Constraint-Driven Rank-Based Learning for Information Extraction

Sameer Singh    Limin Yao
Sebastian Riedel    Andrew McCallum

Department of Computer Science
University of Massachusetts, Amherst

Human Language Technologies: North American Chapter of the Association for Computational Linguistics (NAACL HLT)
June 2-4, 2010
Outline

1 Motivation

2 Background
   Undirected Graphical Models
   Inference and Learning

3 Semi-Supervised Rank-Based Learning
   Self-Training
   Constraints
   Self-Training and Constraints
   Model and Constraints

4 Experiments

5 Conclusions
Outline

1 Motivation

2 Background
  Undirected Graphical Models
  Inference and Learning

3 Semi-Supervised Rank-Based Learning
  Self-Training
  Constraints
  Self-Training and Constraints
  Model and Constraints

4 Experiments

5 Conclusions
Supervised Information Extraction

- Information Extraction models are becoming complex:
  - capture higher-order dependencies
  - represent tasks like coreference
  - jointly infer multiple tasks
Supervised Information Extraction

- Information Extraction models are becoming complex:
  - capture higher-order dependencies
  - represent tasks like coreference
  - jointly infer multiple tasks
- These additional edges make inference really slow

Motivation

- Training requires inference before each update:
  - over the whole dataset (gradient descent)
  - over a subset of the dataset (stochastic gradient descent)
  - over a single instance (perceptron)

SampleRank can efficiently train complex models - by incorporating updates within inference.
Supervised Information Extraction

- Information Extraction models are becoming complex:
  - capture higher-order dependencies
  - represent tasks like coreference
  - jointly infer multiple tasks
- These additional edges make inference really slow
- Training requires inference before each update:
  - over the whole dataset (gradient descent)
  - over a subset of the dataset (stochastic gradient descent)
  - over a single instance (perceptron)

SampleRank∗ can efficiently train complex models by incorporating updates within inference.

But what about Semi-Supervised Learning?

∗Khashayar et al., 2008 and Wick et al., 2009
Supervised Information Extraction

- Information Extraction models are becoming complex:
  - capture higher-order dependencies
  - represent tasks like coreference
  - jointly infer multiple tasks
- These additional edges make inference really slow
- Training requires inference before each update:
  - over the whole dataset (gradient descent)
  - over a subset of the dataset (stochastic gradient descent)
  - over a single instance (perceptron)
- SampleRank* can efficiently train complex models
  - by incorporating updates within inference

---

*Khashayar et al., 2008 and Wick et al., 2009*
Supervised Information Extraction

- Information Extraction models are becoming complex:
  - capture higher-order dependencies
  - represent tasks like coreference
  - jointly infer multiple tasks
- These additional edges make inference really slow
- Training requires inference before each update:
  - over the whole dataset \((\text{gradient descent})\)
  - over a subset of the dataset \((\text{stochastic gradient descent})\)
  - over a single instance \((\text{perceptron})\)
- SampleRank\(^*\) can efficiently train complex models
  - by incorporating updates \textit{within inference}

\textbf{But what about Semi-Supervised Learning?}

\(^*\)Khashayar et al., 2008 and Wick et al., 2009
Constraint-Based SSL

Sometimes we have prior knowledge about the tasks:

- e.g. “California” is a LOCATION
- encoded as *constraints* on features
Motivation

Constraint-Based SSL

Sometimes we have prior knowledge about the tasks:

- e.g. “California” is a LOCATION
- encoded as *constraints* on features

Use this knowledge to learn the model

- Constraint-Driven Learning (CODL): Chang *et al.*, ACL 2007
- Generalized Expectations (GE): Mann, McCallum, ACL 2008
- Alternating Projection (AP): Bellare *et al.*, UAI 2009
Constraint-Based SSL

Sometimes we have prior knowledge about the tasks:

- e.g. “California” is a LOCATION
- encoded as *constraints* on features

Use this knowledge to learn the model

- Constraint-Driven Learning (CODL): *Chang et al.*, ACL 2007
- Generalized Expectations (GE): *Mann, McCallum*, ACL 2008

*All these methods also require inference before updates*
Outline

1 Motivation

2 Background
   Undirected Graphical Models
   Inference and Learning

3 Semi-Supervised Rank-Based Learning
   Self-Training
   Constraints
   Self-Training and Constraints
   Model and Constraints

4 Experiments

5 Conclusions
Factor Graphs

- Undirected bipartite graph over variables \((x, y)\) and factors
- Each factor is associated with a scalar potential
  - dot product between parameters and features over neighbors
- Probability distribution represented by the graph:

\[
p(y|x) = \frac{1}{Z(x)} \prod_{j \in \mathcal{F}} \exp\langle \theta_j, \phi_j(x_j, y_j) \rangle
\]
MCMC Inference

- Each sample is a configuration of the variables
- Proposal function changes $y \rightarrow y^c$
- Acceptance probability depends on ratio of the model scores

$$\frac{p(y|x)}{p(y^c|x)} = \prod_{j \in \mathcal{F}} \frac{\exp \langle \theta_j, \phi_j(x_j, y_j) \rangle}{\exp \langle \theta_j, \phi_j(x_j, y^c_j) \rangle}$$
Updates parameters within MCMC-inference

Requires a truth function $F : Y \rightarrow R$
- defined as $-\mathcal{L}(y, y_L)$, where $\mathcal{L}$ is the loss, $y_L$ is labeled data
- e.g. accuracy, $F1$-score, etc.

