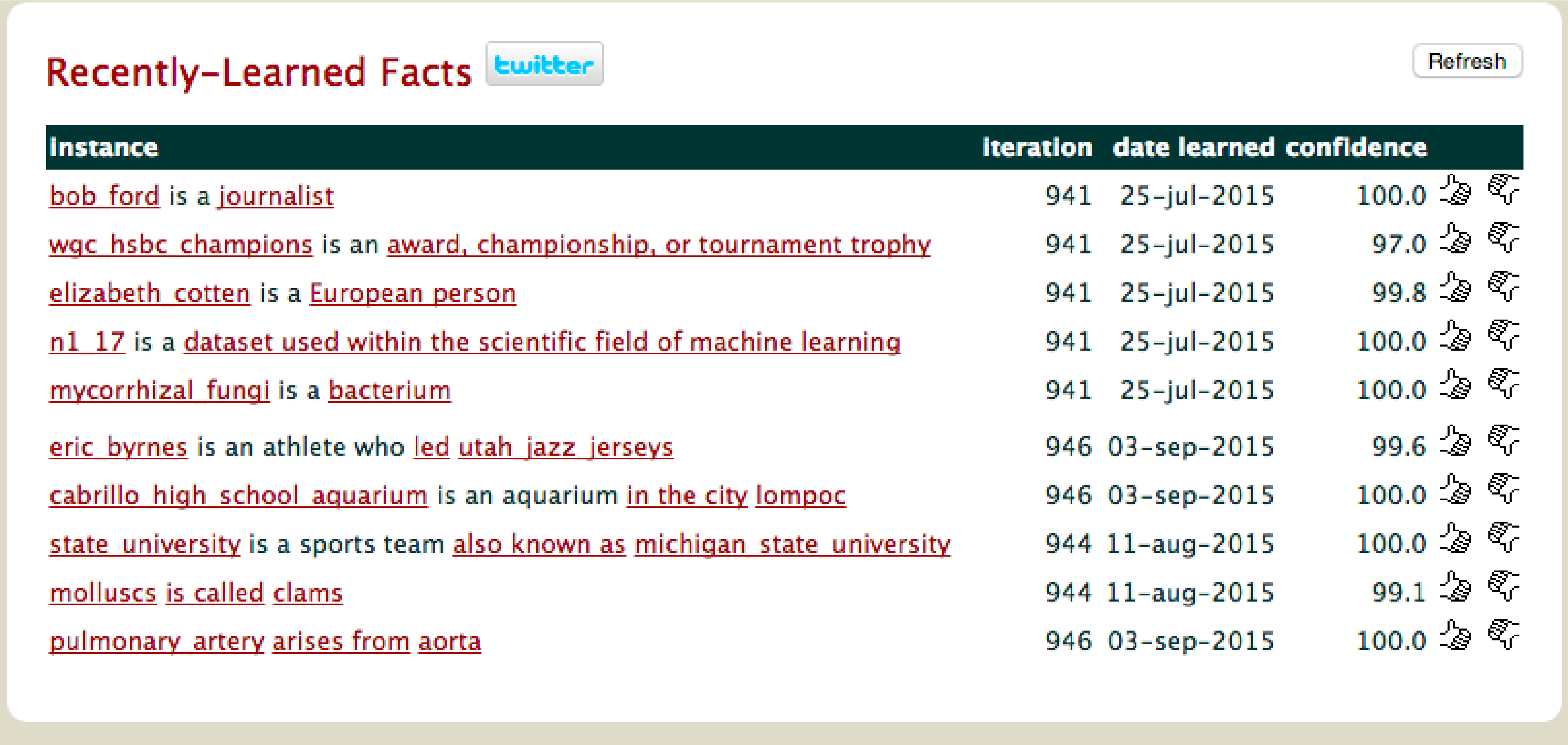
NELL: Never-Ending Language Learning

Can computers learn to read? We think so. "Read the Web" is a research project that attempts to create a computer system that learns over time to read the web. Since January 2010, our computer system called NELL (Never-Ending Language Learner) has been running continuously, attempting to perform two tasks each day:

- First, it attempts to "read," or extract facts from text found in hundreds of millions of web pages (e.g., `playsInstrument(George_Harrison, guitar)`).
- Second, it attempts to improve its reading competence, so that tomorrow it can extract more facts from the web, more accurately.

So far, NELL has accumulated over 15 million candidate beliefs by reading the web, and it is considering these at different levels of confidence. NELL has high confidence in 1,471,011 of these beliefs — these are displayed on this website. It is not perfect, but NELL is learning. You can track NELL's progress below or @cmunell on Twitter, browse and download its knowledge base, read more about our technical approach, or join the discussion group.

NELL: Never-Ending Language Learner



Recently-Learned Facts

instance	iteration	date learned	confidence	
bob ford is a journalist	941	25-jul-2015	100.0	👍 👎
wgc hsbc champions is an award, championship, or tournament trophy	941	25-jul-2015	97.0	👍 👎
elizabeth cotten is a European person	941	25-jul-2015	99.8	👍 👎
n1 17 is a dataset used within the scientific field of machine learning	941	25-jul-2015	100.0	👍 👎
mycorrhizal fungi is a bacterium	941	25-jul-2015	100.0	👍 👎
eric byrnes is an athlete who led utah jazz jerseys	946	03-sep-2015	99.6	👍 👎
cabrillo high school aquarium is an aquarium in the city lompoc	946	03-sep-2015	100.0	👍 👎
state university is a sports team also known as michigan state university	944	11-aug-2015	100.0	👍 👎
molluscs is called clams	944	11-aug-2015	99.1	👍 👎
pulmonary artery arises from aorta	946	03-sep-2015	100.0	👍 👎

Computer Reading the Web

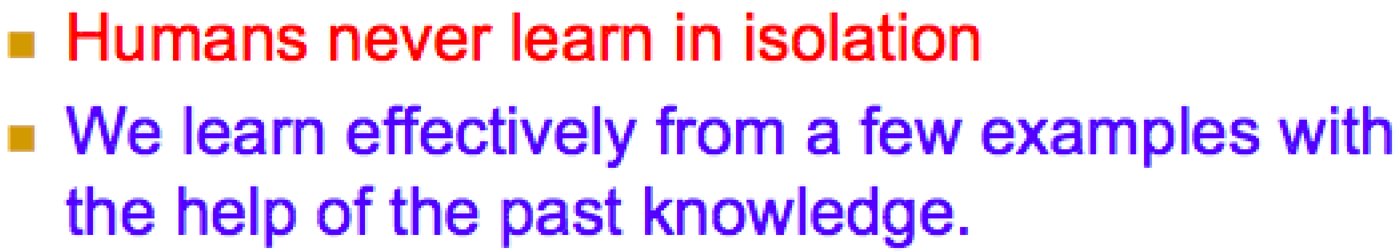
1. Classify noun phrases (NP's) by category

The Problem with Semi-Supervised Bootstrap Learning

- Paris
- Pittsburgh
- Seattle
- Cupertino

The Problem with Semi-Supervised Bootstrap Learning

- Paris
- Pittsburgh
- Seattle
- Cupertino



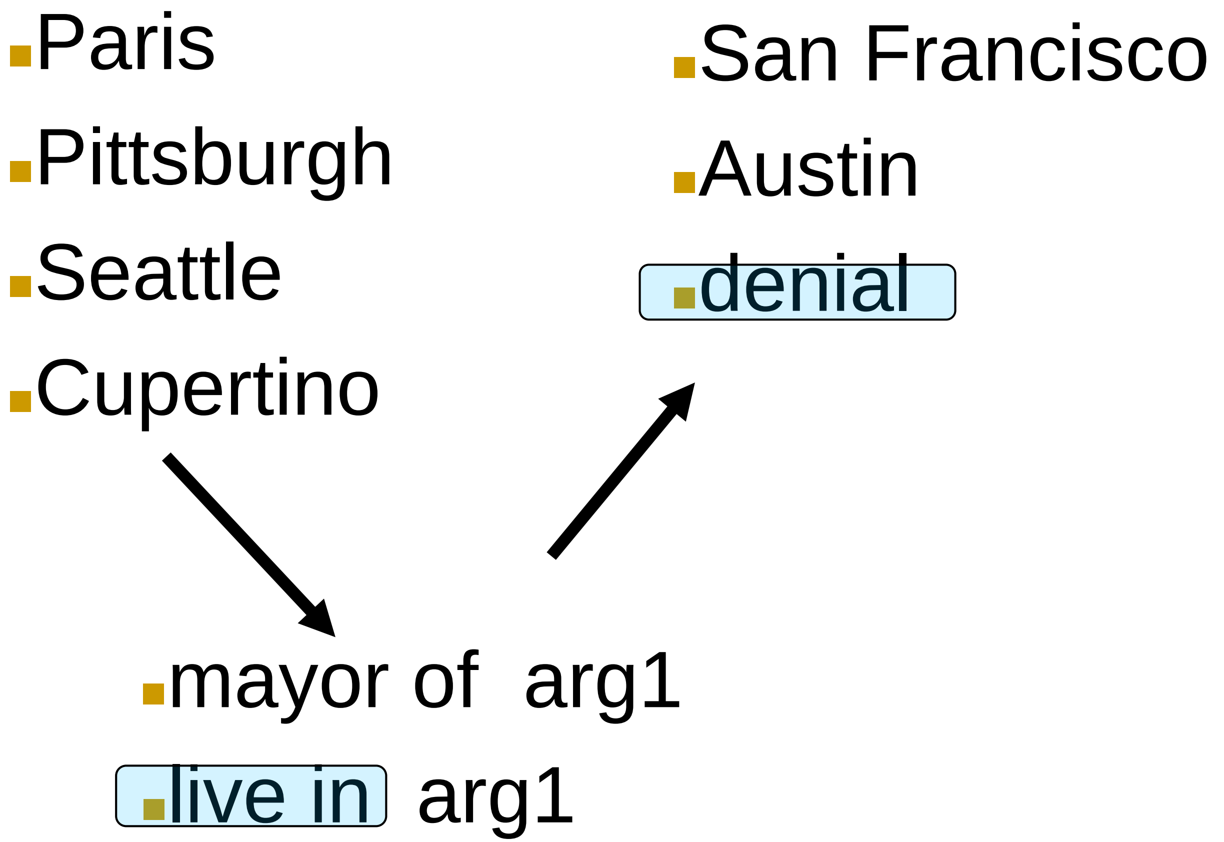
- Humans never learn in isolation
- We learn effectively from a few examples with the help of the past knowledge.

The Problem with Semi-Supervised Bootstrap Learning

- Paris
- Pittsburgh
- Seattle
- Cupertino

  - mayor of arg1
  - live in arg1

The Problem with Semi-Supervised Bootstrap Learning

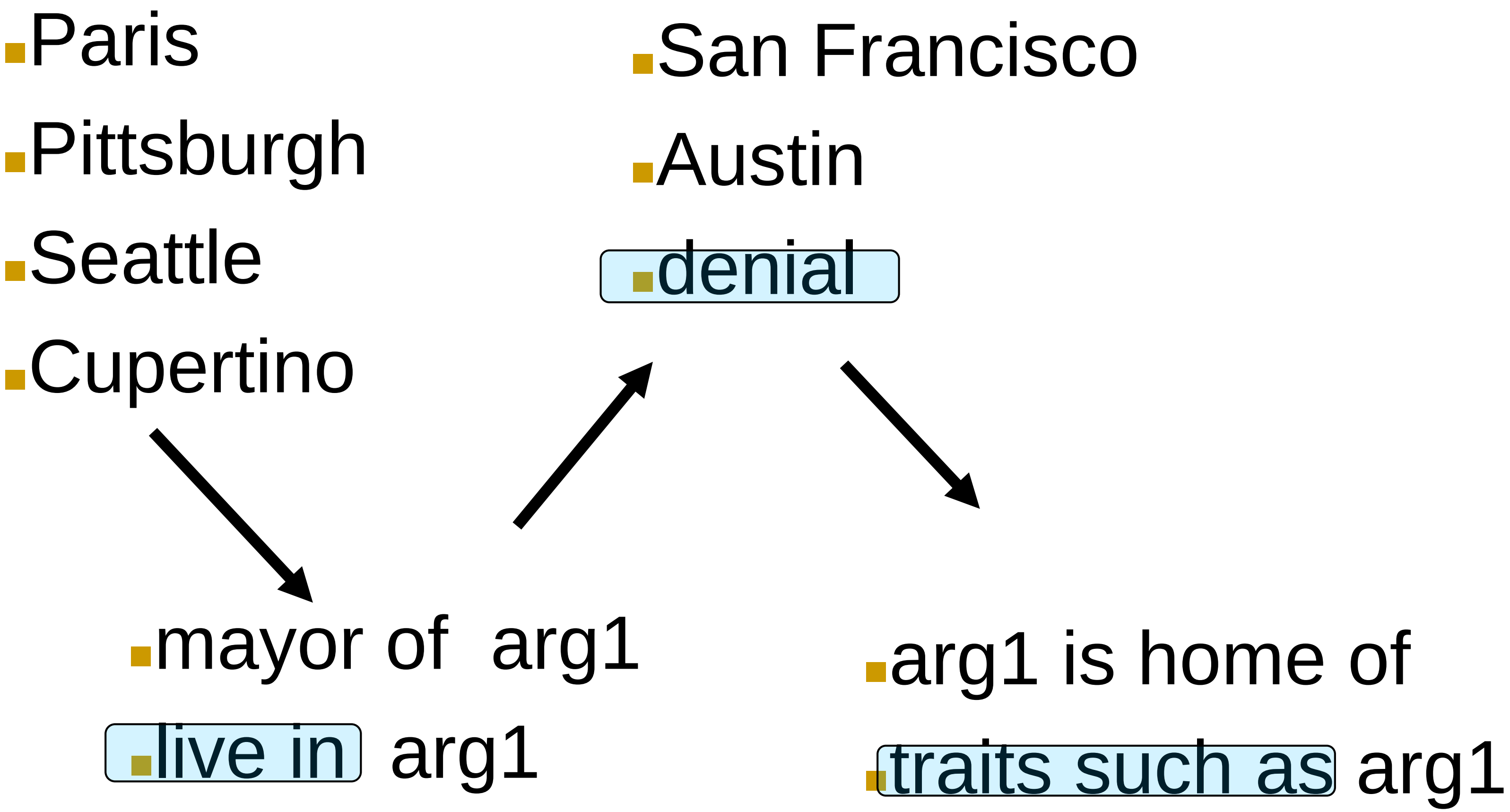


- Paris
- Pittsburgh
- Seattle
- Cupertino

- San Francisco
- Austin
- denial

- mayor of arg1
- live in arg1

The Problem with Semi-Supervised Bootstrap Learning



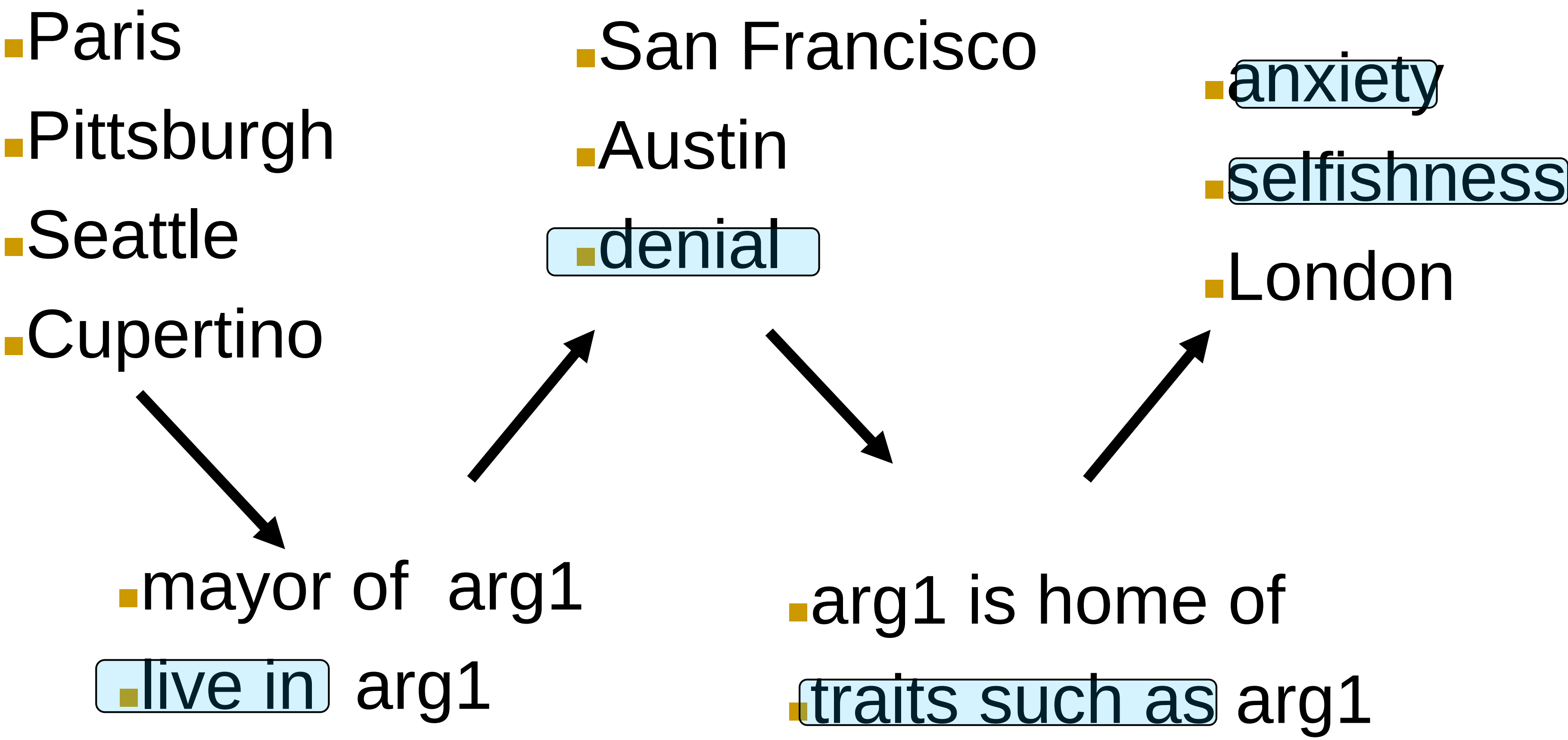
- Paris
- Pittsburgh
- Seattle
- Cupertino

- San Francisco
- Austin
- denial

- mayor of arg1
- live in arg1

- arg1 is home of
- traits such as arg1

The Problem with Semi-Supervised Bootstrap Learning



- Paris
- Pittsburgh
- Seattle
- Cupertino

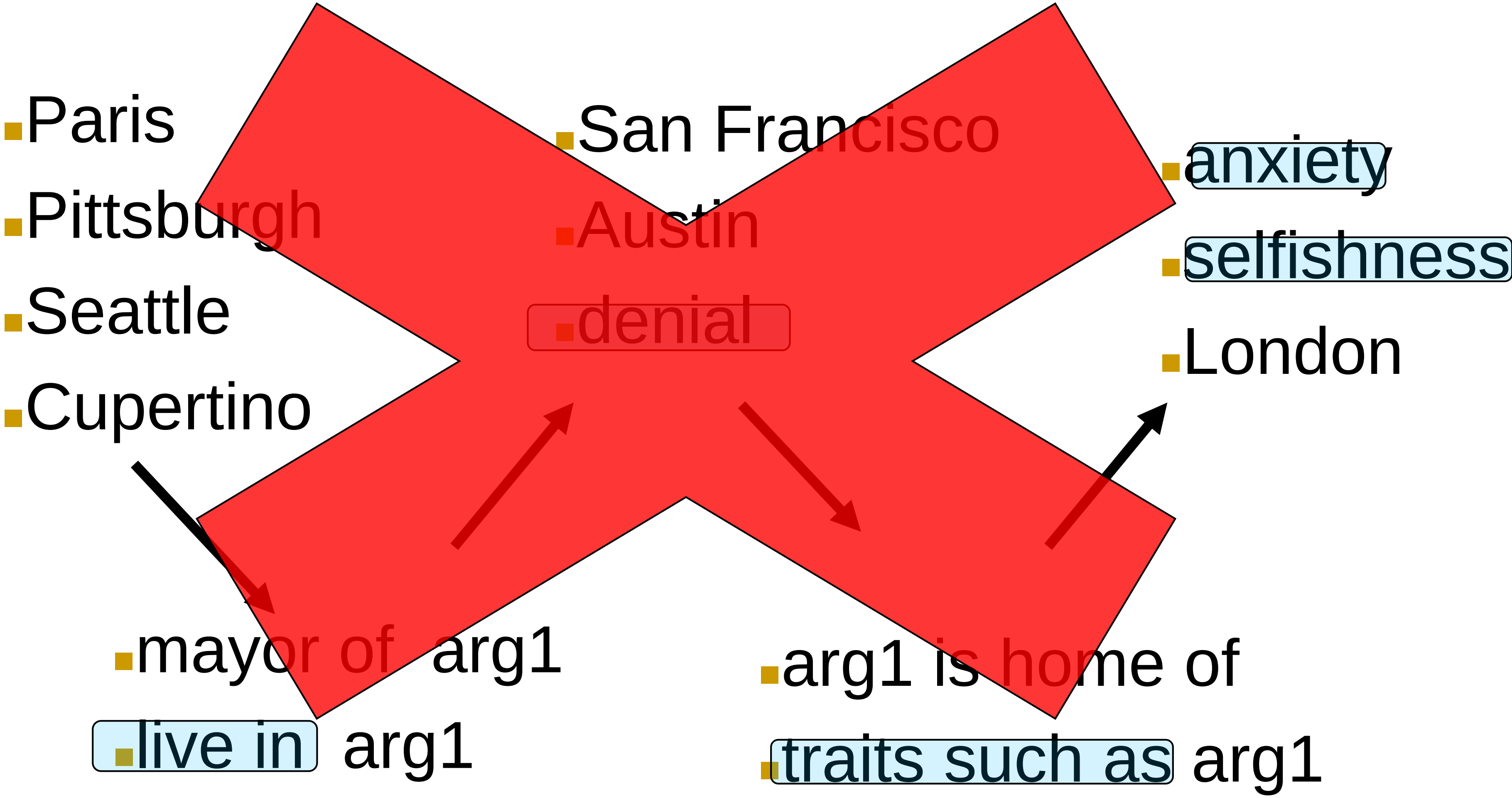
- mayor of arg1
- live in arg1

- San Francisco
- Austin
- denial

- arg1 is home of
- traits such as arg1

- anxiety
- selfishness
- London

The Problem with Semi-Supervised Bootstrap Learning



- Paris
- Pittsburgh
- Seattle
- Cupertino

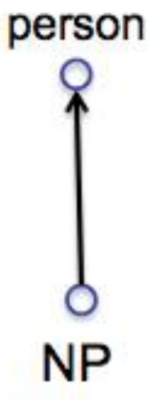
- mayor of arg1
- live in arg1

- San Francisco
- Austin
- denial

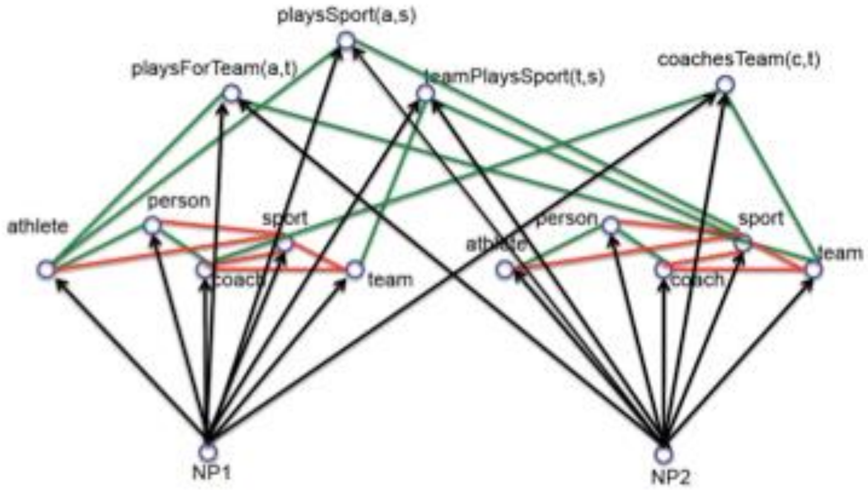
- arg1 is home of
- traits such as arg1

- anxiety
- selfishness
- London

Key Idea 1: Coupled semi-supervised training of many functions

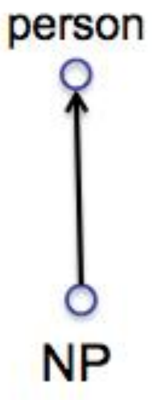


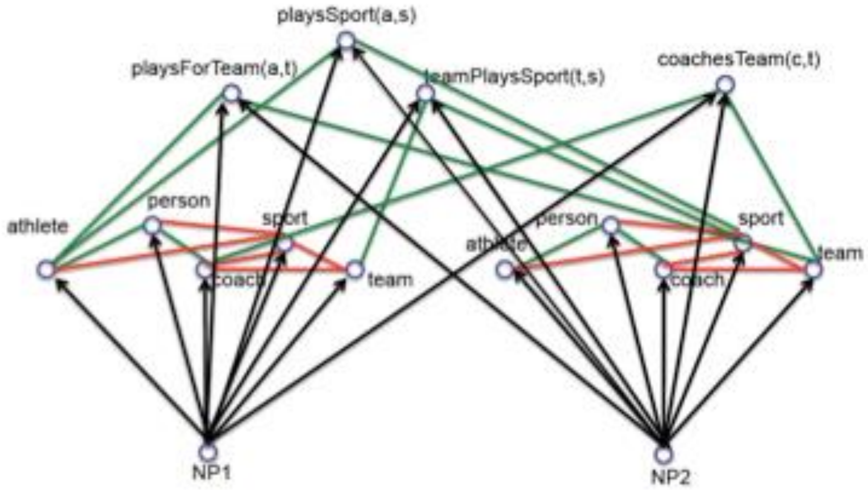
hard (underconstrained) semi-supervised learning problem



much easier (more constrained) semi-supervised learning problem

Key Idea 1: Coupled semi-supervised training of many functions





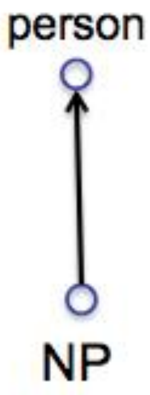
hard (underconstrained) semi-supervised learning problem

much easier (more constrained) semi-supervised learning problem

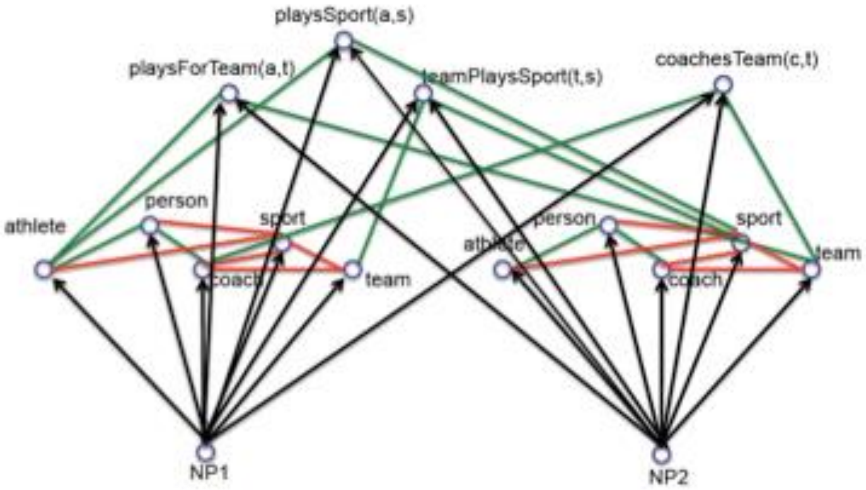
Let's call this: Machine Learning (ML) 1.0

- Isolated learning has limitations.

