Motivation

- Entity and Relation Extraction

- Difficulties as a Learning Task
  - Complex learning tasks require significant annotated resources (particularly structured learning tasks).
  - Complex annotation tasks require significant funding resources.
  - Good performance requires significant feature engineering.
    - Avoiding knowledge engineering is why we used machine learning.
    - Feature engineering generally adapts poorly to new domains.
Some Questions

- What can be done with a little bit of knowledge and a little bit of annotated data?
- How can we effectively incorporate existing (semantic) resources to provide this knowledge?
- Can this process be facilitated with an interaction protocol between the learner and the expert?

Yes We Can!

Semantic Feature Abstraction

- We incrementally augment our feature space by abstracting lexical elements with their semantic class

  His father was rushed to Westlake Hospital, an arm of Resurrection Health Care, in west suburban Chicagoland.

  His father was rushed to Westlake Hospital, an \texttt{sem:subsidiary} of Resurrection Health Care, \texttt{sem:locative_preposition} \texttt{sem:compass_direction} suburban Chicagoland.

- Semantic classes are defined through existing resources. \cite{PantelLin2002, Fellbaum1998}
- Increases expressivity; results in a simpler classification task, but increases risk of adding ambiguity.
- We reduce this risk with an interactive learning protocol.
Outline

- Introduction
- Learning Protocols
- Interactive Feature Space Refinement
- Analysis and Discussion
- Experimental Results
- Future Directions and Conclusions

Learning Protocols

Data

\[ S = \{x_i\}_{i=1}^m \]
\[ x \in \mathcal{X} \]

Domain Expert

\[ \Phi^*(x) \rightarrow x, h^* \]
**Divine Intervention**

- Oracle Feature Space: *lexical features, semantic classes*
- Standard Discriminative Classifiers: *lexical features*
- This Work: augment the lexical feature space with semantic features interactively, thereby resulting in a simpler classification problem.

---

**Supervised Learning**

- Training Data: $S = \{x_i, y_i\}_{i=1}^m$  
  $x \in \mathcal{X}$  
  $y \in \mathcal{Y}$

- Domain Expert: $\Phi^*(x) \rightarrow \mathbf{x}, h^*$

- Learning Algorithm: $\mathcal{A}(\mathcal{H}, \mathcal{L}, \Phi, S) \rightarrow h$

- Induce $h \sim h^*$
Supervised Learning

- **Training Data**
  \[ S = \{ x_i, y_i \}_{i=1}^{m} \]
  \[ x \in \mathcal{X}, \quad y \in \mathcal{Y} \]

- **Domain Expert**
  \[ \Phi^*(x) \rightarrow \mathcal{X}, \quad h^* \]

- **Learning Algorithm**
  \[ \mathcal{A}(\mathcal{H}, \mathcal{L}, \Phi, S) \rightarrow h \]

- Domain Expert

  \[ h^*, \Phi^* (x) \rightarrow x, \quad h^* \]

  Induce \( h \sim h^* \)

  Learning algorithm tells us for a given \( \{ \text{hypothesis space, loss function, feature space} \} \) triple, how well \( h \) approximates \( h^* \) with given data.

  The sample complexity increases as the hypothesis space becomes more complex and the number of irrelevant features increases (among other properties).

Active Learning

- **Training Data**
  \[ S_l = \{ x_i, y_i \}_{i=1}^{l} \]
  \[ S_u = \{ x_j \}_{j=1}^{u} \]

- **Domain Expert**
  \[ \Phi^*(x) \rightarrow \mathcal{X}, \quad h^* \]

- **Learning Algorithm**
  \[ \mathcal{A}(\mathcal{H}, \mathcal{L}, \Phi, S) \rightarrow h \]

- **Algorithm Configuration**
  \[ Q : S_u \times h \rightarrow S_{select} \]

- Improve quality of inductive information

- Domain expert

  \[ h^*, \Phi^* (x) \rightarrow x, \quad h^* \]

  Induce \( h \sim h^* \)

  Algorithm Configuration

  Querying function

  Should be improving quality of deductive information
Outline

- Introduction
- Learning Protocols
- Interactive Feature Space Refinement
- Analysis and Discussion
- Experimental Results
- Future Directions and Conclusions

Interactive Feature Space Refinement

Training Data

$S_t = \{x_i, y_i\}_{i=1}^t$
$S_d = \{x_i, y_i\}_{i=t+1}^m$

Learning Algorithm

$A(\mathcal{H}, \mathcal{L}, \Phi, S) \rightarrow h$

Domain Expert

$\Phi^*(x) \rightarrow x, h^*$

Addition of semantic class features

Algorithm Configuration

$Q : S_d \times h \rightarrow S_{select}$

Querying function
An Anthropomorphized Interaction

- Learner: Excuse me, but I really thought I got the following instance correct. Where did I go wrong?

  His father was rushed to [Westlake Hospital]_{ORG}, an arm of [Resurrection Health Care]_{ORG}, in west suburban [Chicagoland]_{ORG}.

- Expert: Oh, you didn’t know that when classifying locations all context compass directions are basically equivalent. I should have told you that.

  His father was rushed to Westlake Hospital, an arm of Resurrection Health Care, in sem:compass_direction suburban Chicagoland.

- Learner: Is this a good representation of compass directions?

  north, east, south, northwest, … , neighboring, westward

- Expert: Well, I wouldn’t include neighboring or westward.

- Learner: Thanks a lot Mr. Expert…you sure are wise in the ways of classification and semantic categories.

