Multi-Component Word Sense Disambiguation

Massimiliano Ciaramita and Mark Johnson

Brown University

BLLIP: http://www.cog.brown.edu/Research/nlp

Outline

- Pattern classification for WSD
  - Features
  - Flat multiclass averaged perceptron
- Multi-component WSD
  - Generating external training data
  - Multi-component perceptron
- Experiments and results

Pattern classification for WSD

English lexical sample: 57 test words: 32 verbs, 20 nouns, 5 adjectives. For each word $\mathtt{w}$:

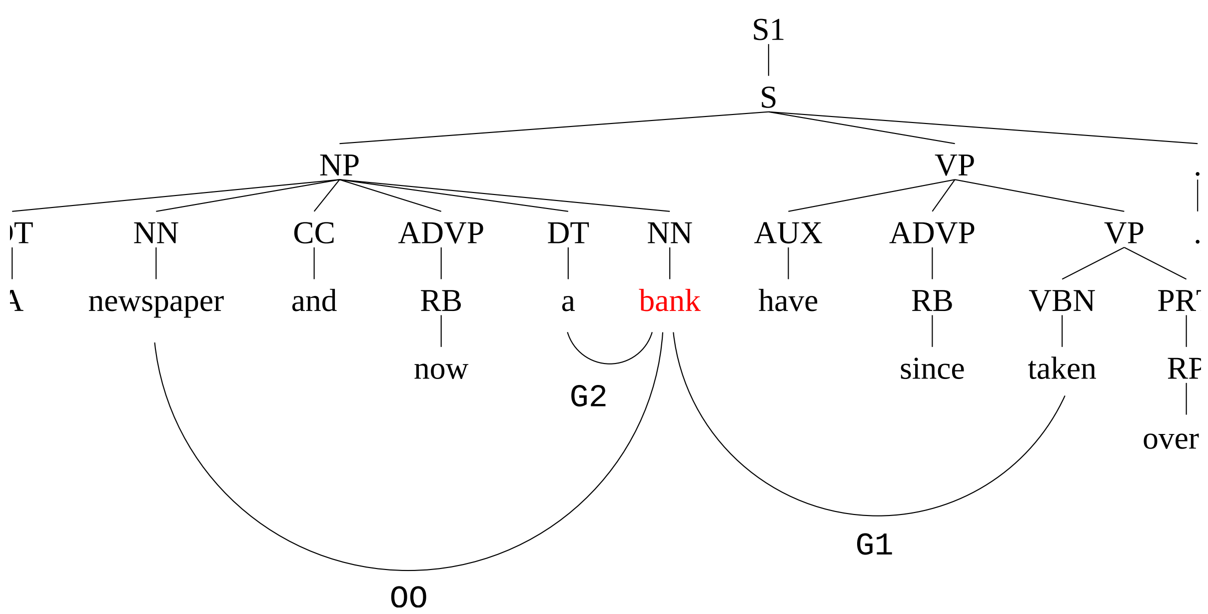
1. **compile a training set:** $S(w) = (x_i, y_i)^n$
   - $x_i \in \mathbb{R}^d$ **a vector of features**
   - $y_i \in Y(w)$, **one of the possible senses of** w
2. **learn a classifier on** $S(w)$**:** $H : \mathbb{R}^d \rightarrow Y(w)$
3. **use the classifier to disambiguate the unseen test data**

Features

- **Standard feature set for wsd (derived from (Yoong and Hwee, 2002))**
  - **"A-DT newspaper-NN and-CC now-RB a-DT *bank-NN* have-AUX since-RB taken-VBN over-RB"**
- **POS of neighboring words** - $\mathtt{P}_{x,x\in\{-3,-2,-1,0,+1,+2,+3\}}$; **e.g.,** $\mathtt{P}_{-1} = \mathtt{DT}$, $\mathtt{P}_0 = \mathtt{NN}$, $\mathtt{P}_{+1} = \mathtt{AUX}$, ...
- **Surrounding words** - $\mathtt{WS}$; **e.g.,** $\mathtt{WS} = \mathtt{take}_v$, $\mathtt{WS} = \mathtt{over}_r$, $\mathtt{WS} = \mathtt{newspaper}_n$
- **N-grams:**
  - $\mathtt{NG}_{x,x\in\{-2,-1,+1,+2\}}$; **e.g.,** $\mathtt{NG}_{-2} = \mathtt{now}$, $\mathtt{NG}_{+1} = \mathtt{have}$, $\mathtt{NG}_{+2} = \mathtt{take}$
  - $\mathtt{NG}_{x,y:(x,y)\in\{(-2,-1),(-1,+1),(+1,+2)\}}$; **e.g.,** $\mathtt{NG}_{-2,-1} = \mathtt{now\_a}$, $\mathtt{NG}_{+1,+2} = \mathtt{have\_since}$