Key Idea 1: Coupled semi-supervised training of many functions



hard (underconstrained) semi-supervised learning problem



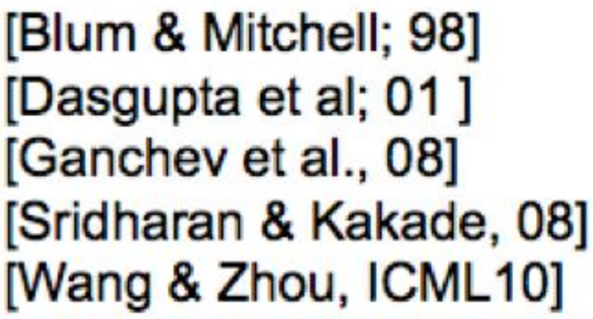
much easier (more constrained) semi-supervised learning problem

- It is rather "silly" not to exploit such sharing in learning or extraction.

Let's call this: Machine Learning (ML) 1.0

- Isolated learning has limitations.

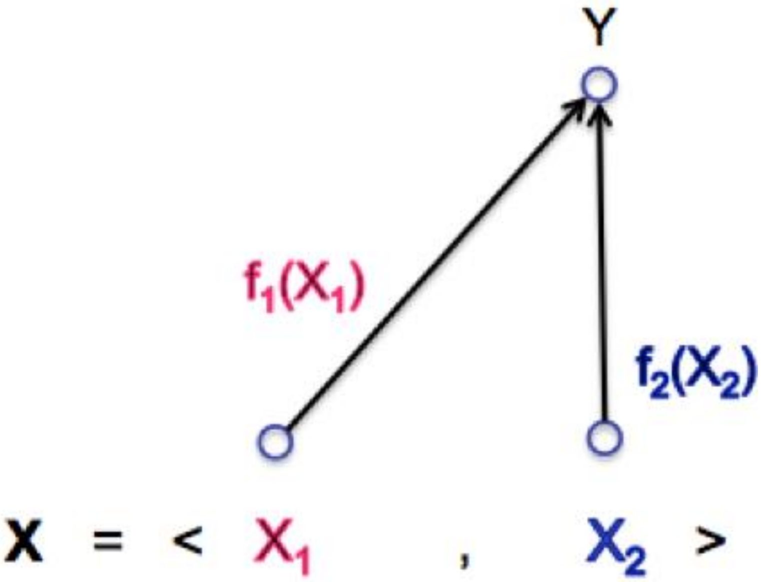
Coupled Training Type 1: Co-training, Multiview, Co-regularization



[Blum & Mitchell; 98]
[Dasgupta et al; 01]
[Ganchev et al., 08]
[Sridharan & Kakade, 08]
[Wang & Zhou, ICML10]

Coupled Training Type 1: Co-training, Multiview, Co-regularization

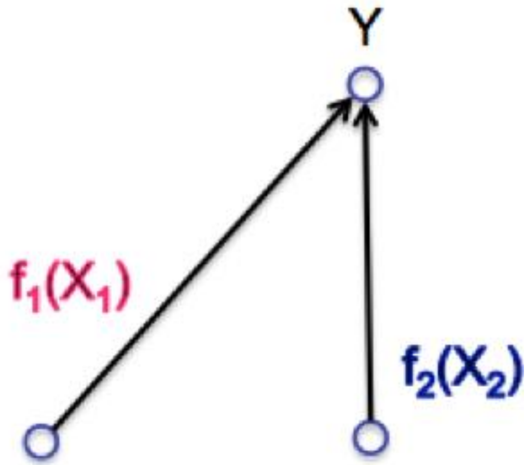
[Blum & Mitchell; 98]
[Dasgupta et al; 01]
[Ganchev et al., 08]
[Sridharan & Kakade, 08]
[Wang & Zhou, ICML10]



Constraint: $f_1(x_1) = f_2(x_2)$

Coupled Training Type 1: Co-training, Multiview, Co-regularization

[Blum & Mitchell; 98]
[Dasgupta et al; 01]
[Ganchev et al., 08]
[Sridharan & Kakade, 08]
[Wang & Zhou, ICML10]



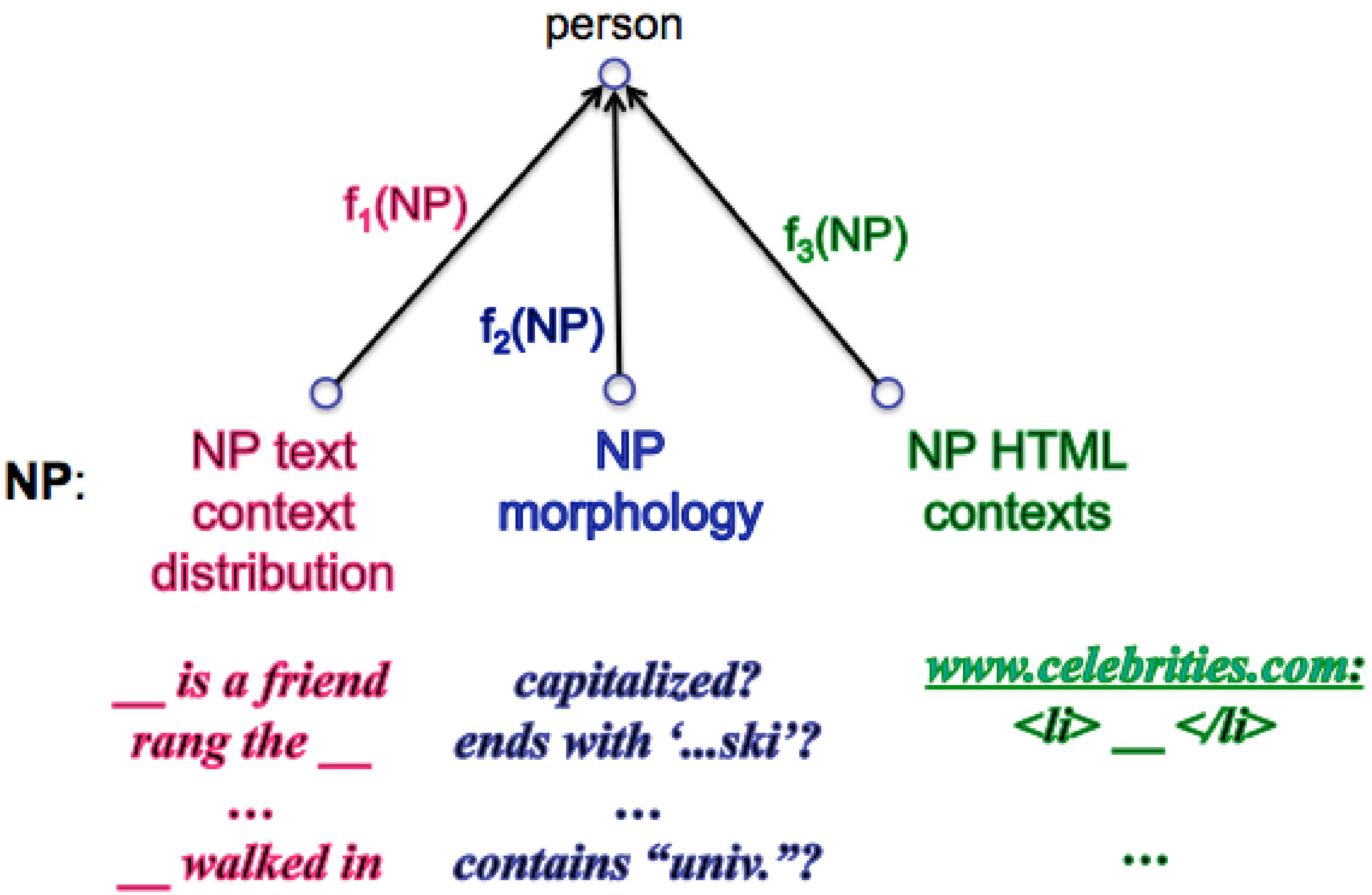
$\mathbf{X} = < X_1 \quad , \quad X_2 >$

Constraint: $f_1(x_1) = f_2(x_2)$

If f_1, f_2 PAC learnable,
X_1, X_2 conditionally indep
Then PAC learnable from
unlabeled data and
weak initial learner

and disagreement between
f_1, f_2 bounds error of each

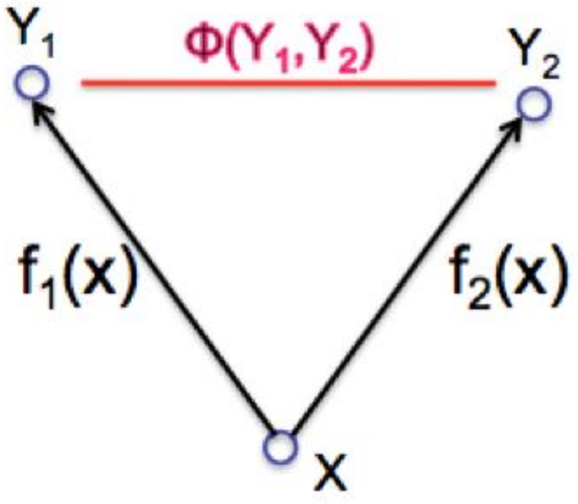
Type 1 Coupling Constraints in NELL



Coupled Training Type 2:

Structured Outputs, Multitask, Posterior Regularization, Multilabel

[Daume, 2008]
[Bakhir et al., eds. 2007]
[Roth et al., 2008]
[Taskar et al., 2009]
[Carlson et al., 2009]

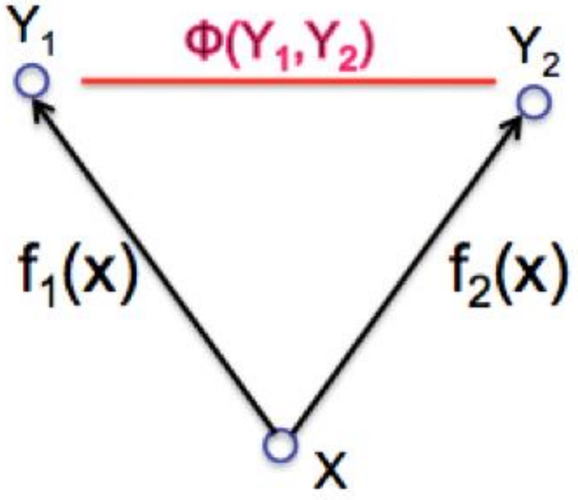


Constraint: $\Phi(f_1(x), f_2(x))$

Coupled Training Type 2:

Structured Outputs, Multitask, Posterior Regularization, Multilabel

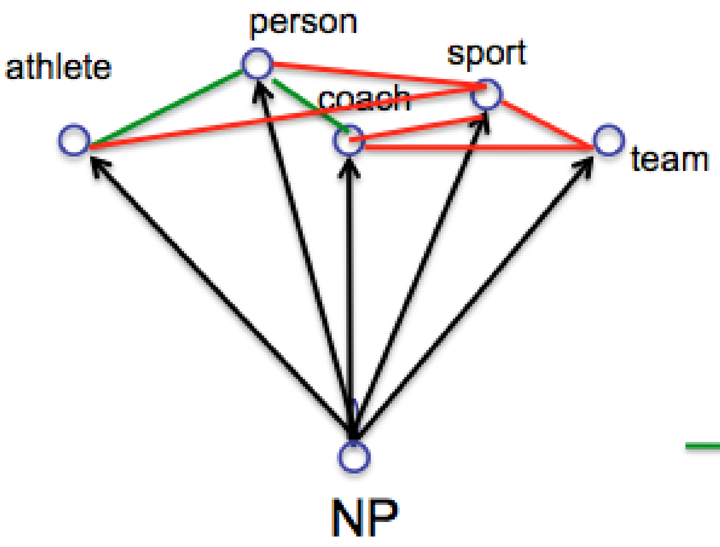
[Daume, 2008]
[Bakhir et al., eds. 2007]
[Roth et al., 2008]
[Taskar et al., 2009]
[Carlson et al., 2009]



Effectiveness ~ probability that $\Phi(Y_1,Y_2)$ will be violated by incorrect f_j and f_k

Constraint: $\Phi(f_1(x), f_2(x))$

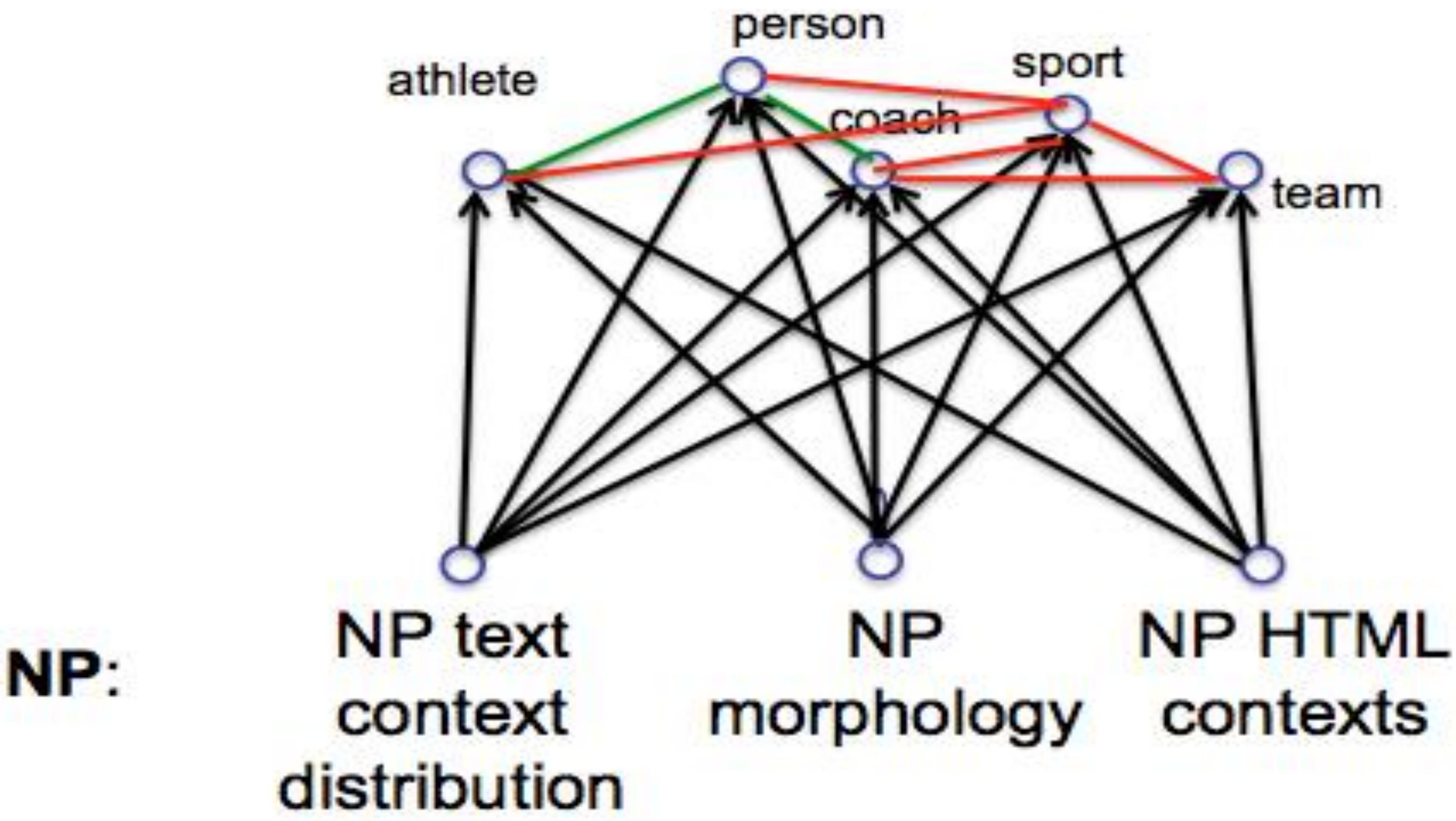
Type 2 Coupling Constraints in NELL



🟩 athlete(NP) → person(NP)

🟥 athlete(NP) → NOT sport(NP)
NOT athlete(NP) ← sport(NP)

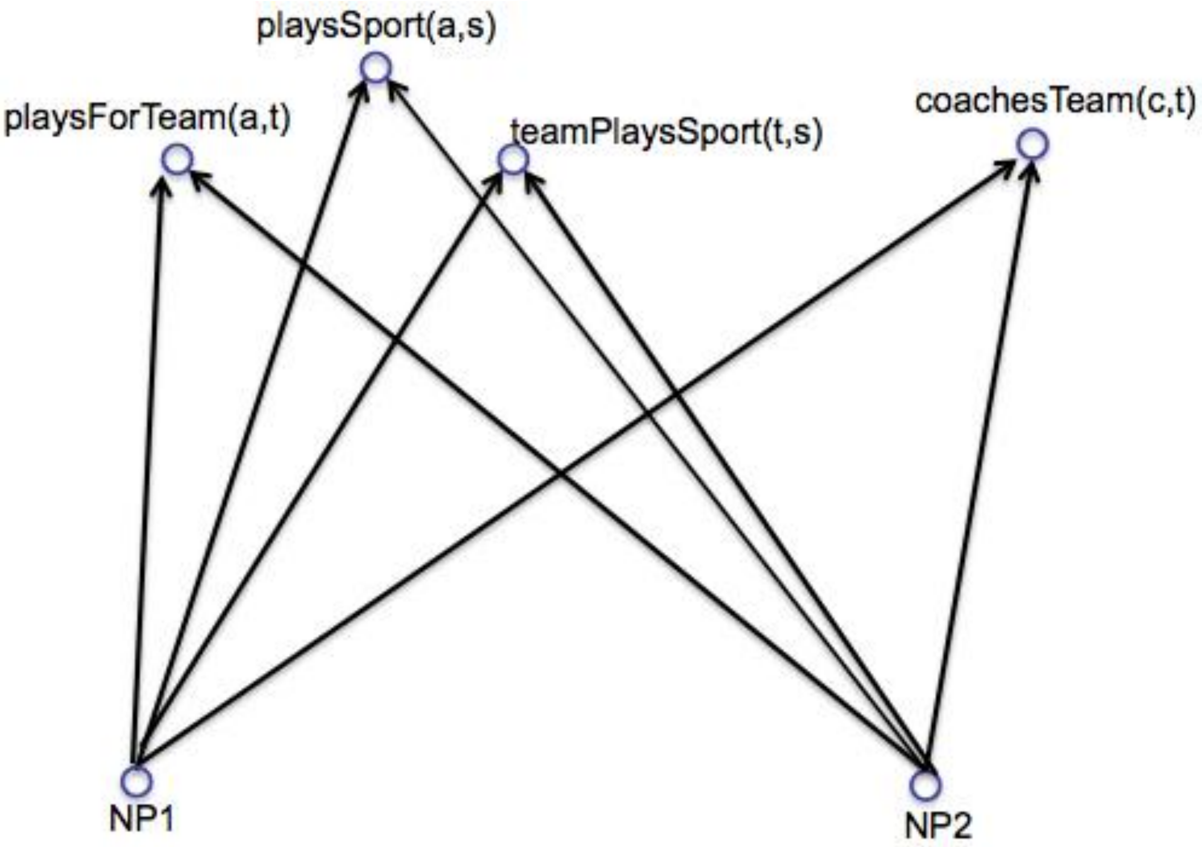
Multi-view, Multi-Task Coupling



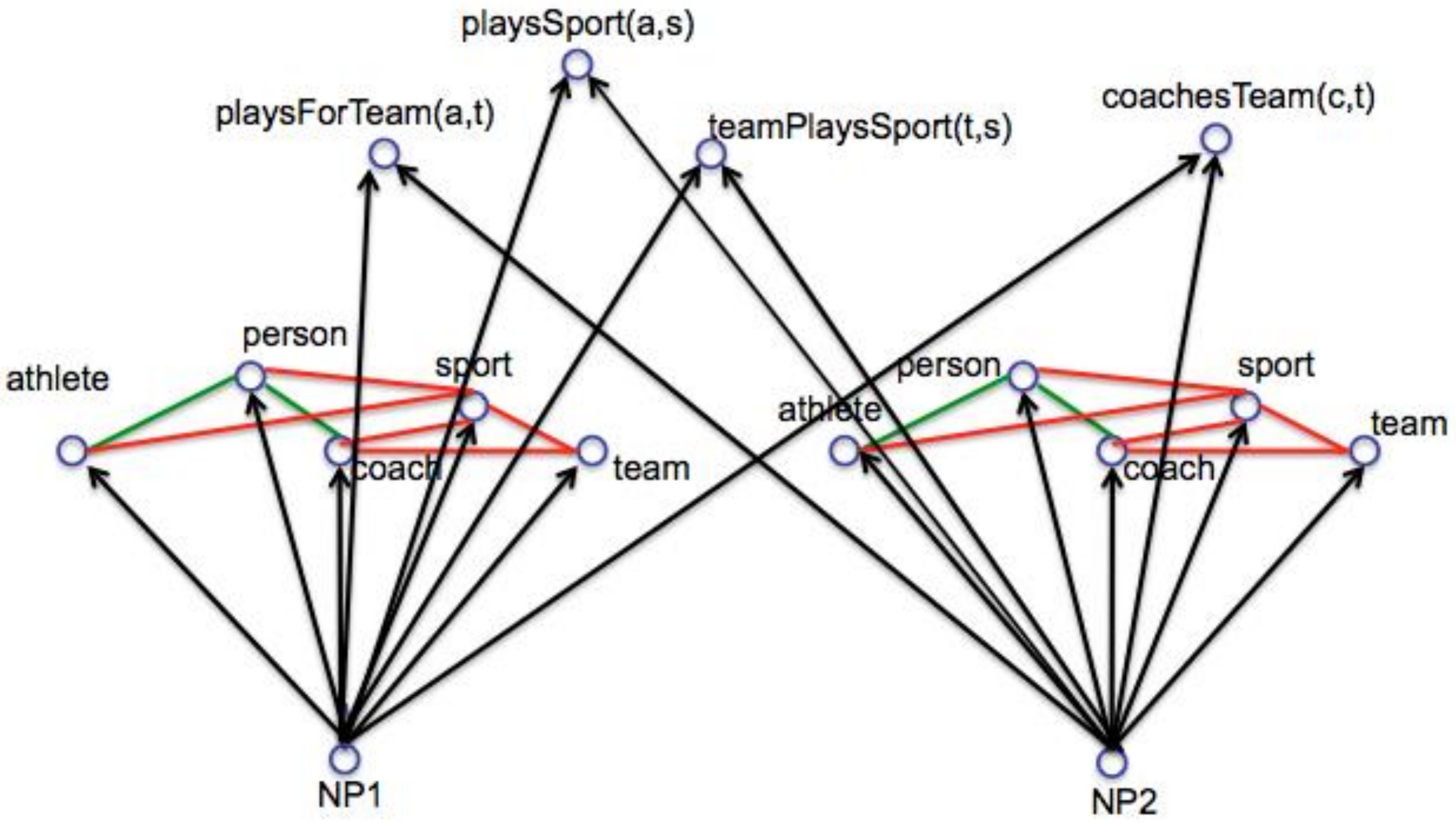
Computer Reading the Web

1. Classify noun phrases (NP's) by category
2. Classify NP pairs by relation

Learning Relations between NP’s

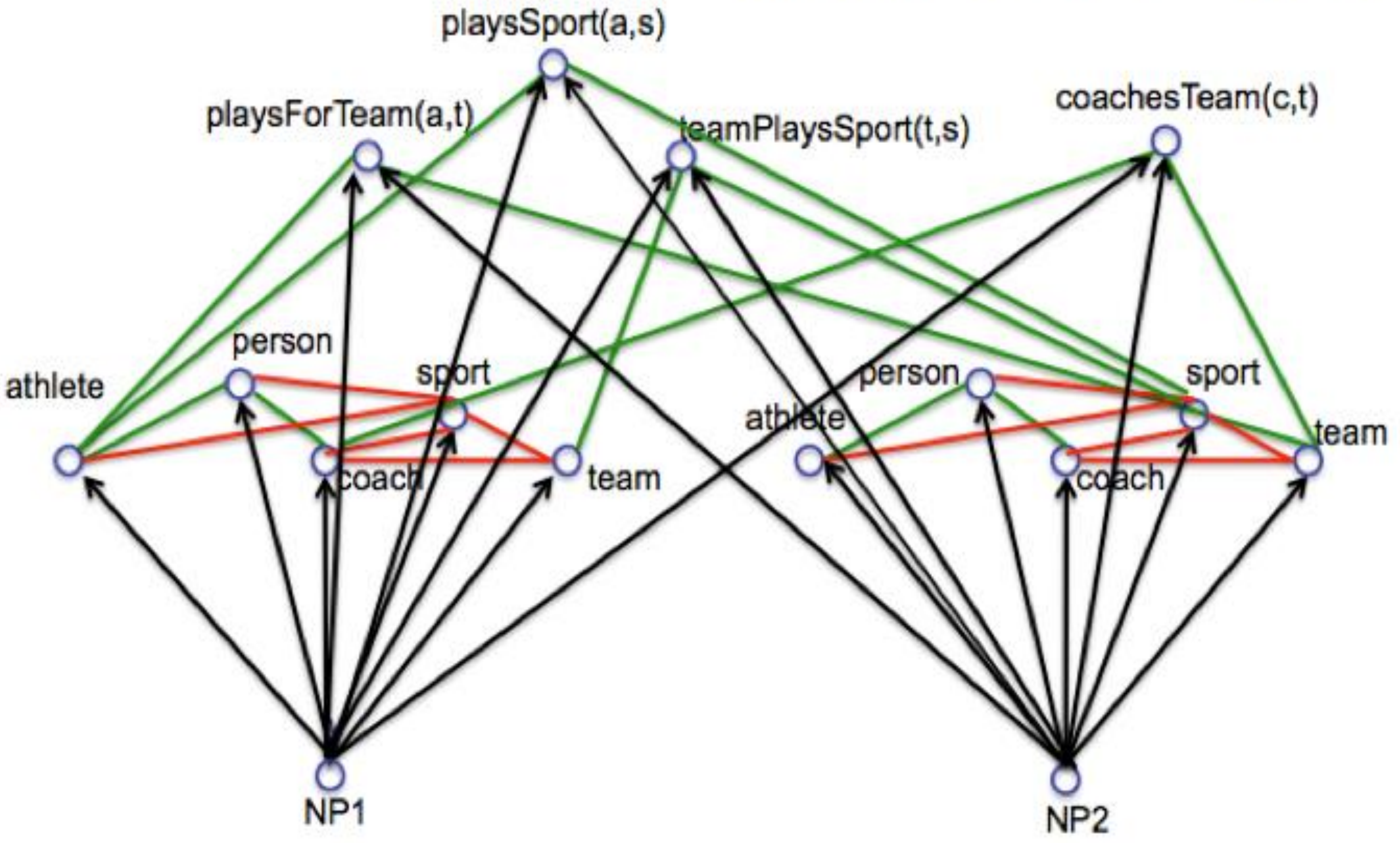


Learning Relations between NP's



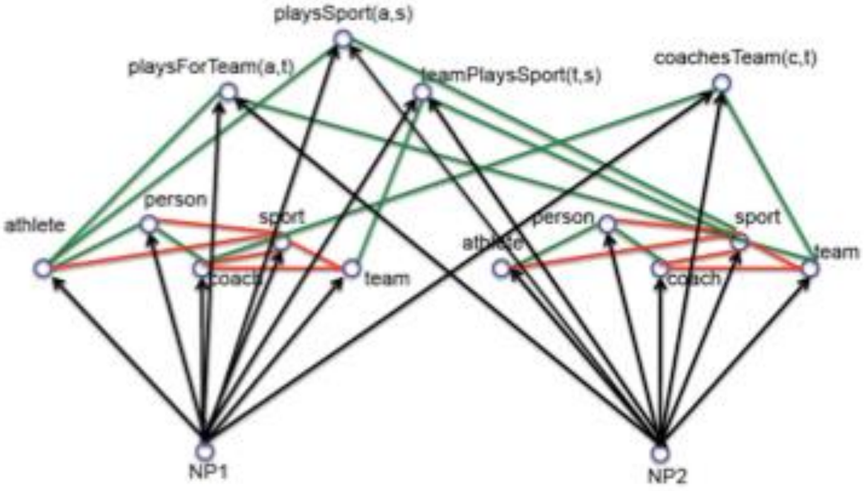
Type 3 Coupling: Argument Types

Constraint: f3(x1,x2) → (f1(x1) AND f2(x2))



— **playsSport(NP1,NP2) → athlete(NP1), sport(NP2)**

Pure EM Approach to Coupled Training



E: jointly estimate latent labels for each function of each unlabeled example

M: retrain all functions, based on these probabilistic labels

Scaling problem:

- **E** step: 20M NP's, 10^{14} NP pairs to label
- **M** step: 50M text contexts to consider for each function □ 10^{10} parameters to retrain
- even more URL-HTML contexts..