Interactive Feature Space Refinement

1) Learning algorithm selects instances for interaction
   a) Instances where learner is confident, but mistaken
   b) Highlights features for which external resources exist and there is evidence that abstraction can help

2) Expert selects features from selected examples which it believes are relevant to classification

3) External resources are used to analyze selected features and a feature space refinement is recommended

4) The expert accepts or rejects the recommended refinement, possibly with modification
Outline

- Introduction
- Learning Protocols
- Interactive Feature Space Refinement
- Analysis and Discussion
- Experimental Results
- Future Directions and Conclusions

Formal Justification of Semantic Feature Abstraction

- The representational assumption is that when learning exclusively without semantic features, we are approximating a feature space which is a linear combination of disjunctions with a (higher dimensionality) linear function.

  \[ f_{sem:compass\_direction} = f_{north} \lor f_{east} \lor \ldots \lor f_{southwest} \]

- It is reasonably easy to show (under some assumptions) that the mistake-bound for a Perceptron algorithm scales linearly with the expected semantic class size and margin analysis may further reduce mistake bound.

  \[ M_{perceptron} \leq \left( \frac{R}{\gamma} \right)^2 \| w^* \| \]

- Lower dimensionality also implies a VC-dimension argument for PAC framework.
Incorporating Semantic Features

- If using semantic classes in place of direct lexical features is so great, why don’t we start with these features?
  - It has been done successfully [Miller, Guinness & Zamanian, 2004; Roth & Li, 2005]
  - Possibly additional expressivity, but almost certainly additional ambiguity and noise
  - Automatically generated resources emphasize coverage and therefore tend to be high recall, low precision machinery.

- An interactive procedure ameliorates these drawbacks
  - When the learner selects areas of the feature space which are inadequate, semantic information is targeted for the task.
  - Expert verification of the process ensures that unnecessary introduction of noise is avoided.

Selecting Examples for Interaction

- The learner should present selected examples in an order that maximizes the number of interactions.
Selecting Examples for Interaction

- The learner should present selected examples in an order that maximizes the number of productive interactions.
  - Random Mistakes – Querying function randomly selects among examples with mistakes and external resources.
  - Uncertainty Sampling – Querying function selects examples with mistakes, external resources, and is near the decision boundary. Commonly used for active learning.
  - Largest Gradient Sampling – Querying function selects examples with the largest mistakes and external resources.
    - This method ensures large updates and will learn the fastest [Bordes, Ertekin, Weston & Bottou, 2005].
Outline

- Introduction
- Learning Protocols
- Interactive Feature Space Refinement
- Analysis and Discussion
- Experimental Results
- Future Directions and Conclusions

Entity and Relation Extraction

- **Data** [Roth & Yih, 2004]
  - 800 training sentences
  - 349 development sentences
  - 287 testing sentences
  - Entity types include \{person, location, organization\}
  - Relation types include \{located_in, work_for, org_based_in, live_in, kill\}

- **Features**
  - Words and bigrams
  - Context
  - Surface Patterns
  - Semantic Lists Interactively

- **Three Stage Pipeline**
  - Use a regularized structured Perceptron [Collins, 2002]
  - Interaction performed at the entity stage, but features added to all stages
Interactive vs. Static Semantic Information

- Baseline System performs semantic abstraction with top-k similarity list from [Lin, 1998]. All possible abstractions are made modulo a small stoplist.
- Interactive model uses same lists as external resources.
- Numbers reported in terms of F1 measure.

<table>
<thead>
<tr>
<th>Level of Semantics</th>
<th>None</th>
<th>Top-5</th>
<th>Top-10</th>
<th>All</th>
<th>15 interactions</th>
<th>25 interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmentation</td>
<td>90.5</td>
<td>89.9</td>
<td>87.3</td>
<td>83.6</td>
<td>92.5</td>
<td>93.7</td>
</tr>
<tr>
<td>Entity Classification</td>
<td>82.6</td>
<td>83.1</td>
<td>79.8</td>
<td>71.0</td>
<td>85.4</td>
<td>86.8</td>
</tr>
<tr>
<td>Relation Classification</td>
<td>55.0</td>
<td>56.3</td>
<td>50.3</td>
<td>41.7</td>
<td>59.8</td>
<td>61.5</td>
</tr>
</tbody>
</table>

Comparing Querying Functions

- Performance Improvement per Interaction
  - Shows that largest gradient selection leads to fruitful feature space augmentation

- Number of Examples Presented per Interaction
  - Since expert can ignore examples, we want all good interactions first
Future Work

- Perform interactions at the relation classification level
  - The entity classification level was chosen somewhat arbitrarily
  - The relation classifications level has more mistakes and more complex features; possibly more productive interactions
- Consider the domain adaptation scenario
  - Interactive semantic abstraction may be a better use of the limited resources in a new domain (that just training data)
- Consider feature space augmentations other than through semantic list abstraction (prefixes, suffixes, etc.)

Related Work

- Active Learning (many works…and growing)
- Learning with Additional Annotation
  - Generalized Expectation Criteria [Druck, Mann, & McCallum, 2008]
  - Annotator Rationales [Zaidan, Eisner & Piatko, 2007]
  - Interactive Label Feedback [Raghavan and Allan, 2007]
- Semantic Resources
  - Distributional/Parsing Similarity [Pantel & Lin, 2002]
  - WordNet [Fellbaum, 1998]
- Incorporating Semantic Resources
  - [Li & Roth, 2005; Miller, Guinness & Zamanian, 2004; Boggess, Agarwal & Davis, 1991]
- Would be happy to hear about others…
Conclusions

- We describe a method to interactively incorporate semantic list information into a discriminative classifier.
- Through an interactive protocol with the learning algorithm, we target the information to the specified task and avoid introducing unnecessary ambiguity.
- We show that our method works for a named-entity and relation extraction system.

Thanks