Syntactic features (Charniak,2000)

- **Governing elements under a phrase** - $\mathtt{G}_1$; **e.g.,** $\mathtt{G}_1 = \mathtt{take\_S}$
- **Governed elements under a phrase** - $\mathtt{G}_2$; **e.g.,** $\mathtt{G}_2 = \mathtt{a\_NP}$, $\mathtt{G}_2 = \mathtt{now\_NP}$
- **Coordinates** - OO; **e.g.,** $\mathtt{OO} = \mathtt{newspaper}$



Multiclass Perceptron (Crammer and Singer, 2003)

- **Discriminant function:** $\mathtt{H}(\mathtt{x}; \mathbf{V}) = \arg\max_{\mathtt{r}=1}^{\mathtt{k}} \langle \mathtt{v}_\mathtt{r}, \mathtt{x} \rangle$
- **Input:** $\mathbf{V} \in \mathbb{R}^{|\mathtt{Y(w)}| \times \mathtt{d}}$, $\mathtt{d} \approx 200,000$, **initialized as** $\mathbf{V} = 0$
- **Repeat** $\mathtt{T}$ **times - passes over training data or** *epochs*

$\mathtt{Multiclass\_Perceptron}((\mathtt{x}, \mathtt{y})^\mathtt{n}, \mathbf{V})$

for $\mathtt{i} = 1$ **to** $\mathtt{i} = \mathtt{n}$
do $\mathtt{E} = \{\mathtt{r} : \langle \mathtt{v}_\mathtt{r}, \mathtt{x}_\mathtt{i} \rangle > \langle \mathtt{v}_\mathtt{y}, \mathtt{x}_\mathtt{i} \rangle\}$
  if $|\mathtt{E}| > 0$
    then 1. $\tau_\mathtt{r} = 1$ for $\mathtt{r} = \mathtt{y}$
      2. $\tau_\mathtt{r} = 0$ for $\mathtt{r} \notin \mathtt{E} \cup \{\mathtt{y}\}$
      3. $\tau_\mathtt{r} = -\frac{1}{|\mathtt{E}|}$ for $\mathtt{r} \in \mathtt{E}$
      for $\mathtt{r} = 1$ **to** $\mathtt{r} = \mathtt{k}$
      do $\mathtt{v}_\mathtt{r} \leftarrow \mathtt{v}_\mathtt{r} + \tau_\mathtt{r} \mathtt{x}_\mathtt{i}$;

Averaged perceptron classifier

- **Perceptron's output:** $\mathbf{V}^{(\mathtt{0})}, \ldots, \mathbf{V}^{(\mathtt{n})}$
- $\mathbf{V}^{(\mathtt{i})}$ **is the weight matrix after the first** $\mathtt{i}$ **training items**
- **Final model:** $\mathbf{V} = \mathbf{V}^{(\mathtt{n})}$
- **Averaged perceptron: (Collins, 2002)**
  - **final model:** $\mathbf{V} = \frac{1}{\mathtt{n}} \sum_{\mathtt{i}=1}^{\mathtt{n}} \mathbf{V}^{(\mathtt{i})}$
  - **reduces the effect of over-training**

Outline

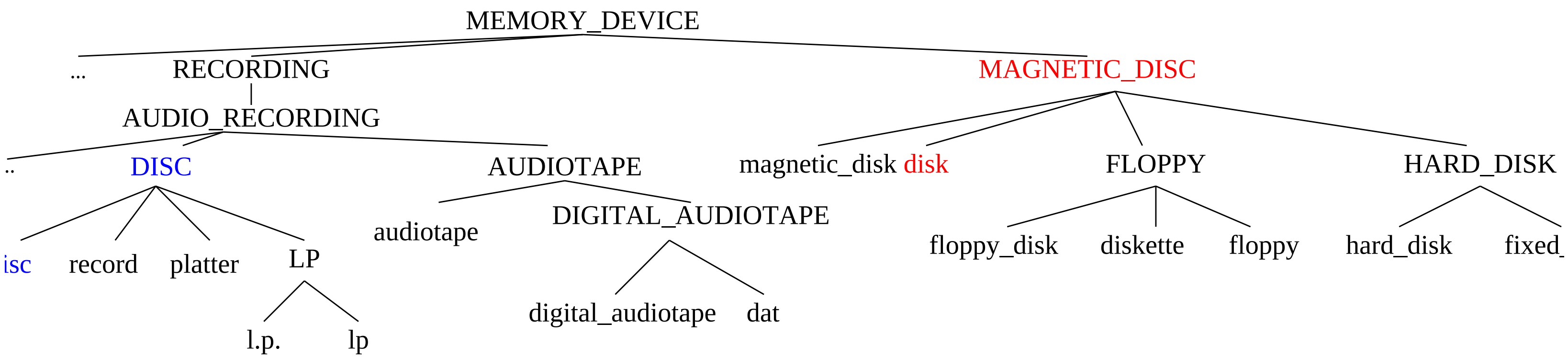
- Pattern classification for WSD
  - Features
  - Flat multiclass perceptron
- Multi-component WSD
  - Generating external training data
  - Multi-component perceptron
- Experiments and results