NELL’s Approximation to EM

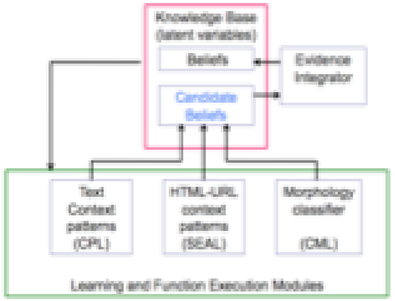
E’ step:

- Consider only a growing subset of the latent variable assignments
  - category variables: up to 250 NP’s per category per iteration
  - relation variables: add only if confident and args of correct type
  - this set of explicit latent assignments ***IS*** the knowledge base

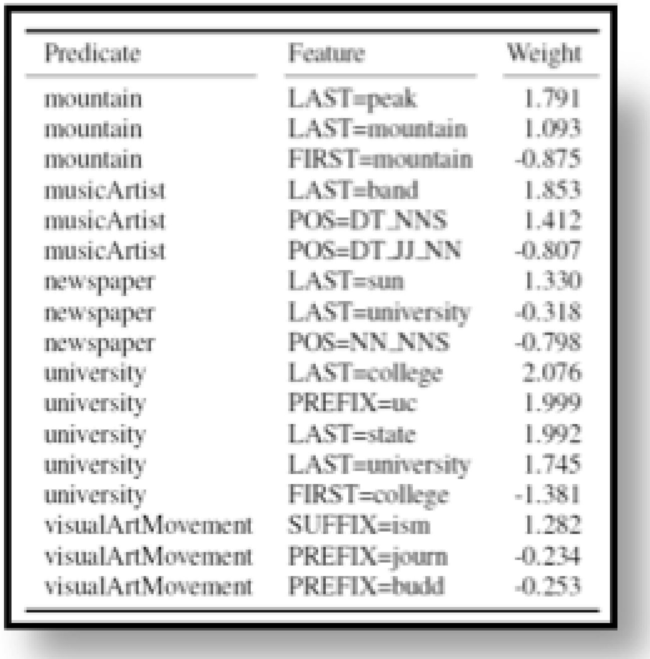
M’ step:

- Each view-based learner retrains itself from the updated KB
- “context” methods create growing subsets of contexts

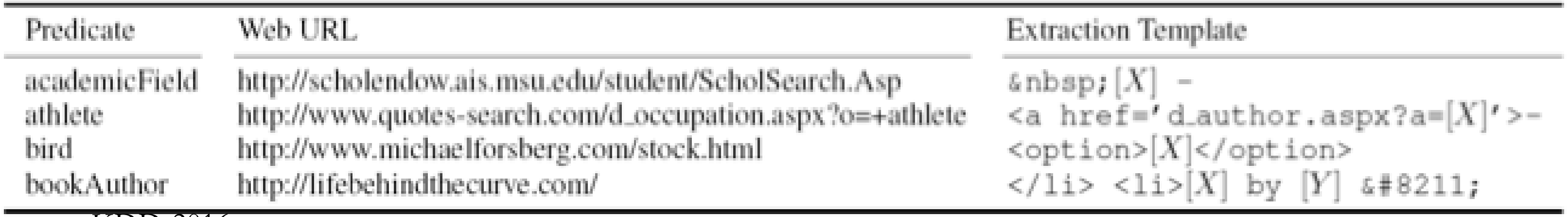
Never-Ending Language Learning



arg1_was_playing_arg2 arg2_megastar_arg1 arg2_icons_arg1
arg2_player_named_arg1 arg2_prodigy_arg1
arg1_is_the_tiger_woods_of_arg2 arg2_career_of_arg1
arg2_greats_as_arg1 arg1_plays_arg2 arg2_player_is_arg1
arg2_legends_arg1 arg1_announced_his_retirement_from_arg2
arg2_operations_chief_arg1 arg2_player_like_arg1
arg2_and_golfing_personalities_including_arg1 arg2_players_like_arg1
arg2_greats_like_arg1 arg2_players_are_steffi_graf_and_arg1
arg2_great_arg1 arg2_champ_arg1 arg2_greats_such_as_arg1
arg2_professionals_such_as_arg1 arg2_hit_by_arg1 arg2_greats_arg1
arg2_icon_arg1 arg2_stars_like_arg1 arg2_pros_like_arg1
arg1_retires_from_arg2 arg2_phenom_arg1 arg2_lesson_from_arg1
arg2_architects_robert_trent_jones_and_arg1 arg2_sensation_arg1
arg2_pros_arg1 arg2_stars_venus_and_arg1 arg2_hall_of_famer_arg1
arg2_superstar_arg1 arg2_legend_arg1 arg2_legends_such_as_arg1
arg2_players_is_arg1 arg2_pro_arg1 arg2_player_was_arg1
arg2_god_arg1 arg2_idol_arg1 arg1_was_born_to_play_arg2
arg2_star_arg1 arg2_hero_arg1 arg2_players_are_arg1
arg1_retired_from_professional_arg2 arg2_legends_as_arg1
arg2_autographed_by_arg1 arg2_champion_arg1

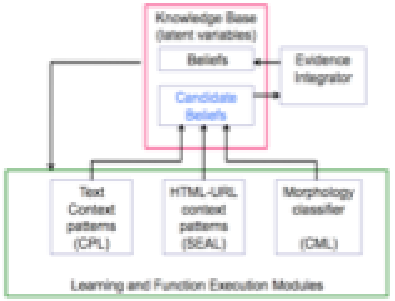


Predicate	Feature	Weight
mountain	LAST=peak	1.791
mountain	LAST=mountain	1.093
mountain	FIRST=mountain	-0.875
musicArtist	LAST=band	1.853
musicArtist	POS=DT_NNS	1.412
musicArtist	POS=DT_JJ_NN	-0.807
newspaper	LAST=sun	1.330
newspaper	LAST=university	-0.318
newspaper	POS=NN_NNS	-0.798
university	LAST=college	2.076
university	PREFIX=uc	1.999
university	LAST=state	1.992
university	LAST=university	1.745
university	FIRST=college	-1.381
visualArtMovement	SUFFIX=ism	1.282
visualArtMovement	PREFIX=journ	-0.234
visualArtMovement	PREFIX=budd	-0.253



Predicate	Web URL	Extraction Template
academicField	http://scholendow.ais.msu.edu/student/ScholSearch.Asp	[X] -
athlete	http://www.quotes-search.com/d_occupation.aspx?o=+athlete	<a href='d_author.aspx?a=[X]'>-
bird	http://www.michaelforsberg.com/stock.html	<option>[X]</option>
bookAuthor	http://lifebehindthecurve.com/	</li> <li>[X] by [Y] &#8211;

Never-Ending Language Learning


Knowledge Base
(latent variables)
Beliefs
Candidate Beliefs
Evidence Integrator
Text Context patterns (CPL)
HTML-URL context patterns (SEAL)
Morphology classifier (CML)
Learning and Function Execution Modules

arg1_was_playing_arg2 arg2_megastar_arg1 arg2_icons_arg1
arg2_player_named_arg1 arg2_prodigy_arg1
arg1_is_the_tiger_woods_of_arg2 arg2_career_of_arg1
arg2_greats_as_arg1 arg1_plays_arg2 arg2_player_is_arg1
arg2_legends_arg1 arg1_announced_his_retirement_from_arg2
arg2_operations_chief_arg1 arg2_player_like_arg1
arg2_and_golfing_personalities_including_arg1 arg2_players_like_arg1
arg2_greats_like_arg1 arg2_players_are_steffi_graf_and_arg1
arg2_great_arg1 arg2_champ_arg1 arg2_greats_such_as_arg1
arg2_professionals_such_as_arg1 arg2_hit_by_arg1 arg2_greats_arg1
arg2_icon_arg1 arg2_stars_like_arg1 arg2_pros_like_arg1
arg1_re
arg2_ar
arg2_pr
arg2_su
arg2_pl
arg2_gc
arg2_st
arg1_retired_from_professional_arg2 arg2_legends_as_arg1
arg2_autographed_by_arg1 arg2_champion_arg1

- Humans never learn in isolation
- We learn effectively from a few examples with the help of the past knowledge.

Predicate	Feature	Weight
mountain	LAST=peak	1.791
		1.093
		-0.875
		1.853
		1.412
		-0.807
		1.330
		-0.318
		-0.798
		2.076
university	PREFIX=uc	1.999
university	LAST=state	1.992
university	LAST=university	1.745
university	FIRST=college	-1.381
visualArtMovement	SUFFIX=ism	1.282
visualArtMovement	PREFIX=journ	-0.234
visualArtMovement	PREFIX=budd	-0.253

Predicate	Web URL	Extraction Template
academicField	http://scholendow.ais.msu.edu/student/ScholSearch.Asp	` [X] -`
athlete	http://www.quotes-search.com/d_occupation.aspx?o=+athlete	`<a href='d_author.aspx?a=[X]'>-`
bird	http://www.michaelforsberg.com/stock.html	`<option>[X]</option>`
bookAuthor	http://lifebehindthecurve.com/	`</li> <li>[X] by [Y] &#8211;`

Computer Reading the Web

1. Classify noun phrases (NP’s) by category
2. Classify NP pairs by relation
3. Discover rules to predict new relation instances

Key Idea 2: Discover New Coupling Constraints

- first order, probabilistic horn clause constraints

0.93 athletePlaysSport(?x,?y) :-athletePlaysForTeam(?x,?z),

teamPlaysSport(?z,?y)

– connects previously uncoupled relation predicates

– infers new beliefs for KB

Example Learned Horn Clauses

0.95 athletePlaysSport(?x,basketball) :- athleteInLeague(?x,NBA)

0.93 athletePlaysSport(?x,?y) :- athletePlaysForTeam(?x,?z)
teamPlaysSport(?z,?y)

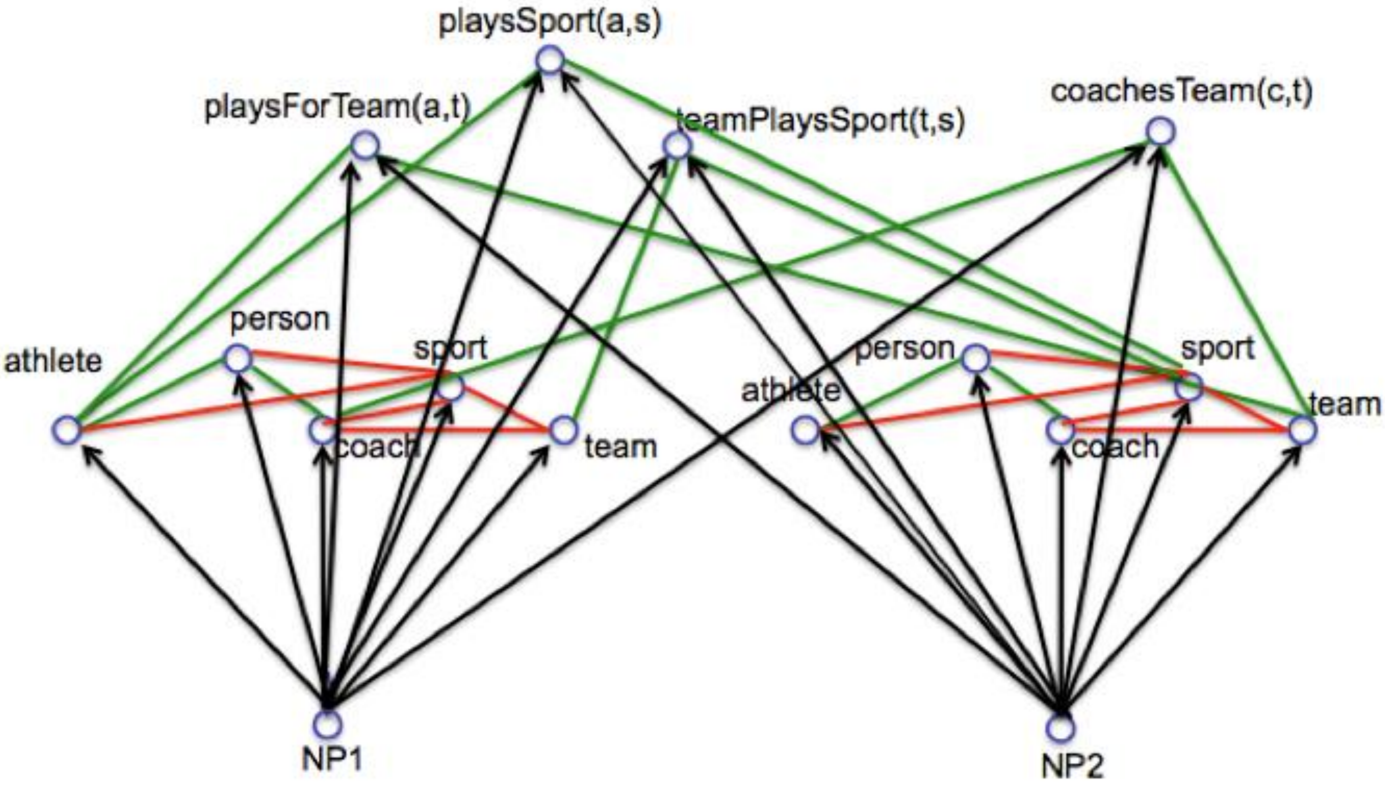
0.91 teamPlaysInLeague(?x,NHL) :- teamWonTrophy(?x,Stanley_Cup)

0.90 athleteInLeague(?x,?y):-athletePlaysForTeam(?x,?z),
teamPlaysInLeague(?z,?y)

0.88 cityInState(?x,?y) :- cityCapitalOfState(?x,?y),
cityInCountry(?y,USA)

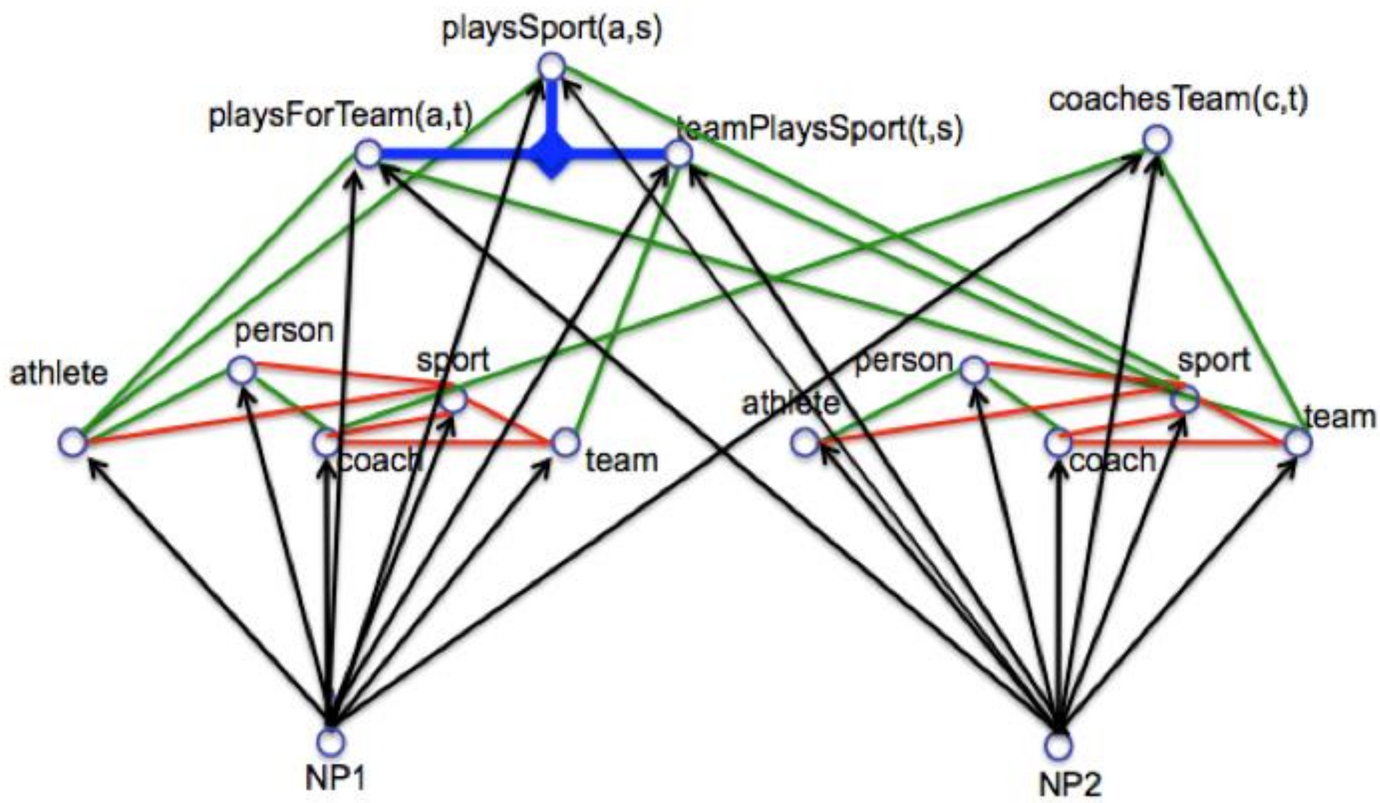
0.62* newspaperInCity(?x,New_York) :- companyEconomicSector(?x,media),
generalizations(?x,blog)

Learned Probabilistic Horn Clause Rules

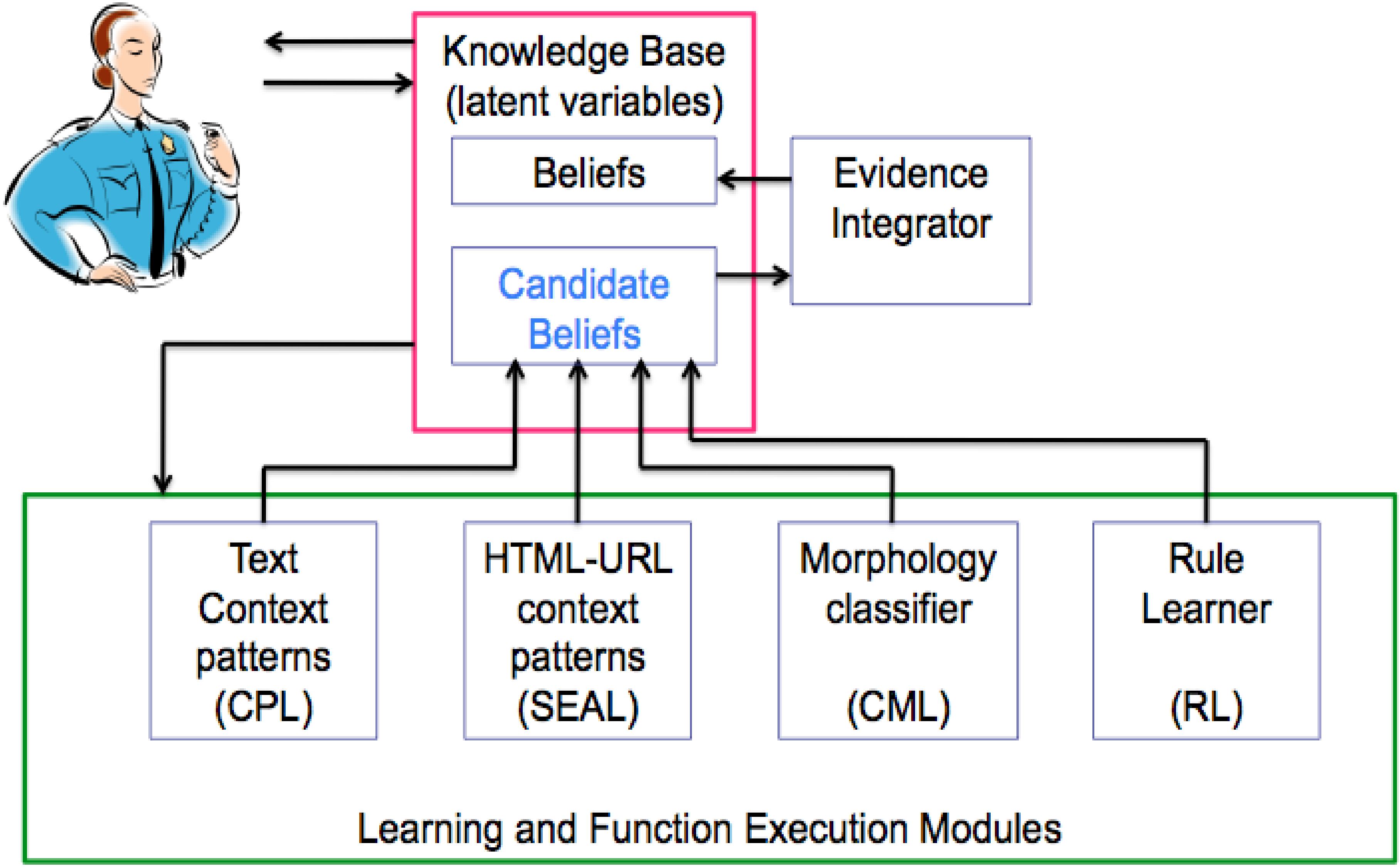


Learned Probabilistic Horn Clause Rules

0.93 playsSport(?x,?y) ← playsForTeam(?x,?z), teamPlaysSport(?z,?y)



NELL Architecture



Knowledge Base (latent variables)

Beliefs

Candidate Beliefs

Evidence Integrator

Text Context patterns (CPL)

HTML-URL context patterns (SEAL)

Morphology classifier (CML)

Rule Learner (RL)

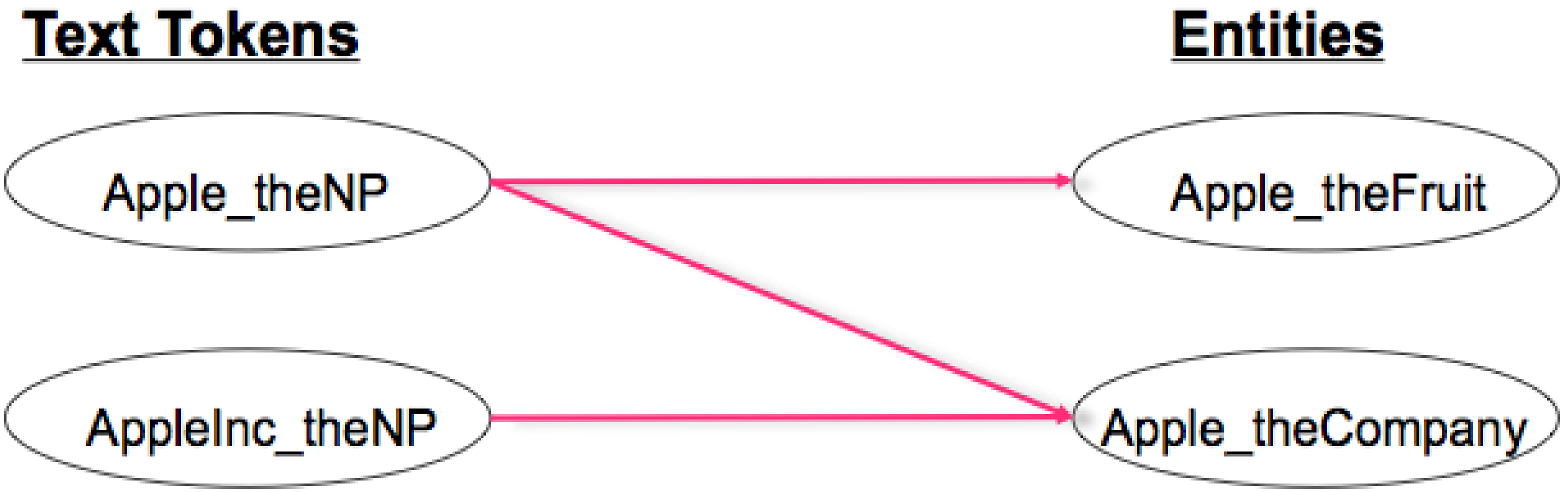
Learning and Function Execution Modules

Computer Reading the Web

1. Classify noun phrases (NP's) by category
2. Classify NP pairs by relation
3. Discover rules to predict new relation instances
4. Learn which NP's (co)refer to which latent concepts

Distinguish Text Tokens from Entities

[Jayant Krishnamurthy]



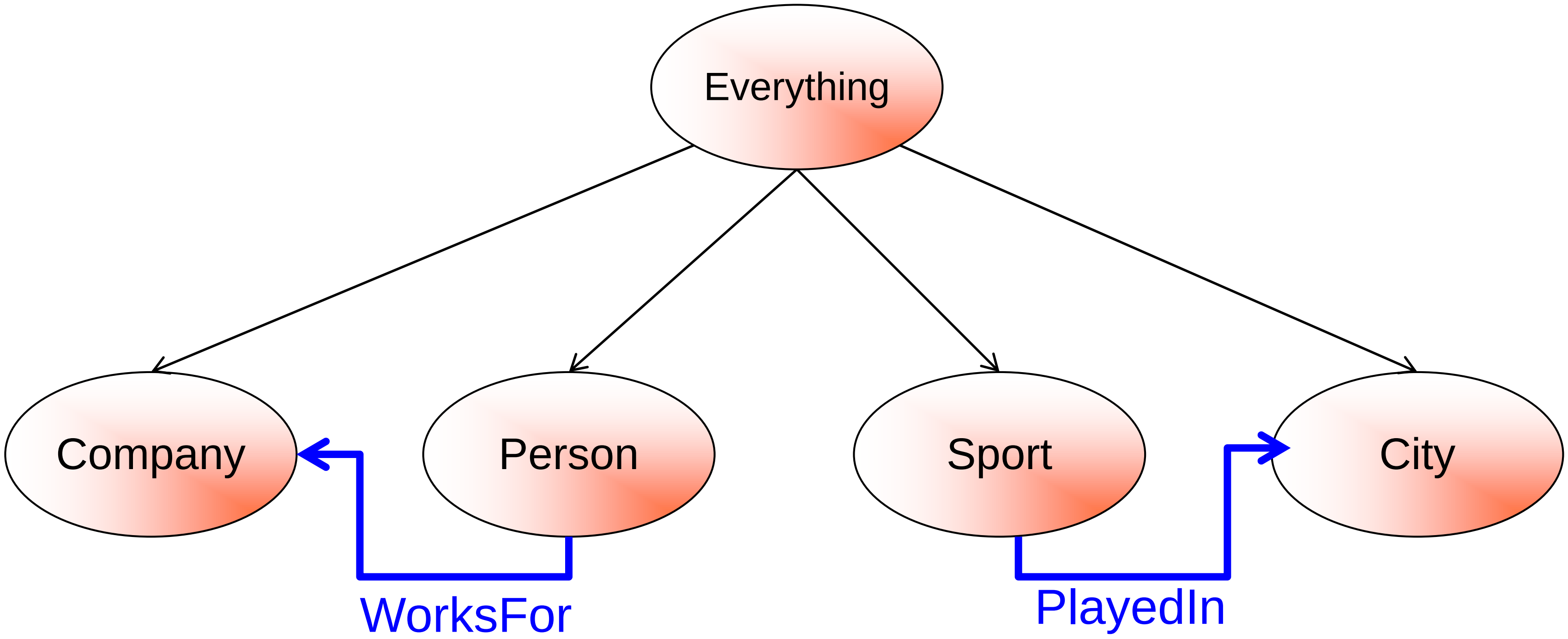
Coreference Resolution:

- Co-train classifier to predict coreference as f(string similarity, extracted beliefs)
- Small amount of supervision: ~10 labeled coreference decisions
- Cluster tokens using f as similarity measure

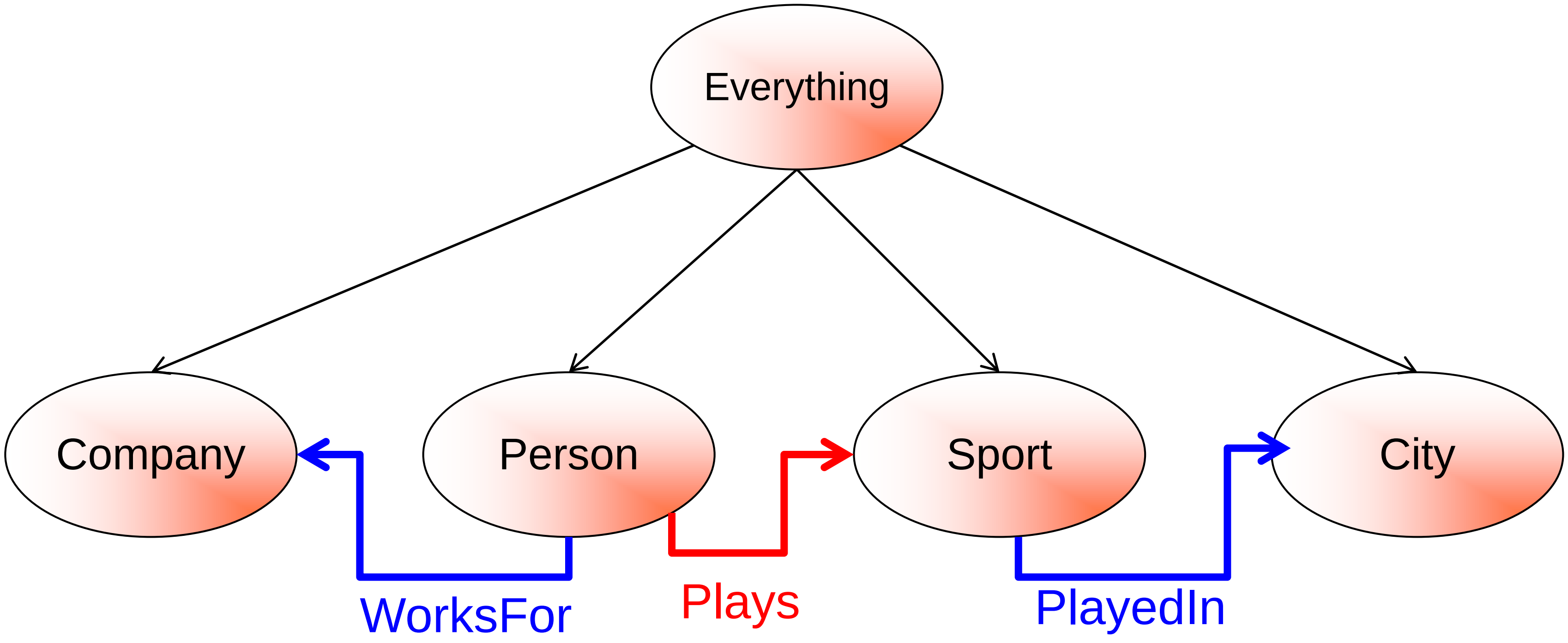
Computer Reading the Web

1. Classify noun phrases (NP's) by category
2. Classify NP pairs by relation
3. Discover rules to predict new relation instances
4. Learn which NP's (co)refer to which latent concepts
5. Discover new relations to extend ontology

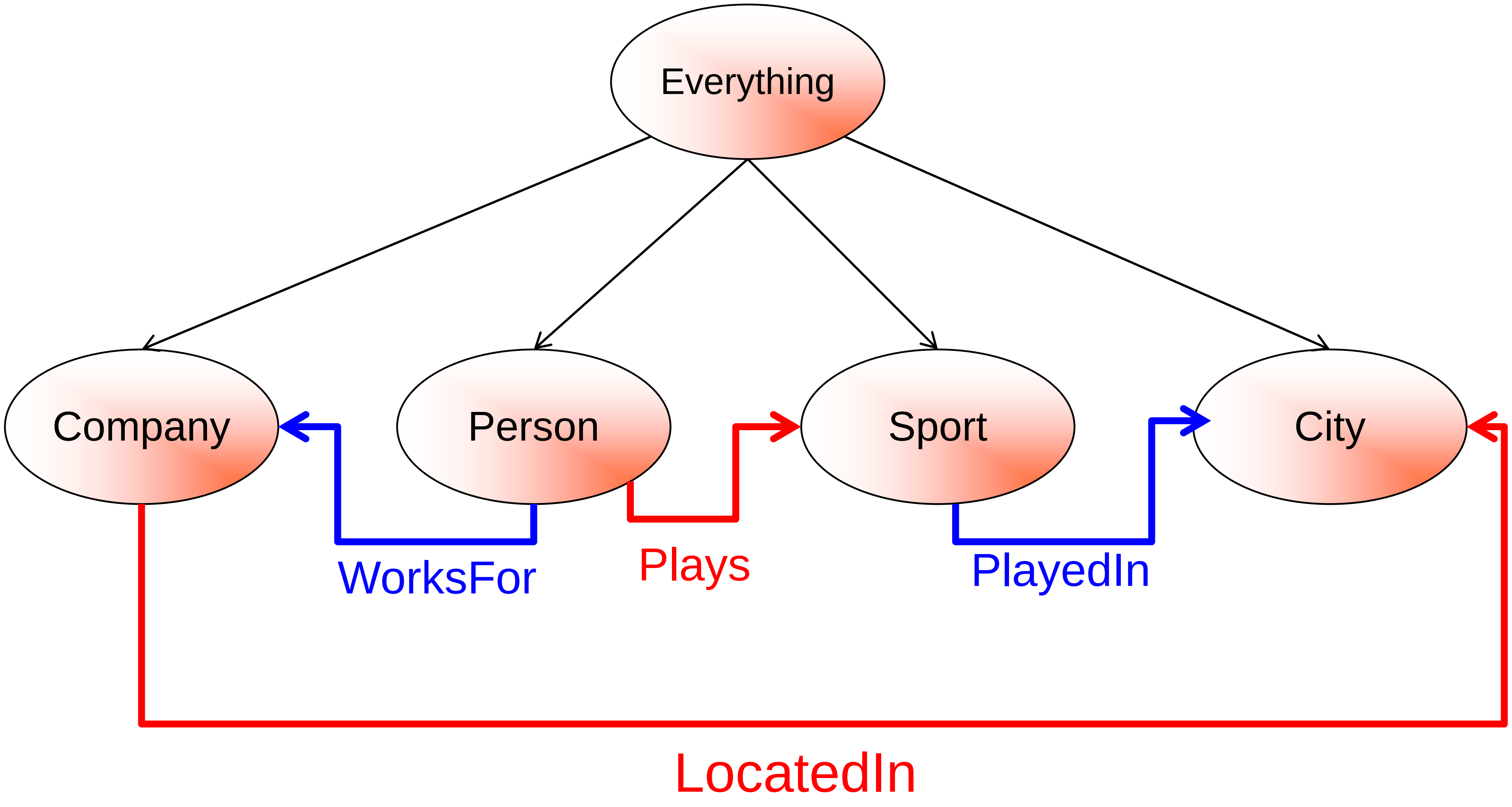
OntExt (Ontology Extension)



OntExt (Ontology Extension)



OntExt (Ontology Extension)



Prophet

- Mining the Graph representing NELL’s KB to:

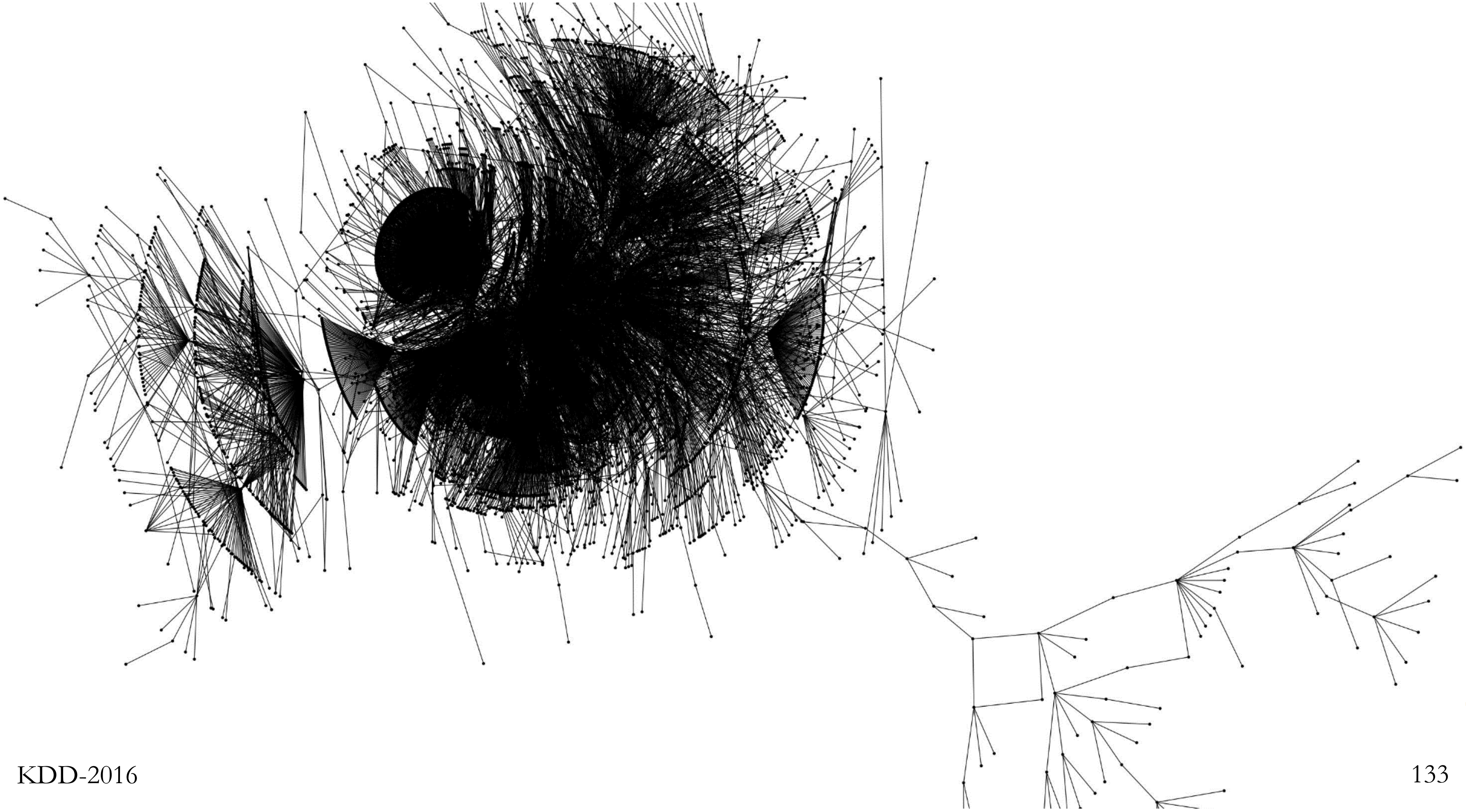
1. Extend the KB by predicting new relations (edges)that might exist between pairs of nodes;
2. Induce inference rules;
3. Identify misplaced edges which can be used by NELL as hints to identify wrong connections between nodes (wrong fats);