†SampleRank: Khashayar et al., 2008 and Wick et al., 2009
Rank-Based Learning

- Updates parameters within MCMC-inference
- Requires a truth function $\mathcal{F} : \mathbf{Y} \rightarrow \mathcal{R}$
  - defined as $-\mathcal{L}(\mathbf{y}, \mathbf{y}_L)$, where $\mathcal{L}$ is the loss, $\mathbf{y}_L$ is labeled data
  - e.g. accuracy, $F1$-score, etc.
- Each pair of consecutive samples $(\mathbf{y}, \mathbf{y}^c)$ is ranked by:
  1. the model: $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{y}^c|\mathbf{x})$
  2. the truth function: $\mathcal{F}(\mathbf{y})$ and $\mathcal{F}(\mathbf{y}^c)$
- If the rankings disagree, parameters are updated
- Shown to be efficient and achieve high-accuracy

---

† Culotta et al., NAACL–HLT 2007 and Singh et al. ECML–PKDD 2009
‡ SampleRank: Khashayar et al., 2008 and Wick et al., 2009
Outline

1 Motivation

2 Background
   Undirected Graphical Models
   Inference and Learning

3 Semi-Supervised Rank-Based Learning
   Self-Training
   Constraints
   Self-Training and Constraints
   Model and Constraints

4 Experiments

5 Conclusions
Unlabeled Data

- If we can specify $\mathcal{F}$, we can perform Rank-Based Learning
- If $\mathbf{x} \in \text{labeled data}$, $\mathcal{F}(\mathbf{y}) = -\mathcal{L}(\mathbf{y}, \mathbf{y}_L)$
- For unlabeled data, we explore multiple candidates
  - based on existing semi-supervised learning techniques
(Ⅰ) Self-Training

Works as follows:

1. Train model on labeled data
2. Find the predictions on the unlabeled data
3. Add the *confident* predictions to labeled data
4. go to (1)
(Ⅰ) Self-Training

Works as follows:

1. Train model on labeled data
2. Find the predictions on the unlabeled data
3. Add the confident predictions to labeled data
4. go to (1)

Can be directly incorporated into the truth function:

\[ F_s(y) = -\mathcal{L}(y, \hat{y}_U) \]
Encoding Constraints

We may have prior knowledge about our labels
- Constraints \( \{ c_i \} \), where \( c_i(y) \) denotes whether:
  - \( y \) satisfies the constraint (+1)
  - \( y \) violates the constraint (−1)
  - constraint does not apply to \( y \) (0)
Encoding Constraints

We may have prior knowledge about our labels
- Constraints \( \{ c_i \} \), where \( c_i(y) \) denotes whether:
  - \( y \) satisfies the constraint (\(+1\))
  - \( y \) violates the constraint (\(-1\))
  - constraint does not apply to \( y \) (0)

Can be incorporated into the truth function:

\[
\mathcal{F}_c(y) = \sum_i c_i(y)
\]
By themselves, Self-Training and Constraints have major drawbacks - combine the two by including model predictions into the truth function

\[
\mathcal{F}_{sc}(y) = \mathcal{F}_s(y) + \lambda_s \mathcal{F}_c(y) = -\mathcal{L}(y, \hat{y}_U) + \lambda_s \sum_i c_i(y)
\]
(IV) Incorporating Model Scores

Previous function has two potential drawbacks:

1. Since we make updates constantly, $\hat{y}_U$ may be obsolete
2. Obtaining $\hat{y}_U$ requires full inference
(IV) Incorporating Model Scores

Previous function has two potential drawbacks:

1. Since we make updates constantly, \( \hat{y}_U \) may be obsolete
2. Obtaining \( \hat{y}_U \) requires full inference

Instead, use the current model score directly!

\[
F_{mc}(y) = \log p(y|x, \Theta) + \lambda_m F_c(y) \\
\equiv \sum_j \langle \theta_j, \phi_j(x_j, y_j) \rangle + \lambda_m \sum_i c_i(y) \quad \text{§}
\]

\[\text{§Ignore } \log Z(x) \text{ since it is independent of } y\]
1 Motivation

2 Background
   Undirected Graphical Models
   Inference and Learning

3 Semi-Supervised Rank-Based Learning
   Self-Training
   Constraints
   Self-Training and Constraints
   Model and Constraints

4 Experiments

5 Conclusions
Experiments

Setup

- Experiments on a sequential modeling task
  - Compare with existing work
  - Evaluate utility where exact inference is possible

- Cora citation dataset
  - segment into fields such as “author”, “title” and “venue”
  - 300 training, 100 test and 100 dev
  - same constraints as in (Chang et al. ACL 2007)

- The candidates are compared with CODL and Supervised

\[\text{¶ results that did not incorporate constraints during inference}\]
Results

- Labeled Data Size: 55, 60, 65, 70, 75, 80, 85, 90, 95
- Tokenwise-Accuracy
  - Supervised
  - CODL
  - Self-Training
  - Constraints
  - Self+Cons
  - Model+Cons

Figure: Constraint-Driven Rank-Based Learning
Summary

- Incorporate semi-supervision into Rank-Based Learning
  - enabling SSL over complex graphical models
- Approach is competitive on a standard dataset
  - with methods that are intractable for complicated models

Future Work:
- Apply to more complicated, loopy models
- Analysis of which candidate is the best
- Running time comparisons
- Consider more SSL algorithms (e.g. co-training, ... )
Thanks!

Sameer Singh, Limin Yao, Sebastian Riedel, Andrew McCallum

University of Massachusetts, Amherst

{sameer,lmyao,riedel,mccallum}@cs.umass.edu

factorie.googlecode.com