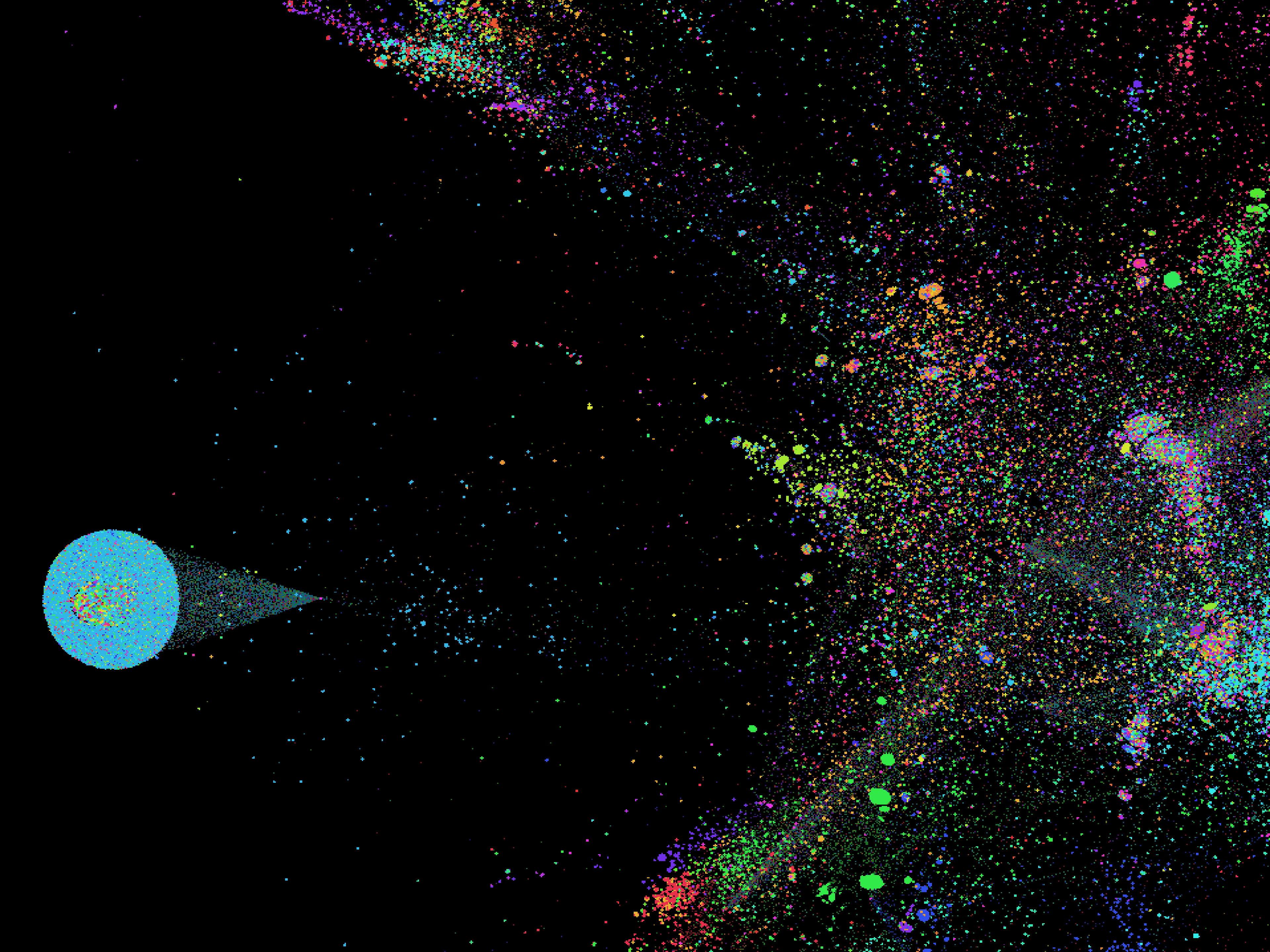
-

Prophet

Appel and Hruschka, 2011

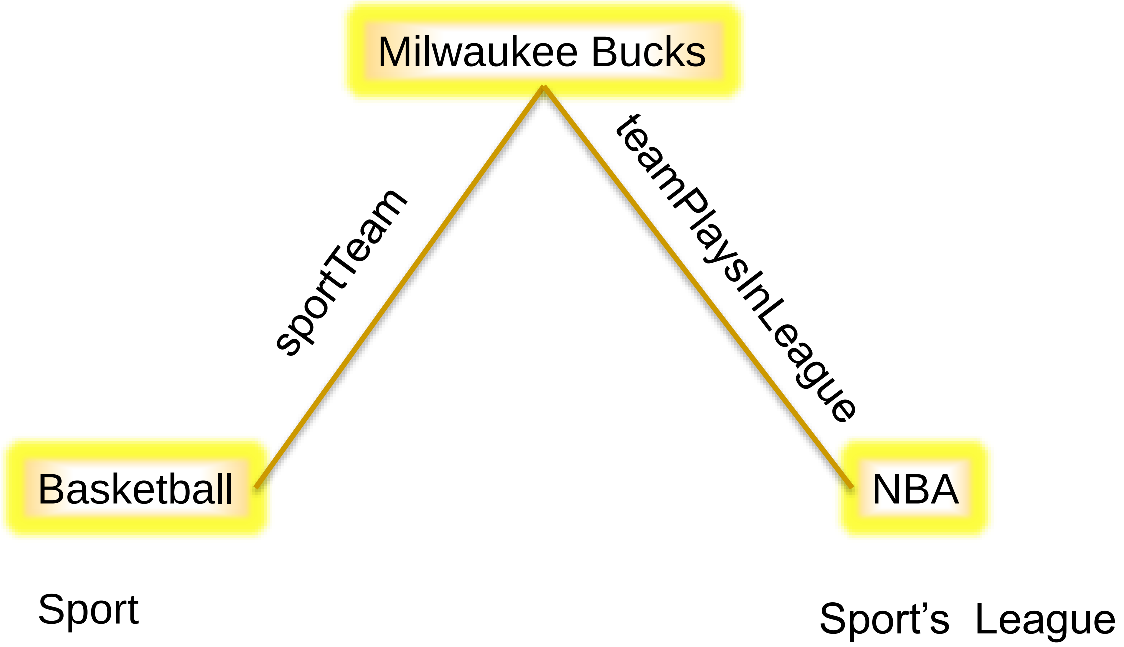
- Find open triangles in the Graph





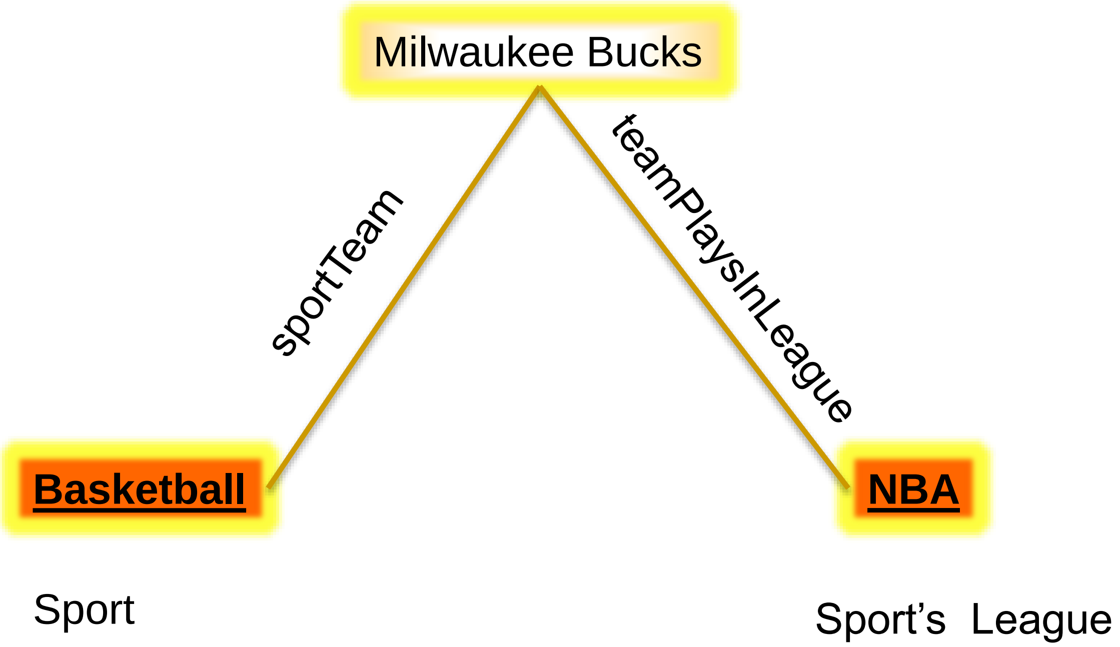
Prophet

- open triangles



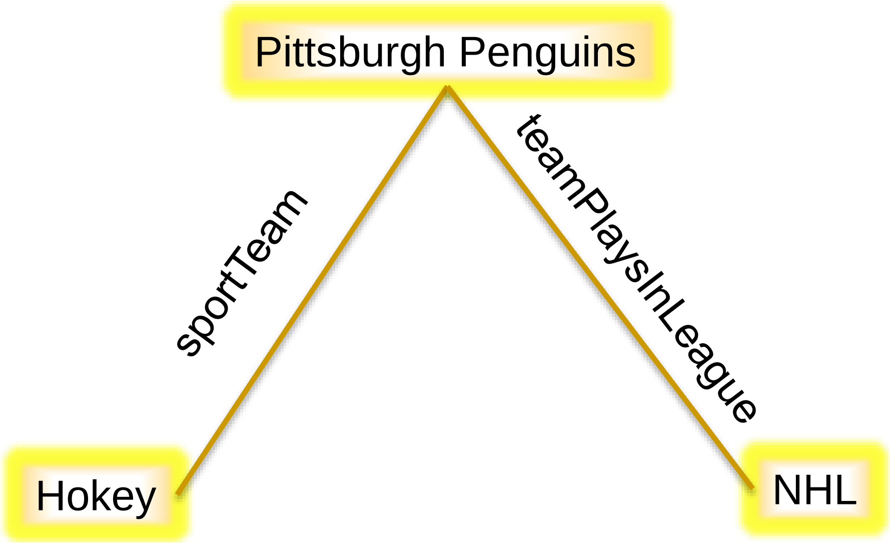
Prophet

- open triangles



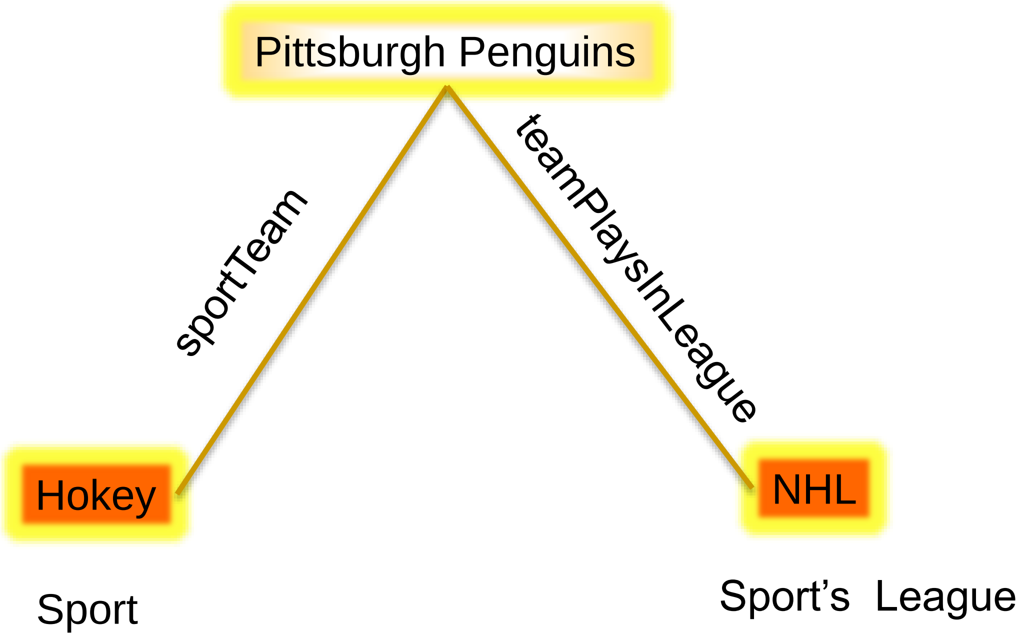
Prophet

- open triangles



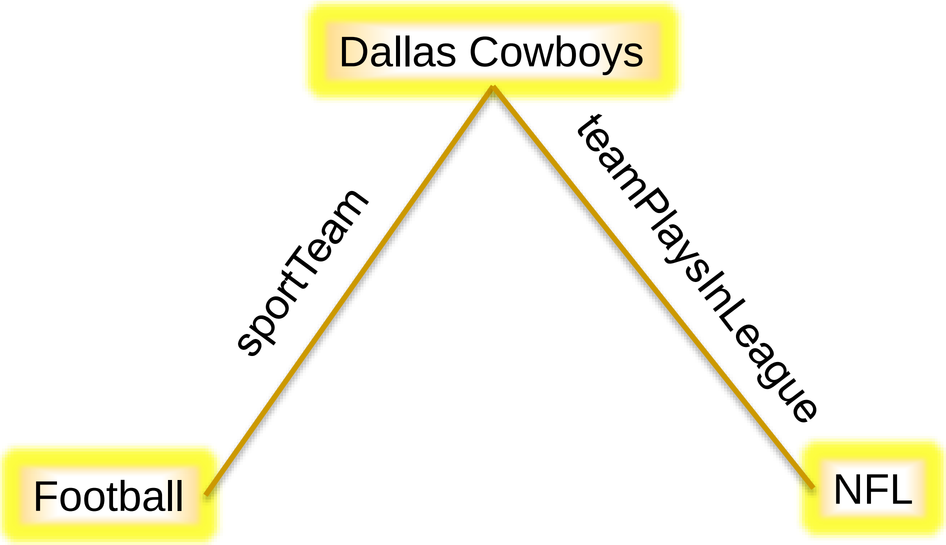
Prophet

- open triangles



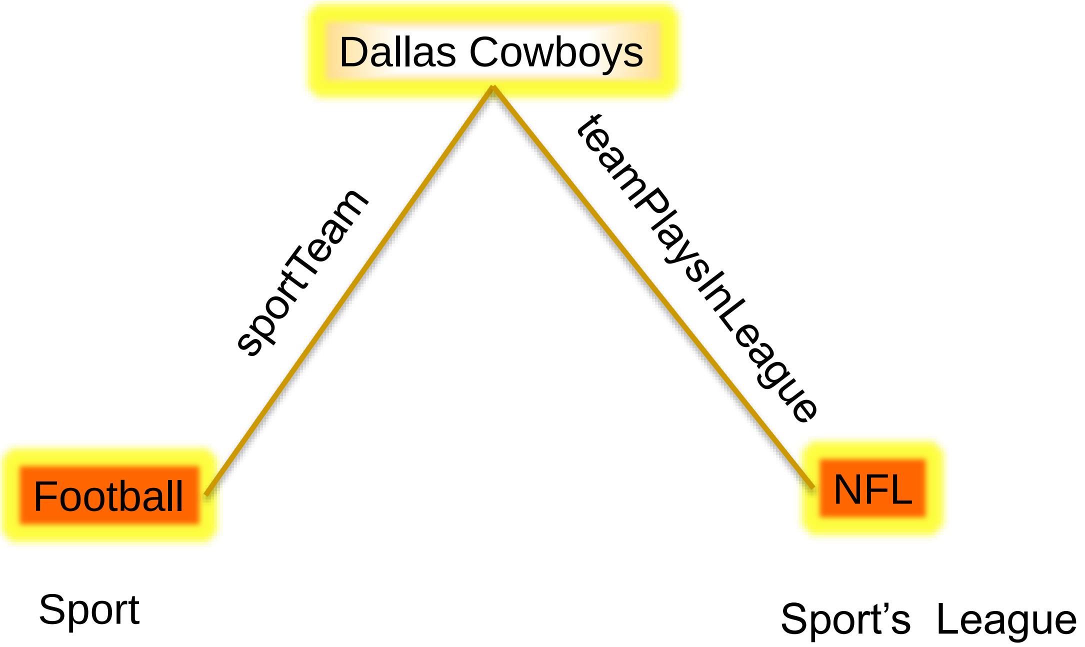
Prophet

- open triangles



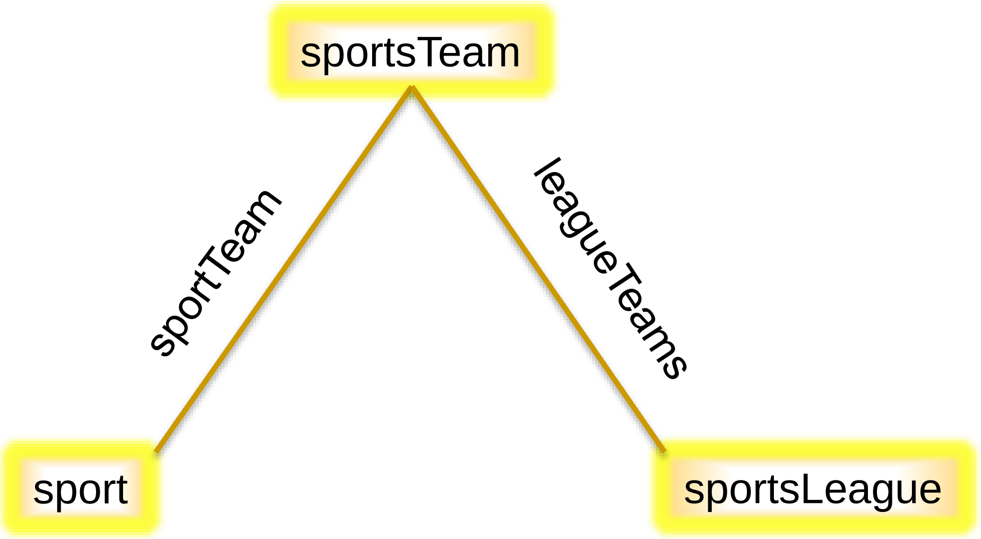
Prophet

- open triangles



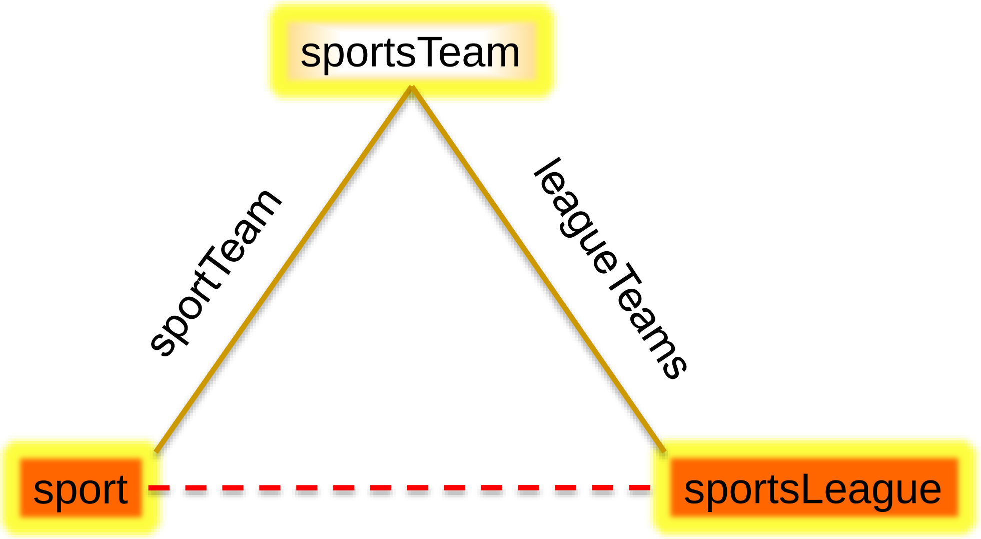
Prophet

- open triangles



Prophet

- open triangles

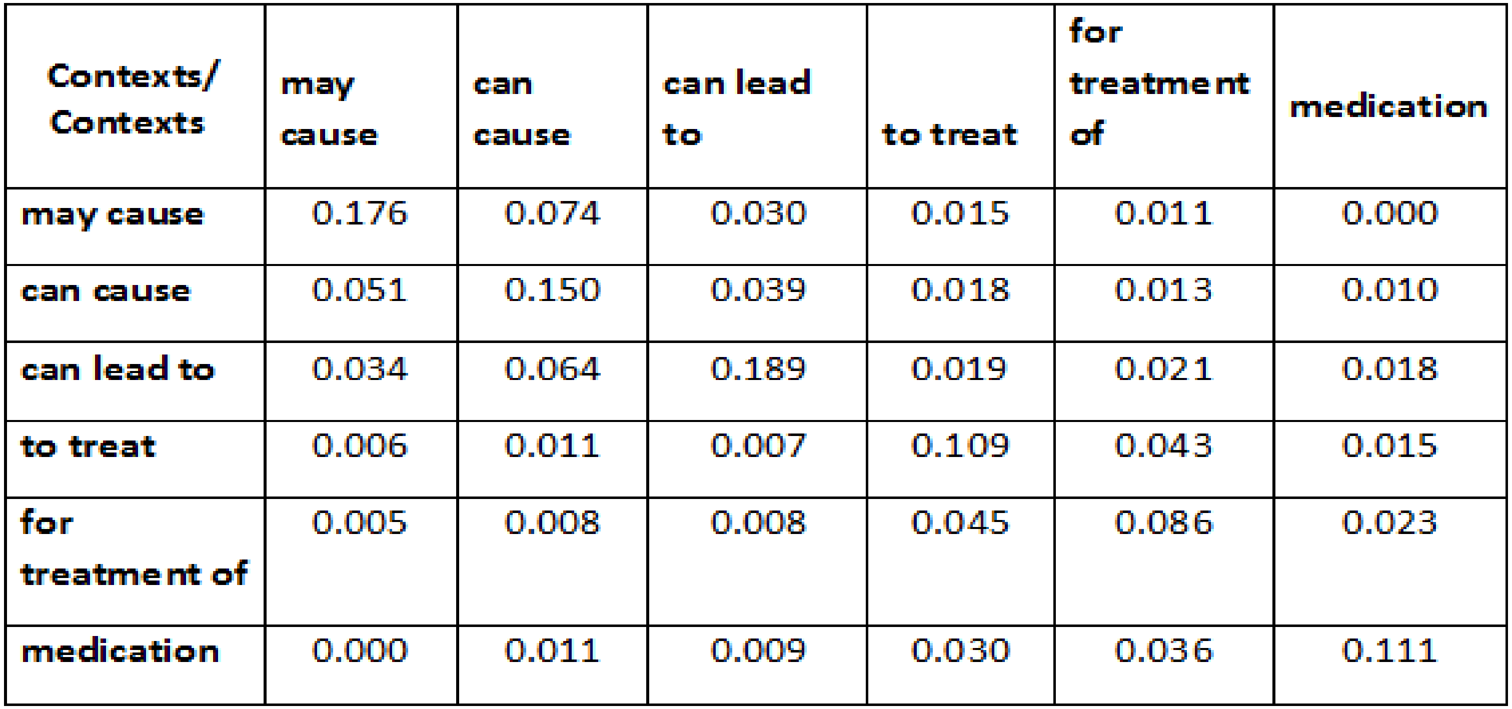


Prophet

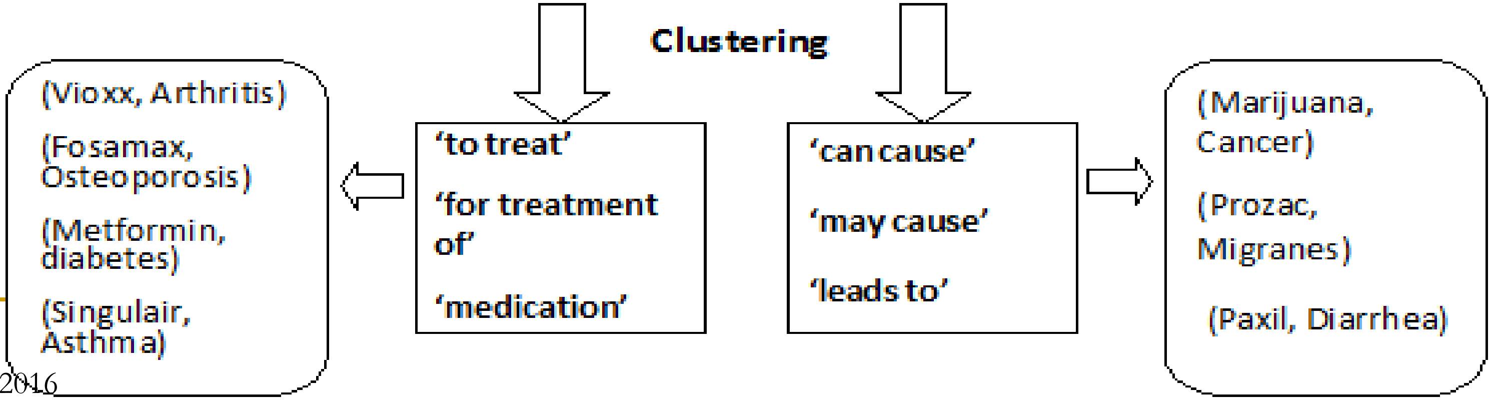
- open triangles
- Name the new relation based on a big textual corpus

OntExt

Mohamed, Hruschka and Mitchell, 2011



Contexts/ Contexts	may cause	can cause	can lead to	to treat	for treatment of	medication
may cause	0.176	0.074	0.030	0.015	0.011	0.000
can cause	0.051	0.150	0.039	0.018	0.013	0.010
can lead to	0.034	0.064	0.189	0.019	0.021	0.018
to treat	0.006	0.011	0.007	0.109	0.043	0.015
for treatment of	0.005	0.008	0.008	0.045	0.086	0.023
medication	0.000	0.011	0.009	0.030	0.036	0.111



Clustering

(Vioxx, Arthritis)

(Fosamax, Osteoporosis)

(Metformin, diabetes)

(Singulair, Asthma)

'to treat'

'for treatment of'

'medication'

'can cause'

'may cause'

'leads to'

(Marijuana, Cancer)

(Prozac, Migranes)

(Paxil, Diarrhea)

NELL: sample of self-added relations

- athleteWonAward
- animalEatsFood
- languageTaughtInCity
- clothingMadeFromPlant
- beverageServedWithFood
- fishServedWithFood
- athleteBeatAthlete
- athleteInjuredBodyPart
- arthropodFeedsOnInsect
- animalEatsVegetable
- plantRepresentsEmotion
- foodDecreasesRiskOfDisease
- clothingGoesWithClothing
- bacteriaCausesPhysCondition
- buildingMadeOfMaterial
- emotionAssociatedWithDisease
- foodCanCauseDisease
- agriculturalProductAttractsInsect
- arteryArisesFromArtery
- countryHasSportsFans
- bakedGoodServedWithBeverage
- beverageContainsProtein
- animalCanDevelopDisease
- beverageMadeFromBeverage

Computer Reading the Web

1. Classify noun phrases (NP's) by category
2. Classify NP pairs by relation
3. Discover rules to predict new relation instances
4. Learn which NP's (co)refer to which latent concepts
5. Discover new relations to extend ontology
6. Learn to infer relation instances via targeted random walks (PRA)

CityLocatedInCountry(Pittsburgh) = ?

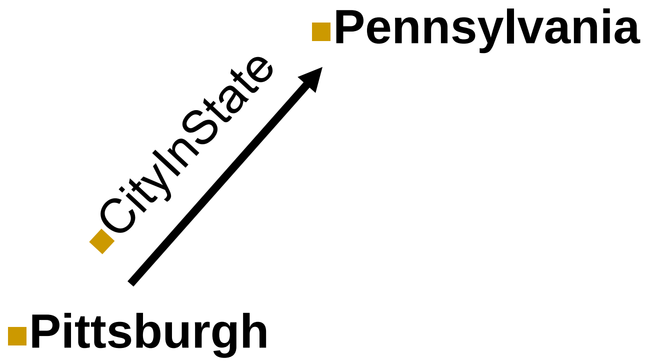
[Lao, Mitchell, Cohen, *EMNLP* 2011]

Pittsburgh

Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry		0.32

CityLocatedInCountry(Pittsburgh) = ?

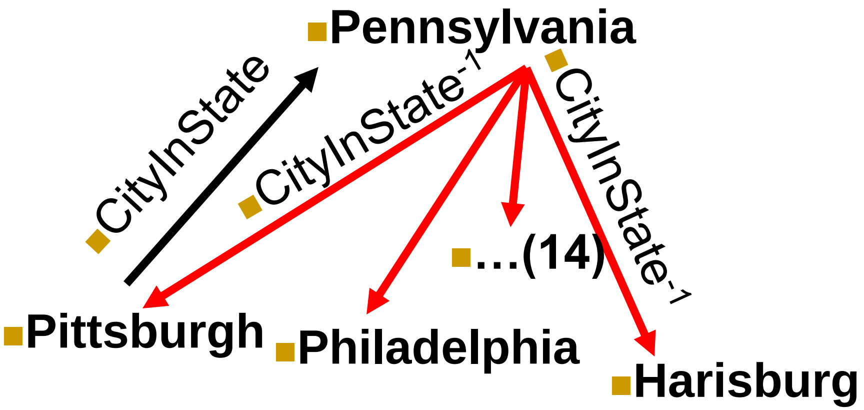
[Lao, Mitchell, Cohen, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry		0.32

CityLocatedInCountry(Pittsburgh) = ?

[Lao, Mitchell, Cohen, *EMNLP* 2011]



Pennsylvania

CityInState

CityInState^{-1}

CityInState^{-1}

…(14)

Pittsburgh

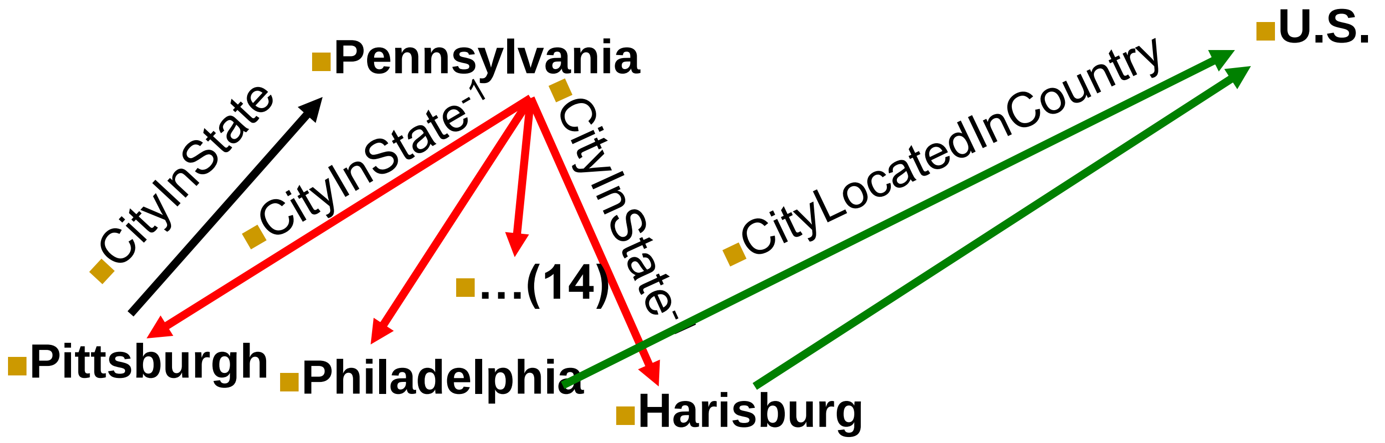
Philadelphia

Harisburg

Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry		0.32

CityLocatedInCountry(Pittsburgh) = ?

[Lao, Mitchell, Cohen, *EMNLP* 2011]



U.S.

Pennsylvania

CityInState

CityInState^{-1}

CityInState-

…(14)

CityLocatedInCountry

Pittsburgh

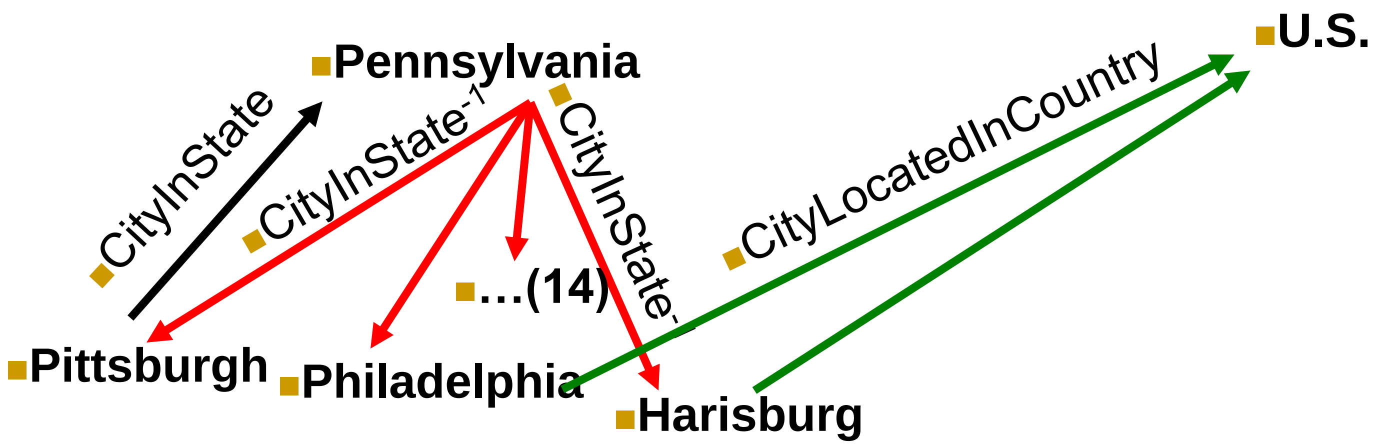
Philadelphia

Harisburg

Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry		0.32

CityLocatedInCountry(Pittsburgh) = ?

[Lao, Mitchell, Cohen, *EMNLP* 2011]

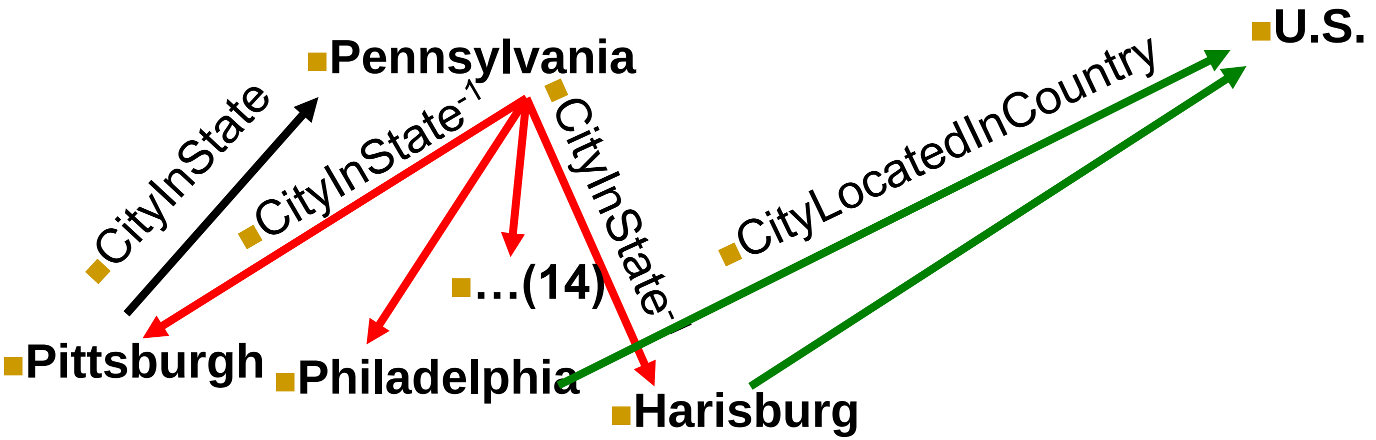


Pr(U.S. | Pittsburgh, TypedPath)

Feature = Typed Path	Feature Value	Logistic Regresssion Weight
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32

CityLocatedInCountry(Pittsburgh) = ?

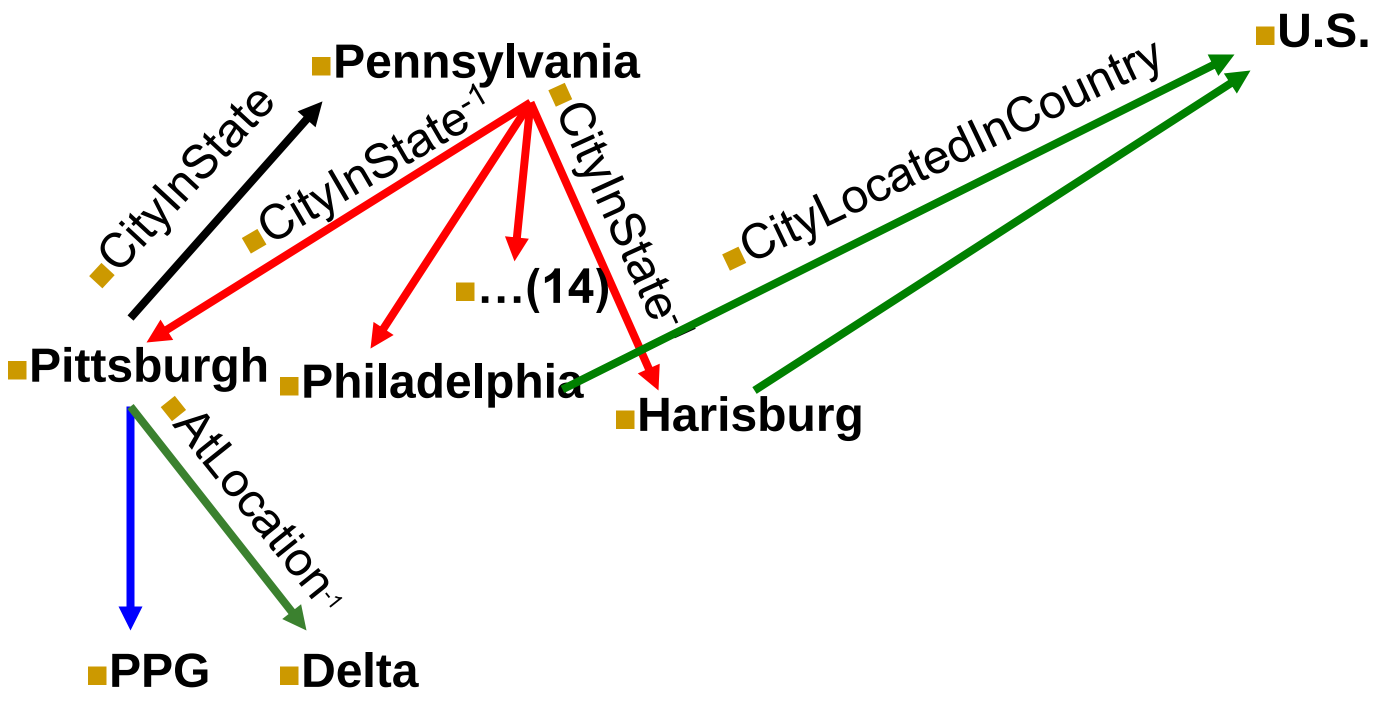
[Lao, Mitchell, Cohen, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32
AtLocation^{-1}, AtLocation, CityLocatedInCountry		0.20

CityLocatedInCountry(Pittsburgh) = ?

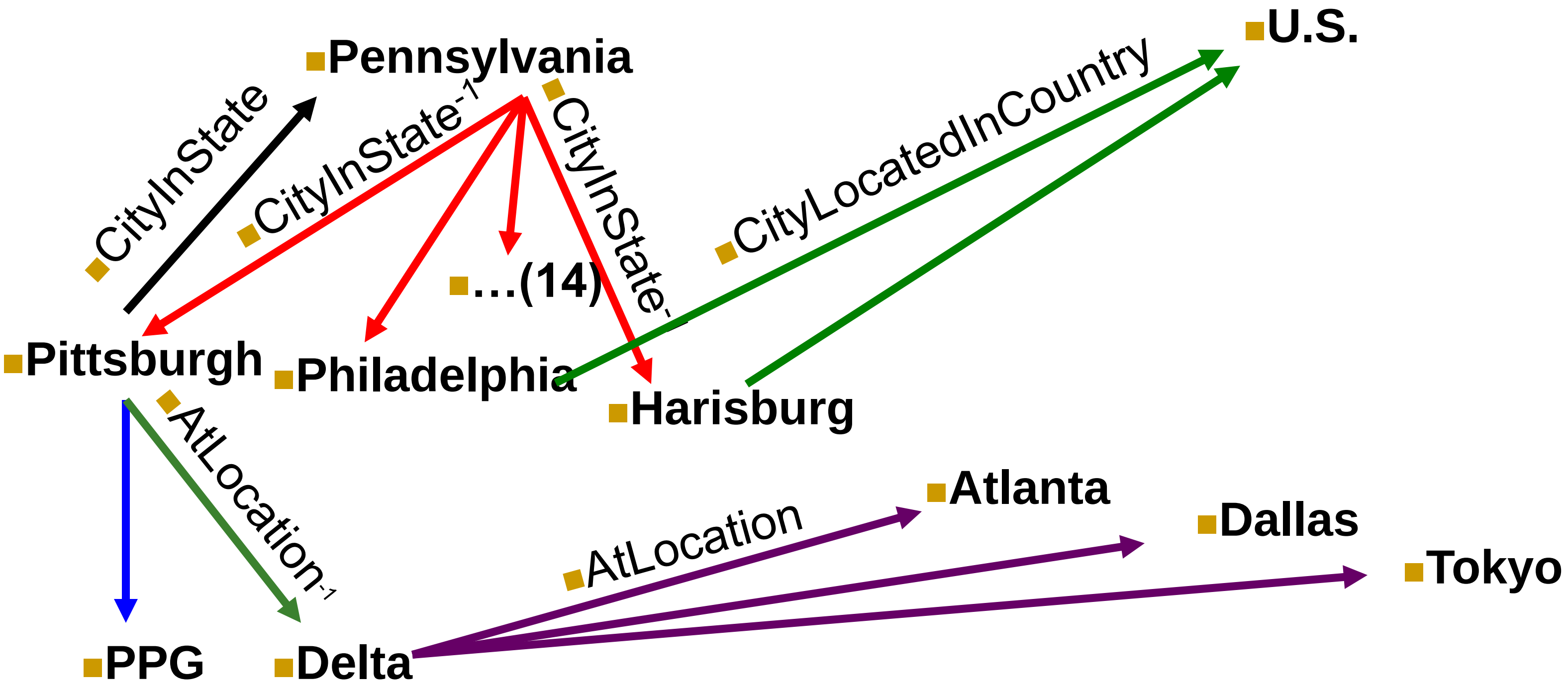
[Lao, Mitchell, Cohen, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32
AtLocation^{-1}, AtLocation, CityLocatedInCountry		0.20

CityLocatedInCountry(Pittsburgh) = ?

[Lao, Mitchell, Cohen, *EMNLP* 2011]



U.S.

Pennsylvania

CityInState

CityInState^{-1}

CityInState^{-1}

CityLocatedInCountry

…(14)

Pittsburgh

Philadelphia

Harisburg

AtLocation^{-1}

AtLocation

Atlanta

Dallas

Tokyo

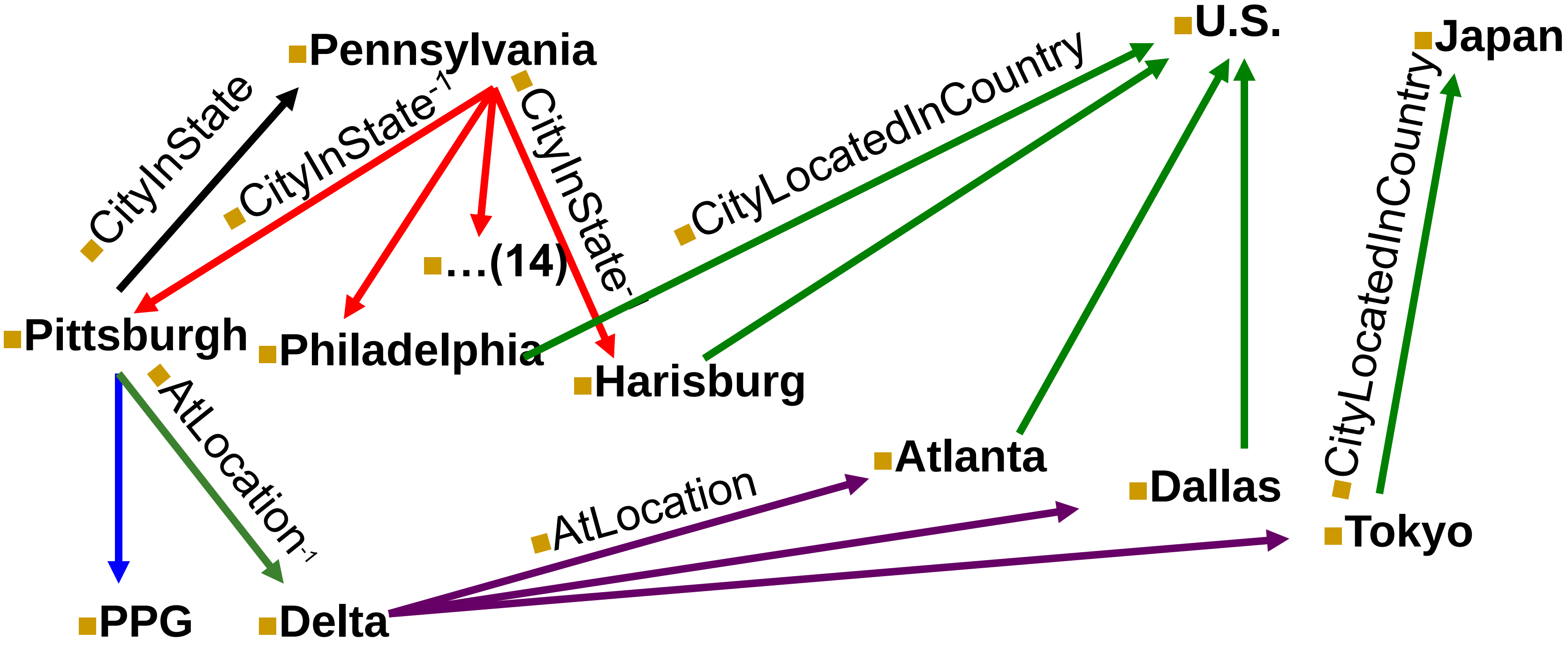
PPG

Delta

Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32
AtLocation^{-1}, AtLocation, CityLocatedInCountry		0.20

CityLocatedInCountry(Pittsburgh) = ?

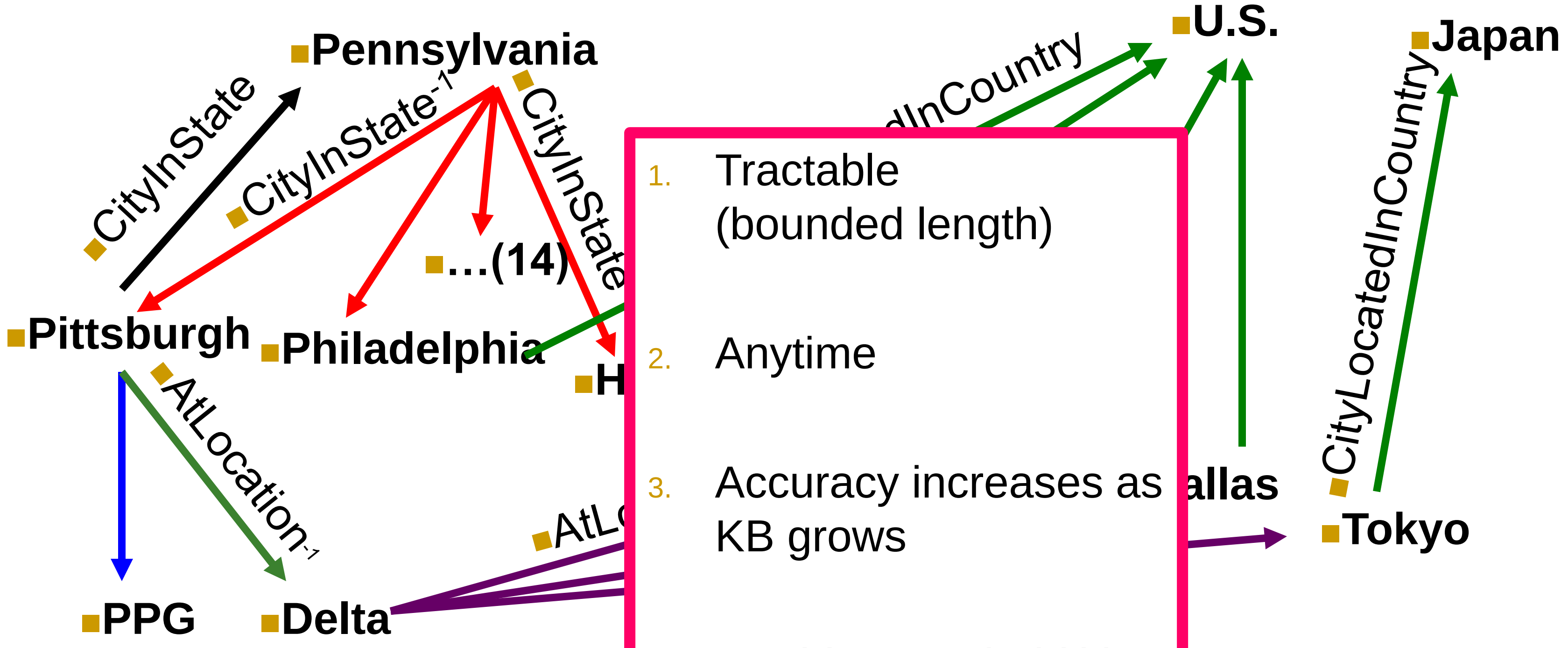
[Lao, Mitchell, Cohen, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate^{-1}, CityLocatedInCountry	0.8	0.32
AtLocation^{-1}, AtLocation, CityLocatedInCountry	0.6	0.20

CityLocatedInCountry(Pittsburgh) = ?

[Lao, Mitchell, Cohen, *EMNLP* 2011]



1. Tractable (bounded length)
2. Anytime
3. Accuracy increases as KB grows
4. combines probabilities from different horn clauses

Feature = Typed Path	**Feature Value**	**Logistic Regresssion Weight**
CityInState, CityInstate-1, CityLocatedInCountry	0.8	0.32
AtLocation-1, AtLocation, CityLocatedInCountry	0.6	0.20
…	…	…

CityLocatedInCountry(Pittsburgh) = U.S. p=0.58

Random walk inference: learned rules

CityLocatedInCountry(*city, country*):

8.04 cityliesonriver, cityliesonriver^{-1}, citylocatedincountry

5.42 hasofficeincity^{-1}, hasofficeincity, citylocatedincountry

4.98 cityalsoknownas, cityalsoknownas, citylocatedincountry

2.85 citycapitalofcountry,citylocatedincountry^{-1},citylocatedincountry

2.29 agentactsinlocation^{-1}, agentactsinlocation, citylocatedincountry

1.22 statehascapital^{-1}, statelocatedincountry

0.66 citycapitalofcountry

.

.

.

- 7 of the 2985 learned rules for CityLocatedInCountry

Key Idea 4: Cumulative, Staged Learning

Learning X improves ability to learn Y

1. Classify noun phrases (NP's) by category
2. Classify NP pairs by relation
3. Discover rules to predict new relation instances
4. Learn which NP's (co)refer to which latent concepts
5. Discover new relations to extend ontology
6. Learn to infer relation instances via targeted random walks (PRA)
7. Vision: connect NELL and NEIL
8. Mutilingual NELL (Portuguese)
9. CrossLingual NELL
10. Learn to microread single sentences
11. Self reflection, self-directed learning
12. Goal-driven reading: predict, then read to corroborate/correct
13. Make NELL learn by conversation (e.g, Twitter)
14. Add a robot body, or mobile phone body, to NELL

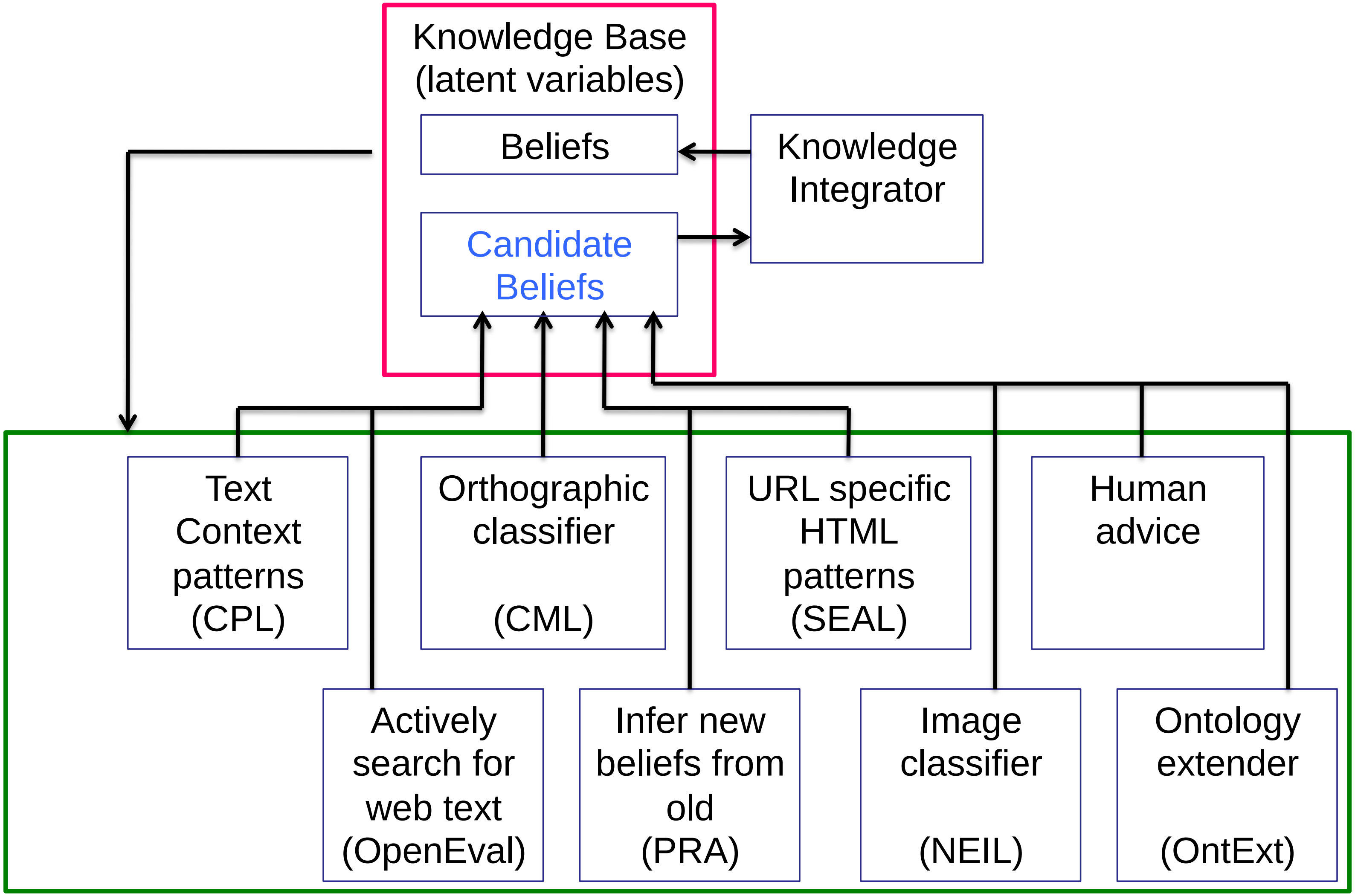
Key Idea 4: Cumulative, Staged Learning

Learning X improves ability to learn Y

1. Classify noun phrases (NP's) by category
2. Classify NP pairs by relation
3. Discover rules to predict new relation instances
4. Learn which NP's (co)refer to which latent concepts
5. Discover new relations to extend ontology
6. Learn to infer relation instances via targeted random walks
7. Vision: connect NELL and NEIL
8. Mutilingual NELL (Portuguese)
9. CrossLingual NELL
10. Learn to microread single sentences
11. Self reflection, self-directed learning
12. Goal-driven reading: predict, then read to corroborate/correct
13. Make NELL learn by conversation (e.g, Twitter)
14. Add a robot body, or mobile phone body, to NELL

NELL is here

NELL Architecture



Knowledge Base (latent variables)

Beliefs

Candidate Beliefs

Knowledge Integrator

Text Context patterns (CPL)

Orthographic classifier (CML)

URL specific HTML patterns (SEAL)

Human advice

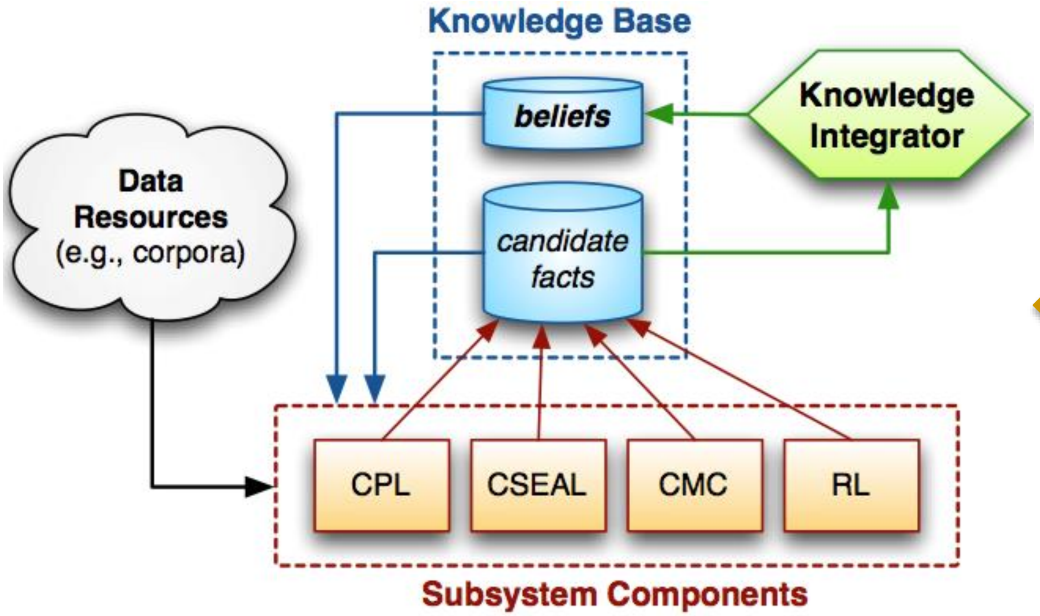
Actively search for web text (OpenEval)

Infer new beliefs from old (PRA)

Image classifier (NEIL)

Ontology extender (OntExt)

Conversing Learning

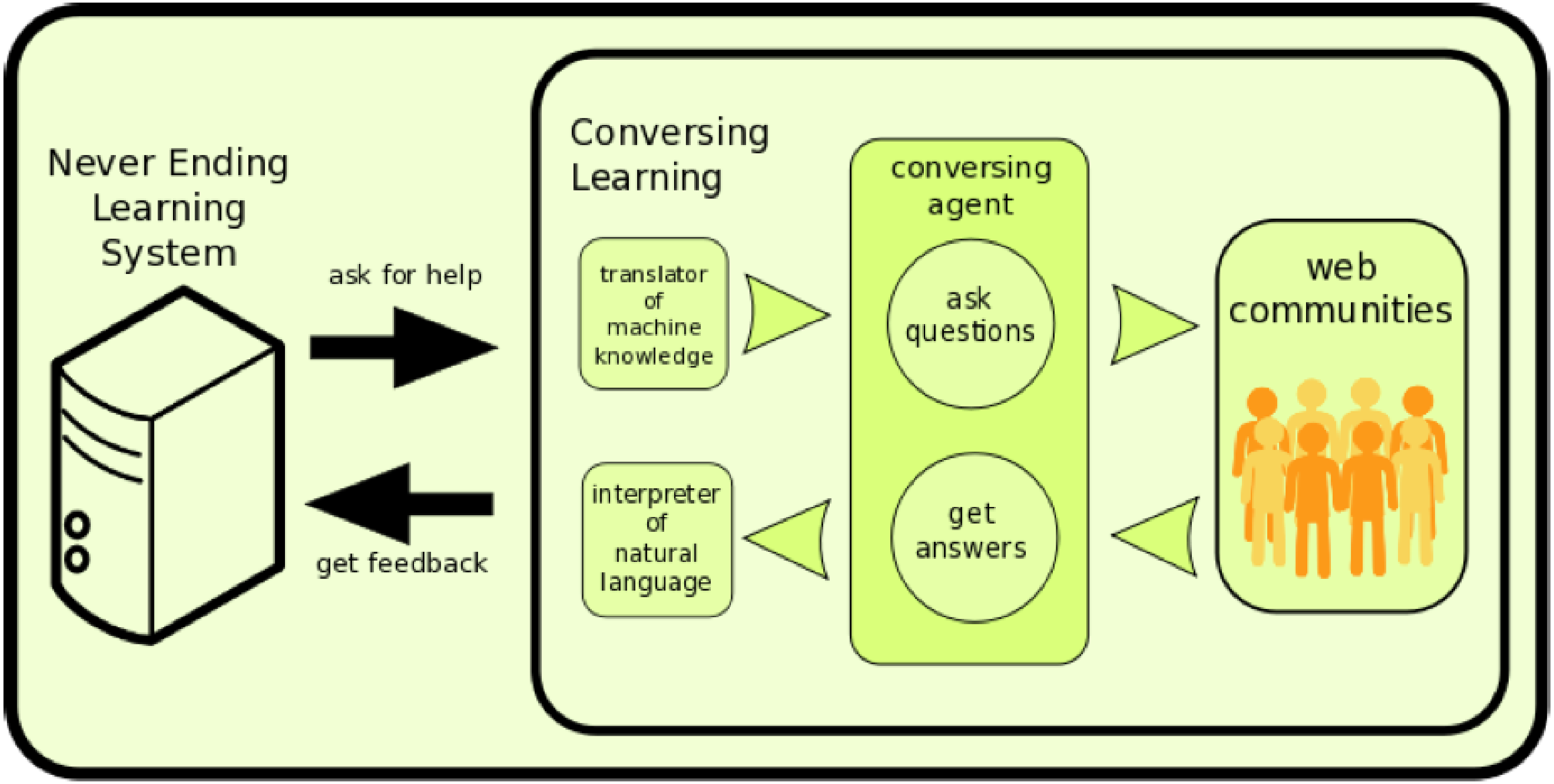


Conversing Learning

- Help to supervise NELL by automatically asking questions on Web Communities

Conversing Learning

- Help to supervise NELL by automatically asking questions on Web Communities

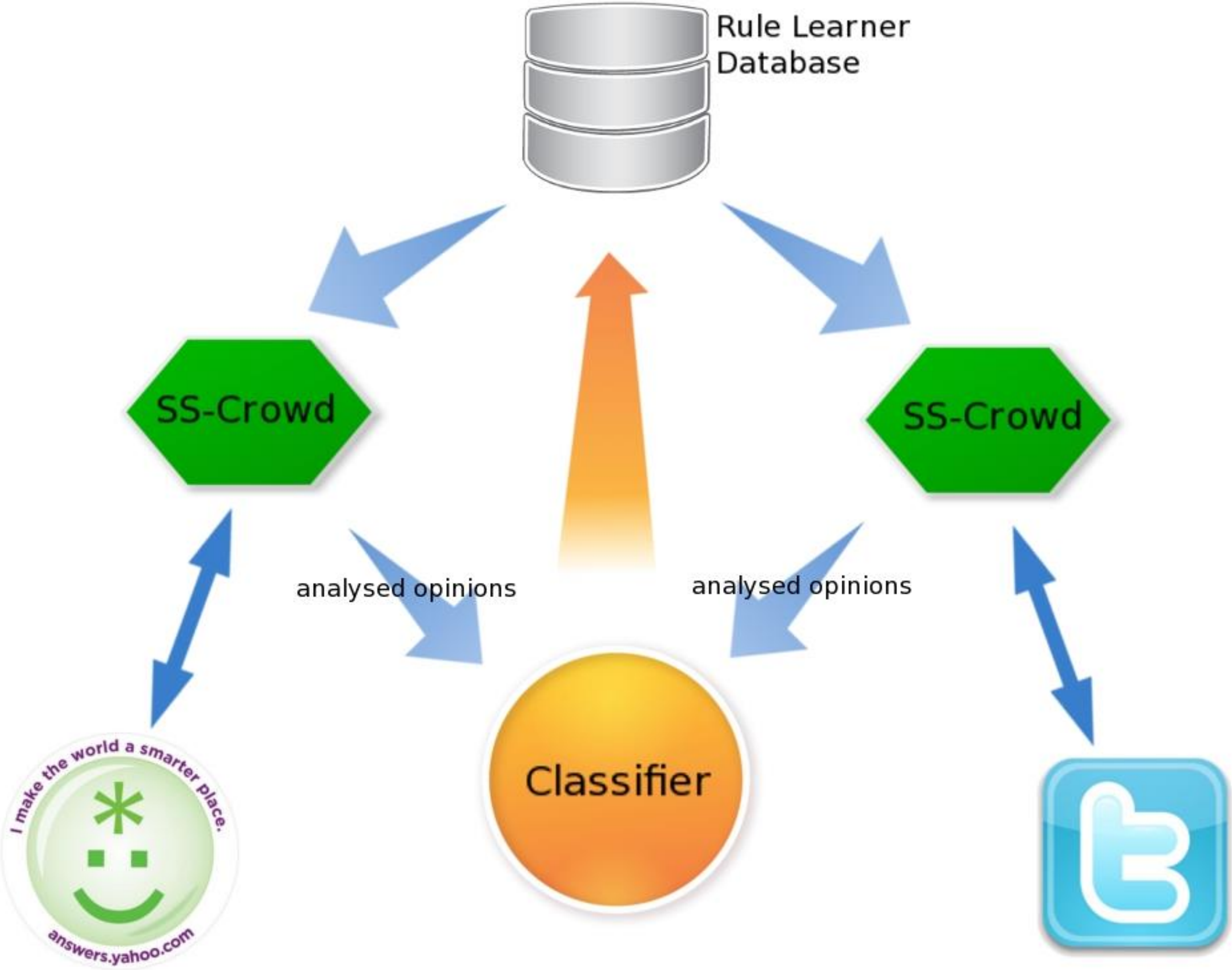


Conversing Learning

- Uses an agent (SS-Crowd) capable of:
  - building questions;
  - Posting questions in Web communities;
  - Fetch answers;
  - Understand the answers;
  - Decide on how much to believe on the answers

Conversing Learning

Pedro & Hruschka



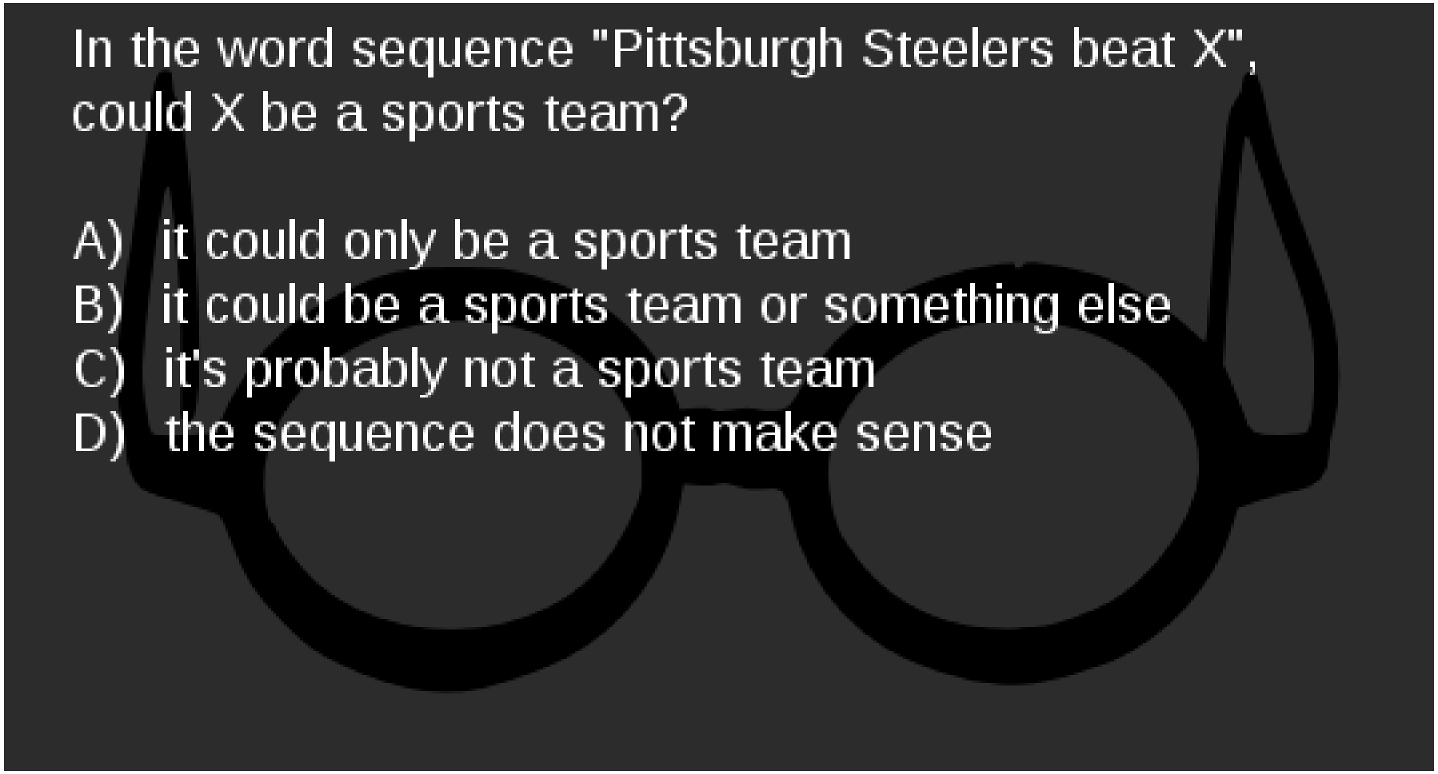
Conversing Learning

- Question: (Yes or No?) If athlete Z is member of team X and athlete Z plays in league Y, then team X plays in league Y.

- Twitter answers sample:
- **No. (Z in X) ∧ (Z in Y) → (X in Y)**

- Yahoo! Answers sample:
- NO, Not in EVERY case. Athlete Z could be a member of football team X and he could also play in his pub's Friday nights dart team. The Dart team could play in league Y (and Z therefore by definition plays in league Y). This does not mean that the football team plays in the darts league!

Conversing Learning



In the word sequence "Pittsburgh Steelers beat X", could X be a sports team?

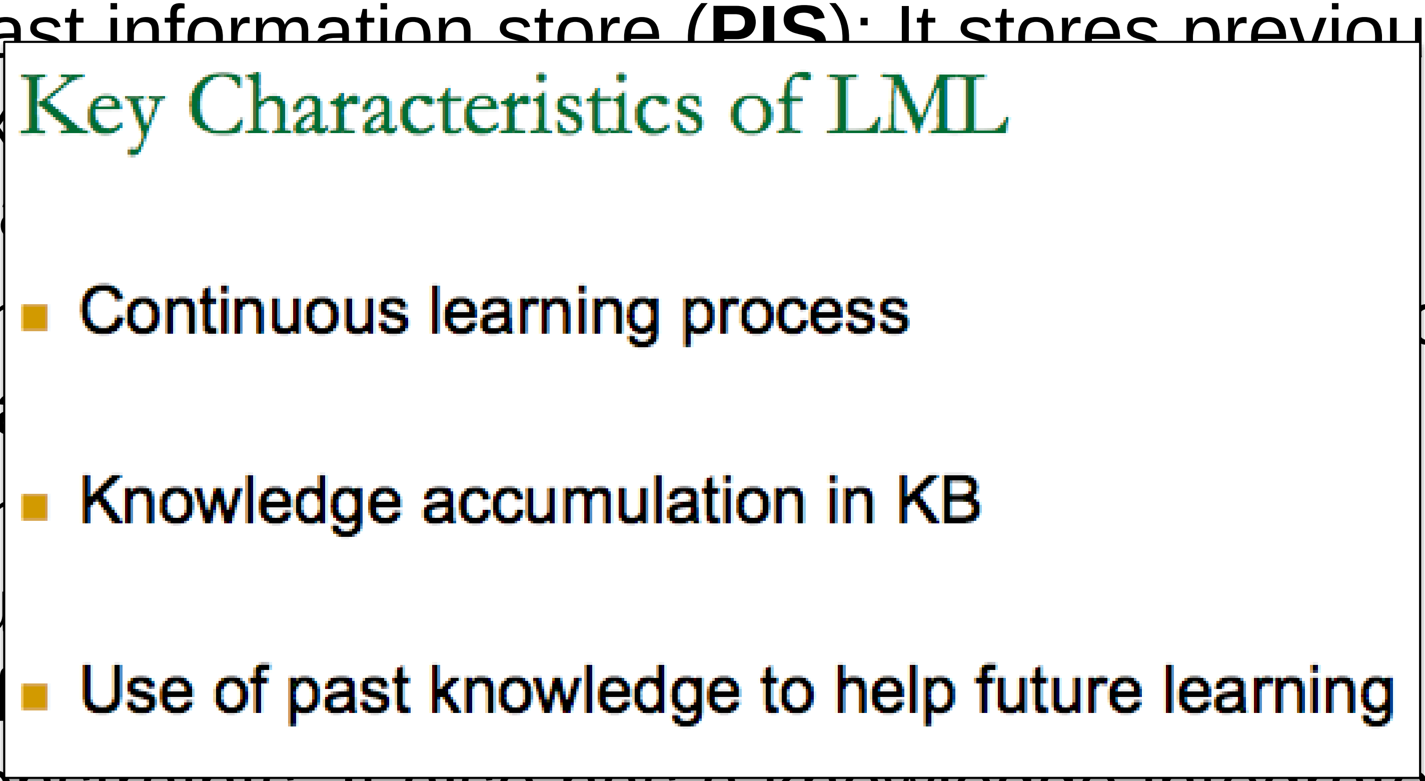
A) it could only be a sports team
B) it could be a sports team or something else
C) it's probably not a sports team
D) the sequence does not make sense

Lifelong Learning components

- Past information store (**PIS**): It stores previously extracted results, phrasings, morphological features, and web page structures.
- Knowledge reasoner (**KR**): Path Ranking Algorithm PRA.
- Knowledge-based learner (**KBL**): Semi-supervised learning using initial and new information in PIS with the help of coupling constraints. It also has a knowledge integrator.

Lifelong Learning components

- Past information store (**PIS**): It stores previously ex fe
- Kn onal lea
- Kn su inf constraints. It also has a knowledge integrator.



Key Characteristics of LML

- Continuous learning process
- Knowledge accumulation in KB
- Use of past knowledge to help future learning

15 Minutes Break

Outline

- A motivating example
- What is lifelong machine learning?
- Related learning tasks
- Lifelong supervised learning
- Semi-supervised never-ending learning
- Lifelong unsupervised learning
- Lifelong reinforcement learning
- Summary

LTM: Lifelong Topic Modeling

(Chen and Liu, ICML-2014)

- Topic modeling (Blei et al 2003) finds topics from a collection of documents.
  - A document is a distribution over topics
  - A topic is a distribution over terms/words, e.g.,
    - {*price*, *cost*, *cheap*, *expensive*, …}

LTM: Lifelong Topic Modeling
(Chen and Liu, ICML-2014)

- Topic modeling (Blei et al 2003) finds topics from a collection of documents.
  - A document is a distribution over topics
  - A topic is a distribution over terms/words, e.g.,
    - {*price*, *cost*, *cheap*, *expensive*, …}
- **Question**: how to find good past knowledge and use it to help new topic modeling tasks?
- **Data**: product reviews in the sentiment analysis context

Sentiment Analysis (SA) Context

- "*The size is great, but pictures are poor*."
  - Aspects (product features): size, picture
- Why lifelong learning can help SA?
  - Online reviews: Excellent data with extensive sharing of aspect/concepts across domains
    - A large volume for all kinds of products
- Why big (and diverse) data?
  - Learn a broad range of reliable knowledge. More knowledge makes future learning easier.

Key Observation in Practice

- A fair amount of aspect overlapping across reviews of different products or domains
  - Every product review domain has the aspect *price*,
  - Most electronic products share the aspect *battery*
  - Many also share the aspect of *screen*.
- This sharing of concepts / knowledge across domains is true in general, not just for SA.
  - It is rather “silly” not to exploit such sharing in learning

Problem setting

- Given a large set of document collections (big data), $D = \{D_1, D_2, \ldots, D_N\}$, learn from each D_i to produce the results S_i. Let $S = \cup_i S_i$.
  - S is called *topic base*
- Goal: Given a test/new collection D^t, learn from D^t with the help of S (and possibly D).
  - D^t in D or D^t not in D
  - The results learned this way should be better than those without the guidance of S (and D)

What knowledge?

- Should be in the same aspect/topic

  => Must-Links

  e.g., {picture, photo}

- Should not be in the same aspect/topic

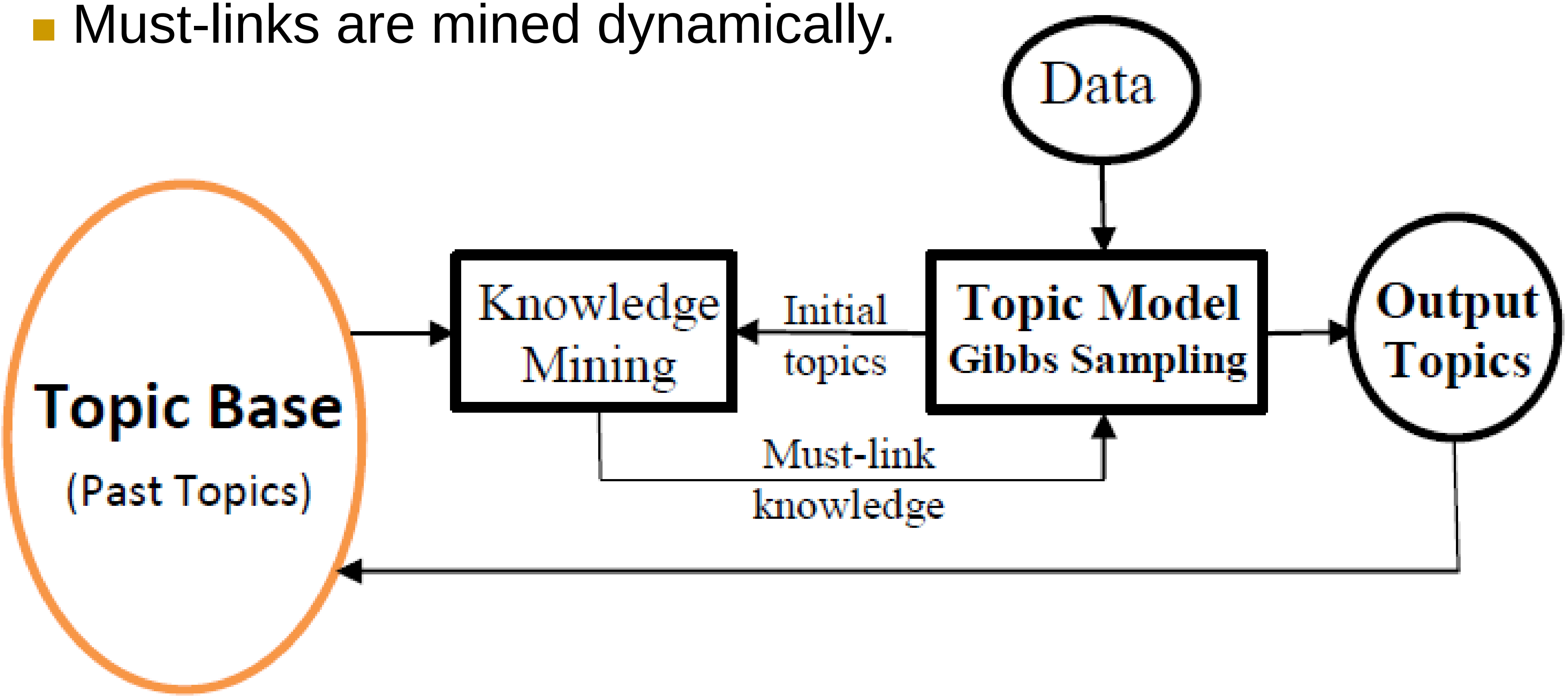
  => Cannot-Links

  e.g., {battery, picture}

Lifelong Topic Modeling (LTM)

(Chen and Liu, ICML 2014)

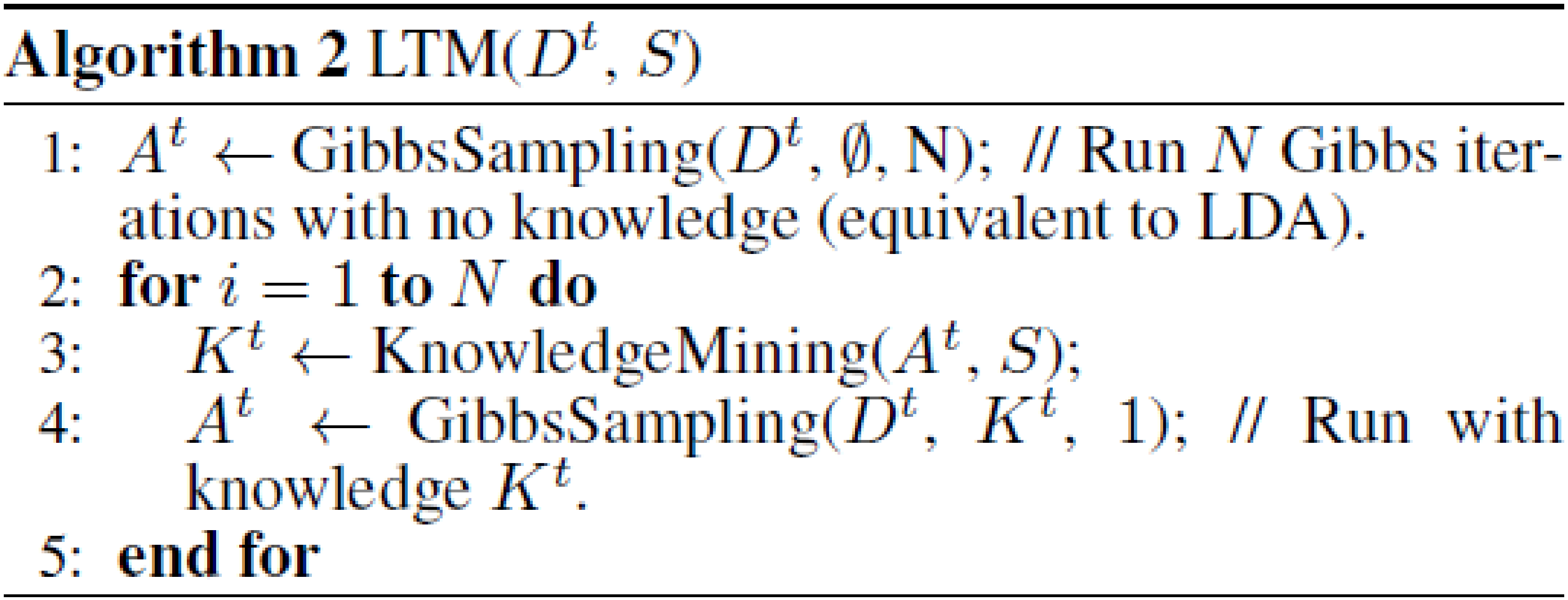
- Must-links are mined dynamically.



LTM Model

- Step 1: Run a topic model (e.g., LDA) on each domain D_i to produce a set of topics S_i called Topic Base
- Step 2: Mine prior knowledge (must-links) and use knowledge to guide modeling.

LTM Model



Algorithm 2 LTM(D^t, S)

```
$A^t \leftarrow$ GibbsSampling($D^t$, $\emptyset$, N); // Run $N$ Gibbs iterations with no knowledge (equivalent to LDA).
for $i = 1$ to $N$ do
    $K^t \leftarrow$ KnowledgeMining($A^t$, $S$);
    $A^t \leftarrow$ GibbsSampling($D^t$, $K^t$, 1); // Run with knowledge $K^t$.
end for
```

Knowledge Mining Function

- Topic matching: find similar topics from topic base for each topic in the new domain

- Pattern mining: find frequent itemsets from the matched topics

An Example

- Given a newly discovered topic:

  {*price*, *book*, *cost*, *seller, money*}

  - We find 3 matching topics from topic base *S*
    - Domain 1: {*price*, *color*, *cost*, *life, picture*}
    - Domain 2: {*cost*, *screen*, *price, expensive, voice*}
    - Domain 3: {*price*, *money*, *customer, expensive*}

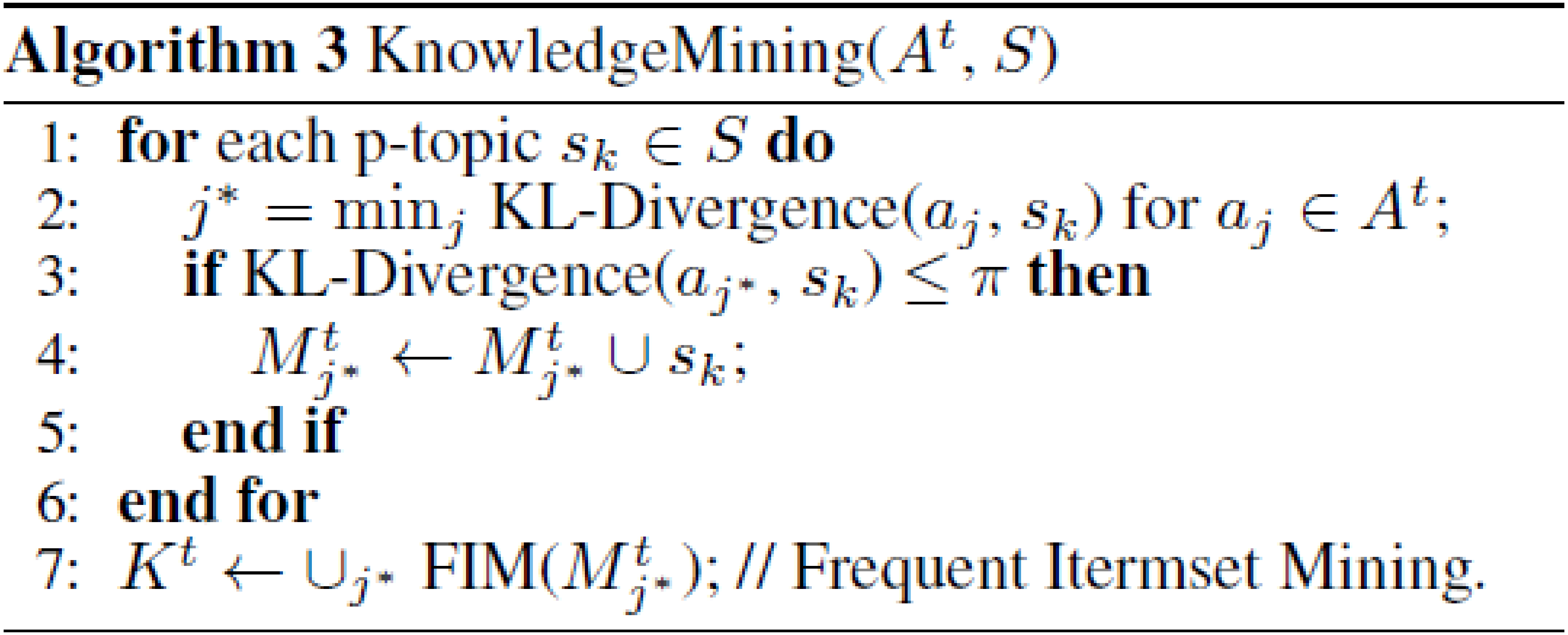
An Example

- Given a newly discovered topic:

  {*price*, *book*, *cost*, *seller, money*}

  - We find 3 matching topics from topic base *S*
    - Domain 1: {*price*, *color*, *cost*, *life, picture*}
    - Domain 2: {*cost*, *screen*, *price, expensive, voice*}
    - Domain 3: {*price*, *money*, *customer, expensive*}
- If we require words to appear in at least two domains, we get two must-links (knowledge):
  - {*price*, *cost*} and {*price*, *expensive*}.
  - Each set is likely to belong to the same aspect/topic.

Knowledge Mining Function

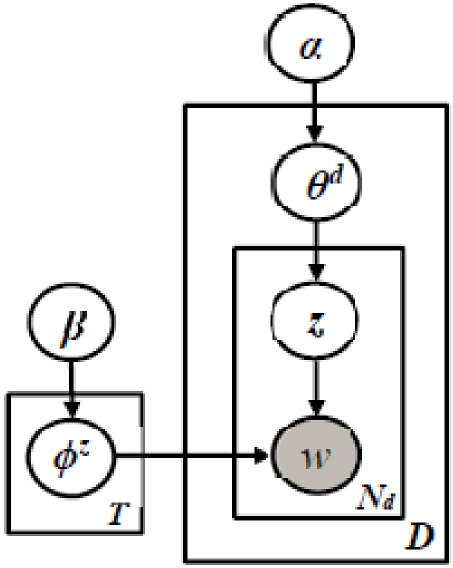


Algorithm 3 KnowledgeMining(A^t, S)

```
for each p-topic $s_k \in S$ do
    $j^* = \min_j$ KL-Divergence($a_j$, $s_k$) for $a_j \in A^t$;
    if KL-Divergence($a_{j^*}$, $s_k$) $\leq \pi$ then
        $M^t_{j^*} \leftarrow M^t_{j^*} \cup s_k$;
    end if
end for
$K^t \leftarrow \cup_{j^*}$ FIM($M^t_{j^*}$); // Frequent Itermset Mining.
```

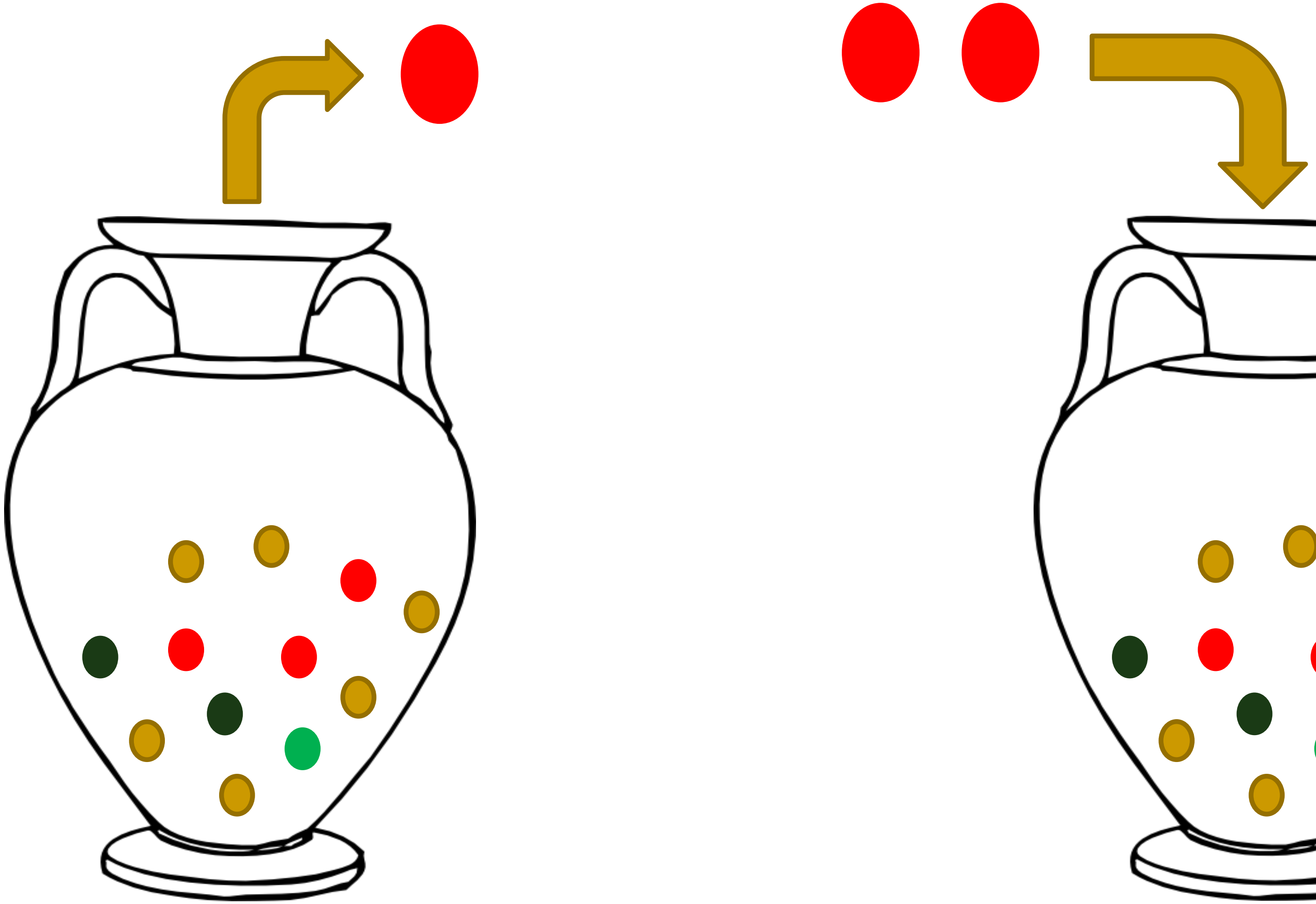
Model Inference: Gibbs Sampling

- How to use the *must-links* knowledge?
  - e.g., {*price*, *cost*} & {*price*, *expensive*}

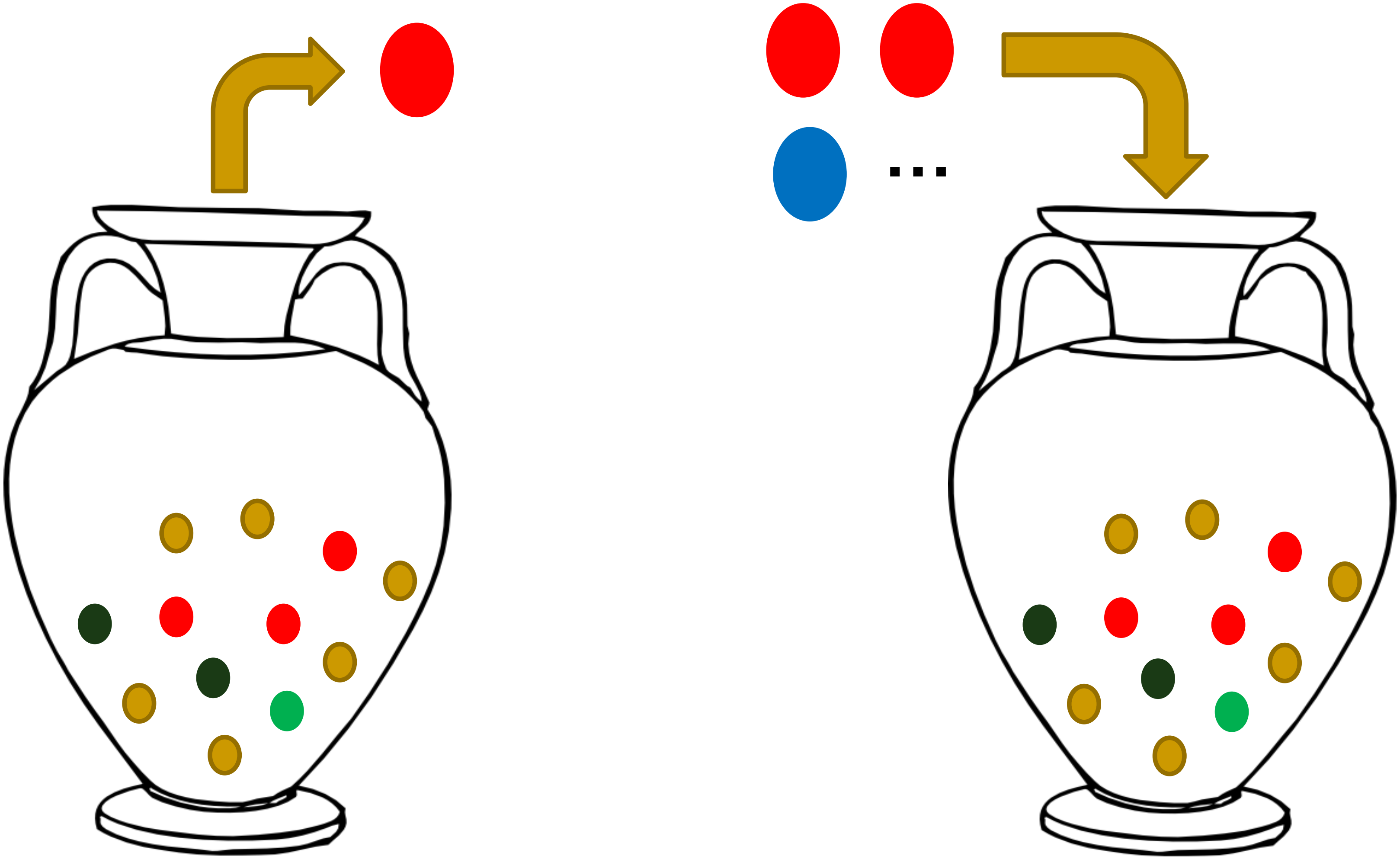


- Graphical model: same as LDA
- But the model inference is very different
  - Generalized Pólya Urn Model (GPU)
- Idea: When assigning a topic *t* to a word *w*, also assign *a fraction of t* to words in must-links sharing with *w*.

Simple Pólya Urn model (SPU)



Generalized Pólya Urn model (GPU)



Experiment Results

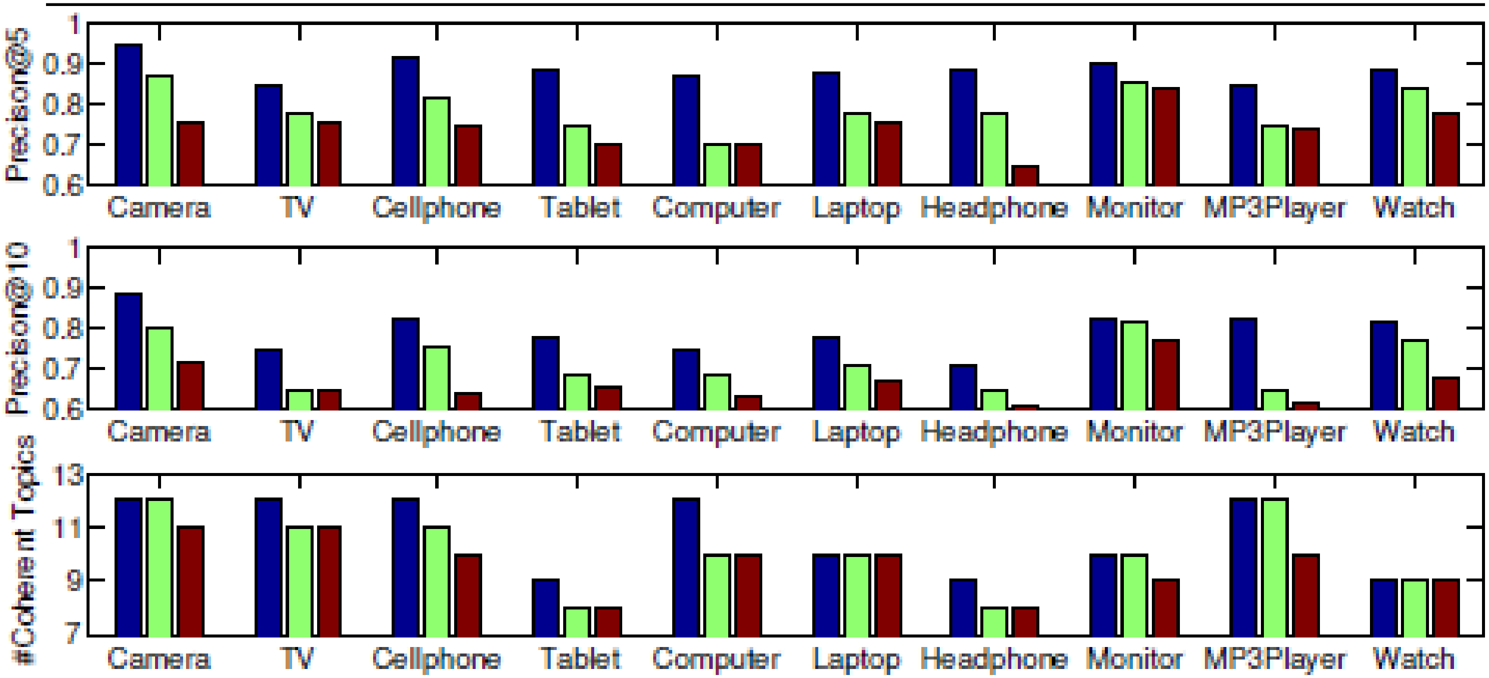


Figure 2. Top & Middle: Topical words *Precision*@5 & *Presicion*@10 of coherent topics of each model respectively; Bottom: number of coherent (#Coherent) topics discovered by each model. The bars from left to right in each group are for LTM, LDA, and DF-LDA. On average, for *Precision*@5 and

LML components of LTM

- Knowledge Base **(KB)**
  - Past information store (**PIS**): It stores topics/aspects generated in the past tasks
    - Also called topic base
  - Knowledge store (**KS**): It contains knowledge mined from PIS: Must-Links
  - Knowledge miner (**KM**): Frequent pattern mining using past topics as transactions
- Knowledge-based learner (**KBL**): LTM is based on Generalized Pólya Urn Model

AMC: Modeling with Small Datasets

(Chen and Liu, KDD-2014)

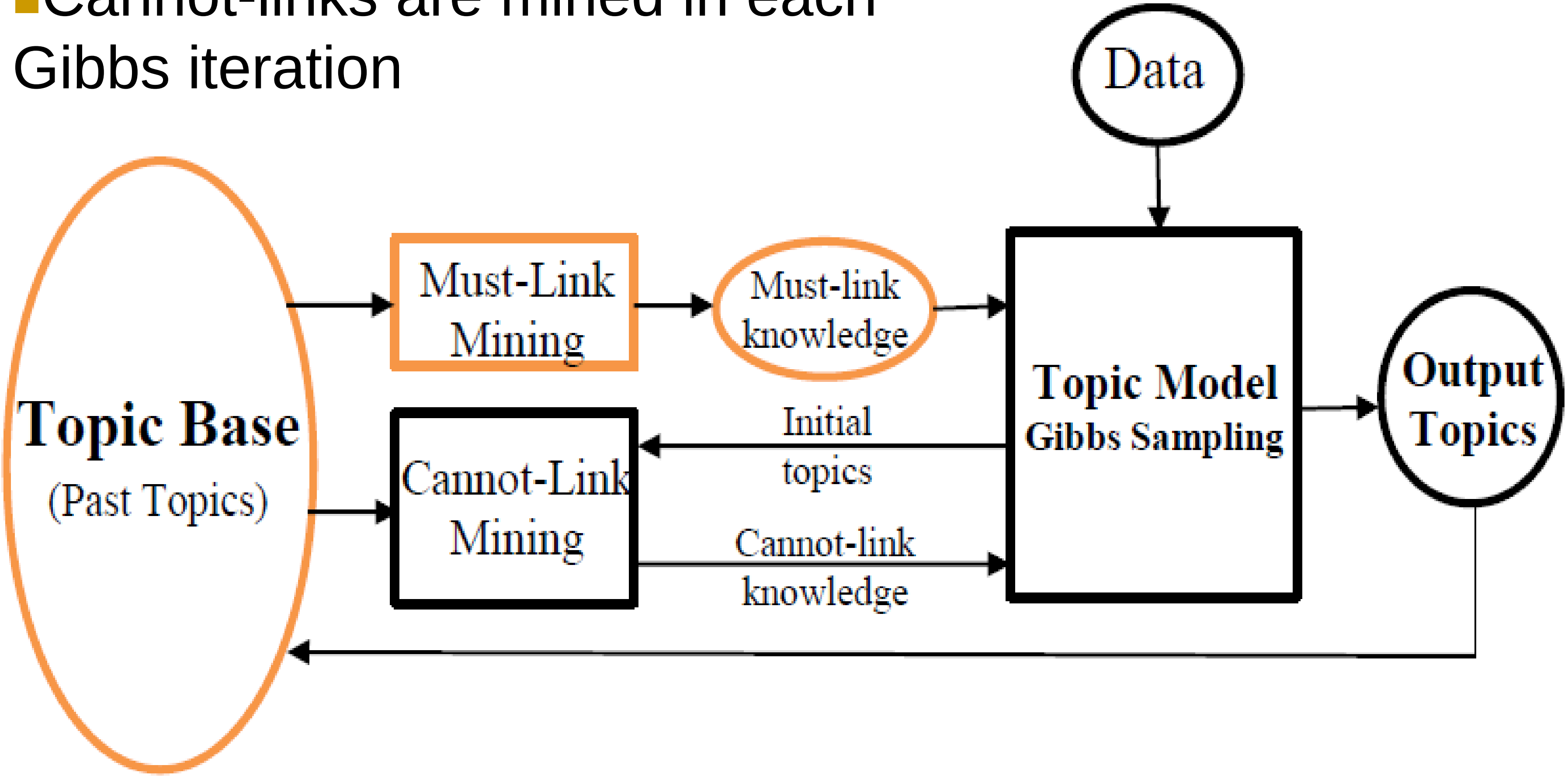
- The LTM model is not sufficient when the data is small for each task because
  - It cannot produce good initial topics for matching to identify relevant past topics.
- AMC mines must-links differently
  - Mine must-links from the PIS without considering the target task/data

Cannot-Links

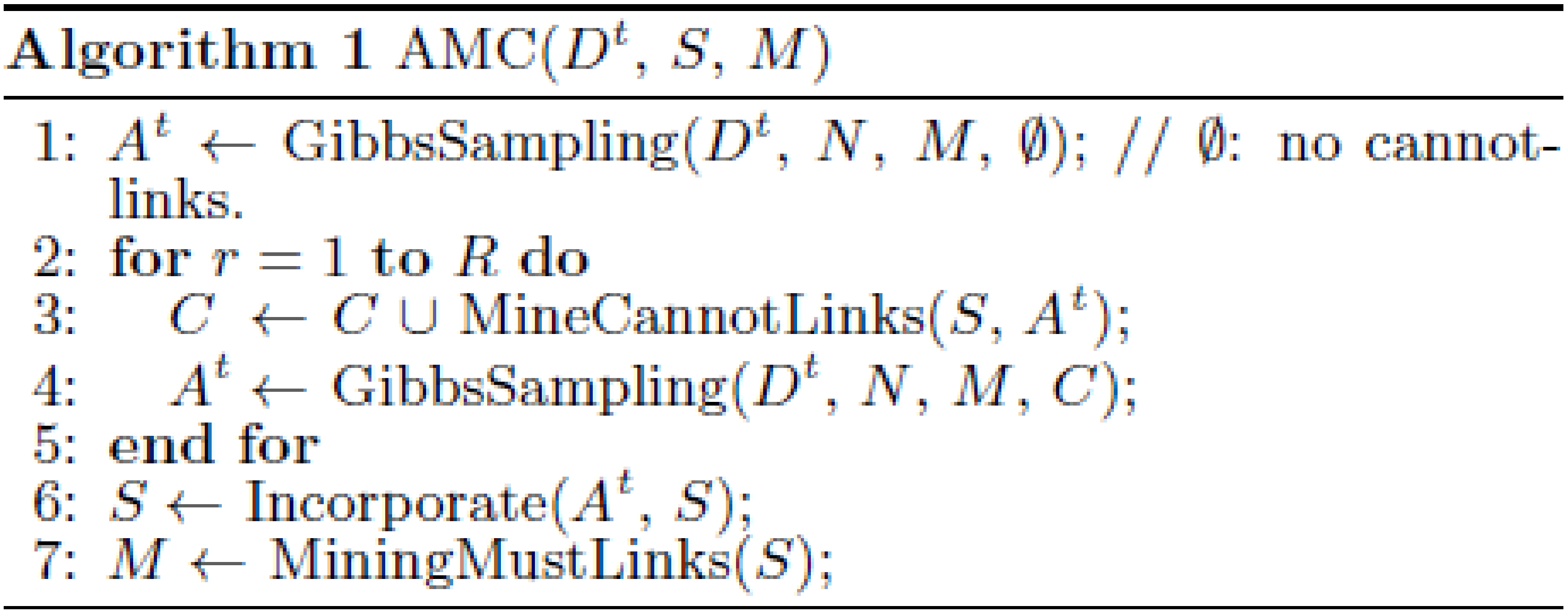
- In this case, we need to mine cannot-links, which is tricky because
  - There is a huge number of cannot-links $O(V^2)$
    - V is the vocabulary size
- We thus need to focus on only those terms that are relevant to target data D^t.
  - That is, we need to embed the process of finding cannot-links in the sampling

Lifelong Topic Modeling – AMC

- Cannot-links are mined in each Gibbs iteration



Overall Algorithm

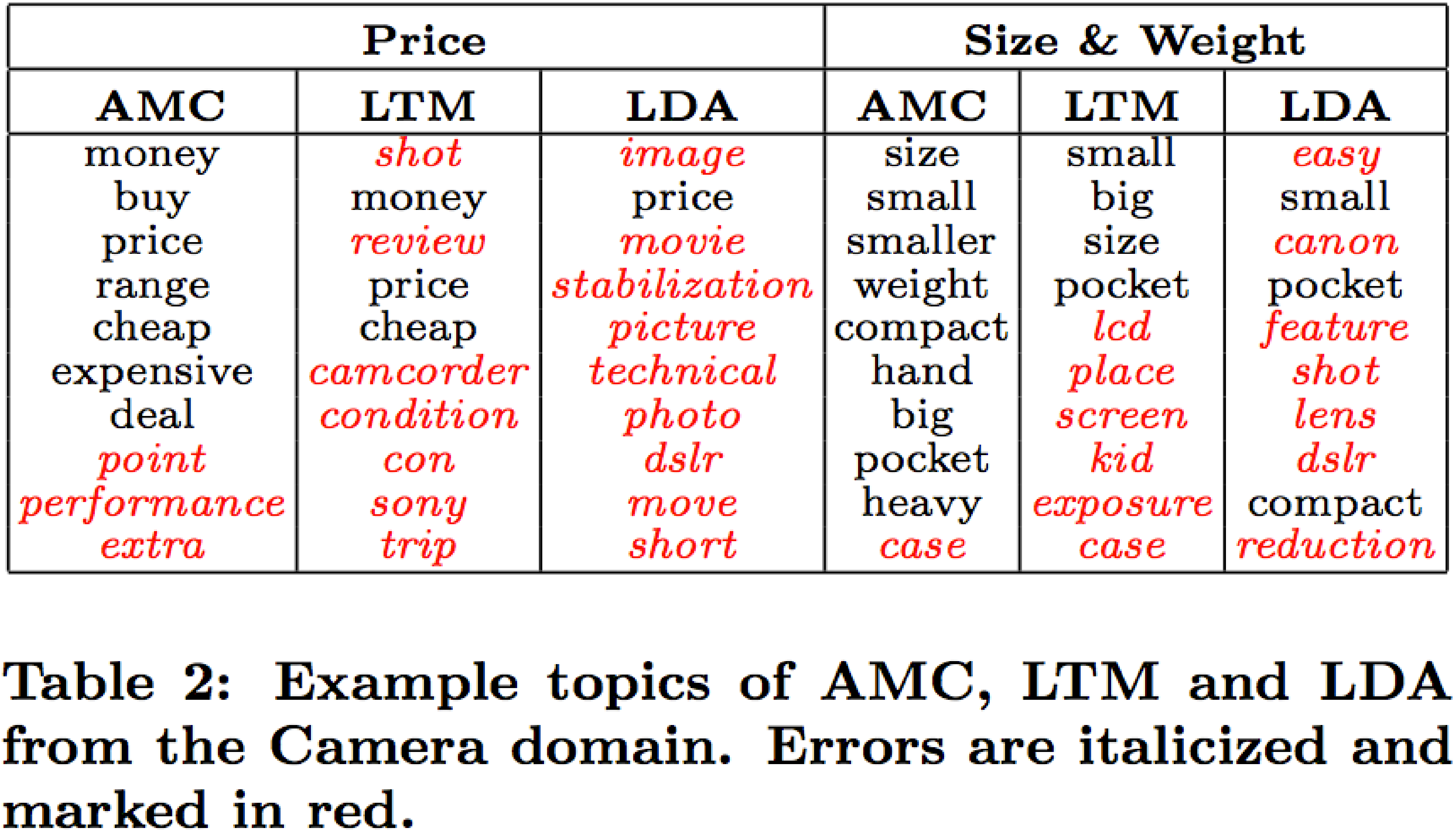


Algorithm 1 AMC(D^t, S, M)

```
$A^t \leftarrow$ GibbsSampling($D^t$, $N$, $M$, $\emptyset$); // $\emptyset$: no cannot-links.
for $r = 1$ to $R$ do
    $C \leftarrow C \cup$ MineCannotLinks($S$, $A^t$);
    $A^t \leftarrow$ GibbsSampling($D^t$, $N$, $M$, $C$);
end for
$S \leftarrow$ Incorporate($A^t$, $S$);
$M \leftarrow$ MiningMustLinks($S$);
```

- Sampling becomes much more complex
    - It proposed M-GPU model (multi-generalized Polya urn model)

AMC results



Price			Size & Weight		
AMC	**LTM**	**LDA**	**AMC**	**LTM**	**LDA**
money	*shot*	*image*	size	small	*easy*
buy	money	price	small	big	small
price	*review*	*movie*	smaller	size	*canon*
range	price	*stabilization*	weight	pocket	pocket
cheap	cheap	*picture*	compact	*lcd*	*feature*
expensive	*camcorder*	*technical*	hand	*place*	*shot*
deal	*condition*	*photo*	big	*screen*	*lens*
point	*con*	*dslr*	pocket	*kid*	*dslr*
performance	*sony*	*move*	heavy	*exposure*	compact
extra	*trip*	*short*	*case*	*case*	*reduction*

Table 2: Example topics of AMC, LTM and LDA from the Camera domain. Errors are italicized and marked in red.

Lifelong Learning components

- Knowledge Base (**KB**)
  - Past information store (**PIS**): It stores topics/aspects generated in the past tasks
  - Knowledge store (**KS**): It contains knowledge mined from PIS: must-links and cannot-links
  - Knowledge miner (**KM**): Frequent pattern mining & …
- Knowledge-based learner (**KBL**): LTM based on multi-generalized Polya urn Model

Reflection on Sentiment Applications

- Sentiment analysis (SA): two key concepts form its core
  - (1) sentiment and (2) sentiment target or aspect
- Key observation: Due to highly focused nature, SA tasks and data have a significant amount of sharing of sentiment and aspect expressions
  - Makes *lifelong learning* promising
- Data: a huge volume of reviews of all kinds

LAST Model

- Lifelong aspect-based sentiment topic model (Wang et al., 2016)

- Knowledge
  - Aspect-opinion pair, e.g., {shipping, quick}
  - Aspect-aspect pair, e.g., {shipping, delivery}
  - Opinion-opinion pair, e.g, {quick, fast}

Aspect Extraction through Lifelong Recommendation

- AER (Aspect Extraction based on Recommendations) (Liu et al., 2016)

- Based on double propagation (Qiu et al, 2011)
  - Using syntactic relations
  - Detecting new aspects using known opinion words
  - Identifying new opinion words using known aspects

Two types of Recomm. in AER

- Similarity-based recommendation
  - Word2vec
  - Trained on a large corpus of 5.8 million reviews

- Aspect associations based recommendation
  - Association rule mining
  - Example: picture, display → video, purchase

Lifelong graph labeling for SA (Shu et al., 2016)

- Problem: opinion target labeling
  - Separating entities and aspects
  - Example: “Although the engine is slightly weak, this car is great.” Entity: car; Aspect: engine
- Suitable for lifelong learning
  - Similar usage or expression across domains

Lifelong graph labeling for SA (Shu et al., 2016)

- Some words can be aspects in some domains, but entities in other domains
  - Battery is an aspect in “Camera”, “Laptop”, “Cellphone”
  - Battery is an entity in product “Battery”

LML knowledge base

- Type modifiers
  - E.g., in “this camera”, type of “camera” is entity
- Relation modifiers
  - E.g., in “the camera’s battery”, “camera” indicates an entity-aspect modifier for “battery”
- Predicted labels from past domains

Outline

- A motivating example
- What is lifelong machine learning?
- Related learning tasks
- Lifelong supervised learning
- Semi-supervised never-ending learning
- Lifelong unsupervised learning
- Lifelong reinforcement learning
- Summary

Reinforcement Learning

- An agent learns actions through trial and error interactions with a dynamic environment
- The agent gets reward/penalty after each action
- Each action changes the state of the environment
- The agent usually needs a large amount of quality experience (cost is high)

Lifelong Reinforcement Learning (LRL)

- Utilize the experience accumulated from other tasks
- Learn faster in a new task with fewer interactions
- Particularly useful in high-dimensional control problems

Example LRL Works

- Lifelong robot learning with knowledge memorization (Thrun and Mitchell 1995)
- Treating each environment as a task (Tanaka and Yamamura 1997)
- Hierarchical Bayesian approach (Wilson et al., 2007)
- Policy Gradient Efficient Lifelong Learning Algorithm (PG-ELLA) (Bou Ammar et al., 2014)

Outline

- A motivating example
- What is lifelong machine learning?
- Related learning tasks
- Lifelong supervised learning
- Semi-supervised never-ending learning
- Lifelong unsupervised learning
- Lifelong reinforcement learning
- Summary

Summary

- This tutorial gave an introduction to LML
  - By no means exhaustive
- Existing LML research is still in its infancy
  - The understanding of LML is very limited
  - Current research mainly focuses on
    - Only one type of tasks in a system
- LML needs big data – to learn a large amount of reliable knowledge of different types.
  - Little knowledge is not very useful

Summary

There are many challenges for LML, e.g.,

- It is desirable to retain as much information and knowledge as possible from the past, but
  - How to "remember" them over time effectively
  - How to represent different forms of knowledge
  - How to consolidate and meta-mine knowledge
  - How to find relevant knowledge to apply
- What is the general way of using different types of knowledge in learning?

Thank You!

Reference (1)

Ana Paula Appel and Estevam Rafael Hruschka Junior. 2011. Prophet--a link-predictor to learn new rules on nell. In Proceedings of 2011 IEEE 11th International Conference on Data Mining Workshops, pages 917–924.

G. H. Bakhir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. V. N. Vishwanathan. 2007. Predicting Structured Data. Cambridge, MA, USA: MIT Press.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993–1022.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In Proceedings of ACL, pages 440–447.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain Adaptation with Structural Correspondence Learning. In Proceedings of EMNLP, pages 120–128.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In Proceedings of COLT, pages 92–100.

Reference (2)

Haitham Bou Ammar, Rasul Tutunov, and Eric Eaton. 2015. Safe policy search for lifelong reinforcement learning with sublinear regret. In Proceedings of ICML.

Andrew Carlson, Justin Betteridge, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2009. Coupling Semi-Supervised Learning of Categories and Relations. In Proceedings of Proc. of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing.

Rich Caruana. 1997. Multitask Learning. Machine learning, 28(1), 41–75.

Zhiyuan Chen and Bing Liu. 2014. Topic Modeling using Topics from Many Domains, Lifelong Learning and Big Data. In Proceedings of ICML, pages 703–711.

Zhiyuan Chen and Bing Liu. 2014. Mining Topics in Documents : Standing on the Shoulders of Big Data. In Proceedings of KDD, pages 1116–1125.

Zhiyuan Chen, Nianzu Ma, and Bing Liu. 2015. Lifelong Learning for Sentiment Classification. In Proceedings of ACL, pages 750–756.

Reference (3)

Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect Extraction with Automated Prior Knowledge Learning. In Proceedings of ACL, pages 347–358.

Sanjoy Dasgupta, Michael L. Littman, and David McAllester. 2001. PAC generalization bounds for co-training. Advances in neural information processing systems, 1, 375–382.

Hal Daumé III. 2008. Bayesian multitask learning with latent hierarchies. In Proceedings of Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pages 135–142.

Geli Fei, Shuai Wang, and Bing Liu. 2016. Learning Cumulatively to Become More Knowledgeable. In Proceedings of KDD.

Kuzman Ganchev, João V Graça, John Blitzer, and Ben Taskar. 2008. Multi-view learning over structured and non-identical outputs. In Proceedings of In Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI.

Jayant Krishnamurthy and Tom M. Mitchell. 2011. Which Noun Phrases Denote Which Concepts. In Proceedings of Proceedings of the Forty Ninth Annual Meeting of the Association for Computational Linguistics.

Reference (4)

Abhishek Kumar, Hal Daum, and Hal Daume Iii. 2012. Learning Task Grouping and Overlap in Multi-task Learning. In Proceedings of ICML, pages 1383–1390.

Bing Liu. 2015. Sentiment Analysis Mining Opinions, Sentiments, and Emotions. Cambridge University Press.

Bing Liu. 2012. Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, 5(1), 1–167.

Qian Liu, Bing Liu, Yuanlin Zhang, Doo Soon Kim, and Zhiqiang Gao. 2016. Improving Opinion Aspect Extraction using Semantic Similarity and Aspect Associations. In Proceedings of AAAI.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In Proceedings of NAACL.

Y. Lashkari M. Metral and Pattie Maes. 1998. Collaborative interface agents. Readings in agents, 111.

Ryszard S. Michalski. 1993. Learning= inferencing+ memorizing. Foundations of Knowledge Acquisition, pages 1–41. Springer.

Reference (5)

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (AAAI'15). AAAI Press 2302-2310.

Thahir Mohamed, Estevam Hruschka Jr., and Tom Mitchell. 2011. Discovering Relations between Noun Categories. In Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1447–1455. Edinburgh, Scotland, UK.: Association for Computational Linguistics.

Guillaume Obozinski, Ben Taskar, and Michael I. Jordan. 2010. Joint Covariate Selection and Joint Subspace Selection for Multiple Classification Problems. Statistics and Computing, 20(2), 231–252.

Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. IEEE Trans. Knowl. Data Eng., 22(10), 1345–1359.

Saulo D. S. Pedro and Estevam R. Hruschka Jr. 2012. Collective intelligence as a source for machine learning self-supervision. In Proceedings of Proc. of the 4th International Workshop on Web Intelligence and Communities, pages 5:1–5:9. NY, USA: ACM

Reference (6)

.Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion Word Expansion and Target Extraction through Double Propagation. Computational Linguistics, 37(1), 9–27.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In Proceedings of Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, pages 117–120.

Paul Ruvolo and Eric Eaton. 2013. ELLA: An efficient lifelong learning algorithm. In Proceedings of ICML, pages 507–515.

Paul Ruvolo and Eric Eaton. 2013. Active Task Selection for Lifelong Machine Learning. In Proceedings of AAAI, pages 862–868.

Daniel L. Silver and Robert Mercer. 1996. The parallel transfer of task knowledge using dynamic learning rates based on a measure of relatedness. Connection Science, 8(2), 277–294.

Daniel L. Silver, Qiang Yang, and Lianghao Li. 2013. Lifelong Machine Learning Systems: Beyond Learning Algorithms. In Proceedings of AAAI Spring Symposium: Lifelong Machine Learning, pages 49–55.

Reference (7)

Lei Shu, Bing Liu, Hu Xu, and Annice Kim. 2016. Separating Entities and Aspects in Opinion Targets using Lifelong Graph Labeling. In Proceedings of EMNLP.

Ray J. Solomonoff. 1989. A system for incremental learning based on algorithmic probability. In Proceedings of Proceedings of the Sixth Israeli Conference on Artificial Intelligence, Computer Vision and Pattern Recognition, pages 515–527.

Karthik Sridharan and Sham M. Kakade. 2008. An Information Theoretic Framework for Multi-view Learning. In Proceedings of COLT, pages 403–414.

Fumihide Tanaka and Masayuki Yamamura. 1997. An approach to lifelong reinforcement learning through multiple environments. In Proceedings of 6th European Workshop on Learning Robots, pages 93–99.

S. Thrun. 1996. Explanation-Based Neural Network Learning: A Lifelong Learning Approach. Kluwer Academic Publishers.

Sebastian Thrun. 1996. Is learning the n-th thing any easier than learning the first? In Proceedings of NIPS, pages 640–646.

Reference (8)

Sebastian Thrun and Joseph O’Sullivan. 1996. Discovering Structure in Multiple Learning Tasks: The TC Algorithm. In Proceedings of ICML, pages 489–497. Morgan Kaufmann.

Shuai Wang, Zhiyuan Chen, and Bing Liu. 2016. Mining Aspect-Specific Opinion using a Holistic Lifelong Topic Model. In Proceedings of WWW.

Wei Wang and Zhi-Hua Zhou. 2010. A new analysis of co-training. In Proceedings of Proceedings of the 27th international conference on machine learning (ICML-10), pages 1135–1142.