Sparse data problem in WSD

- **Thousands of word senses** - 120,000 in Wordnet 2.0
- **Very specific classes** - 50% of noun synsets contain one noun
- **Problem:** training instances often too few for fine-grained semantic distinctions
- **Solution:**
  1. use the hierarchy of Wordnet to find similar word senses and generate external training data for these senses
  2. integrate task-specific and external data with perceptron
- **Intuition** - to classify an instance of the noun disk additional knowledge about concepts such as other "audio" or "computer memory" devices could be helpful

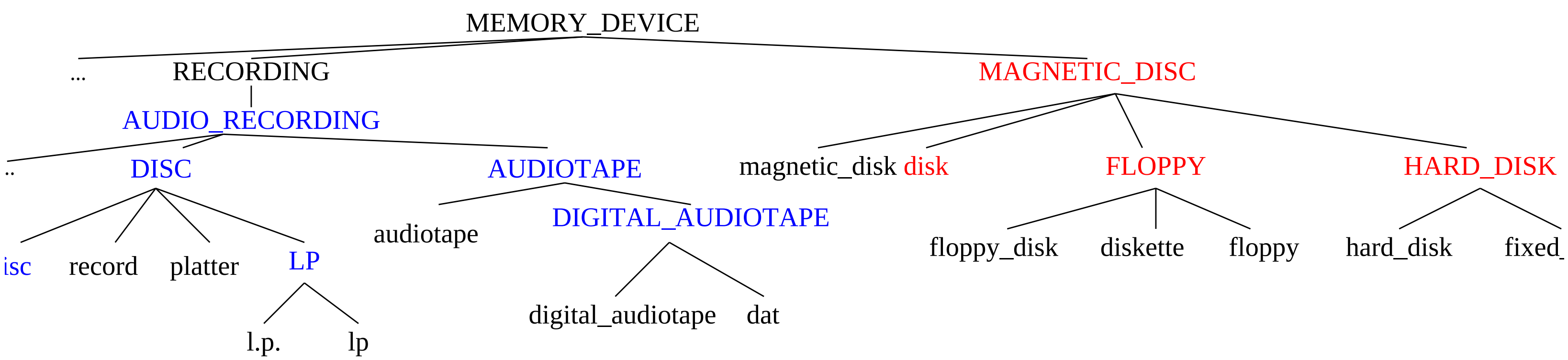
Finding neighbor senses

- disc_1 = memory device for information storing
- disc_2 = phonograph record



Finding neighbor senses

- neighbors(disc$_1$) = floppy disk, hard disk, ...
- neighbors(disc$_2$) = audio recording, lp, soundtrack, audiotape, talking book, digital audio tape, ...



External training data

- **Find neighbors**: for each sense y of a noun or verb in the task a set $\hat{y}$ of $k = 100$ neighbor senses is generated from the Wordnet hierarchy
- **Generate new instances**: for each synset in $\hat{y}$ a training instance $(x_i, \hat{y}_i)$ is compiled from the corresponding Wordnet glosses (definitions/example sentences) using the same set of features
- **Result**: for each noun/verb
  1. task-specific training data $(x_i, y_i)^n$
  2. external training data $(x_i, \hat{y}_i)^m$

Multi-component perceptron

- Simplification of hierarchical perceptron (Ciaramita et al., 2003)
- A weight matrix V is trained on the task-specific data
- A weight matrix M is trained on the external data
- Discriminant function:

$$\mathtt{H}(\mathtt{x}; \mathbf{V}, \mathbf{M}) = \arg\max_{\mathtt{y} \in \mathtt{Y}(\mathtt{w})} \lambda_{\mathtt{y}} \langle \mathtt{v}_{\mathtt{y}}, \mathtt{x} \rangle + \lambda_{\hat{\mathtt{y}}} \langle \mathtt{m}_{\hat{\mathtt{y}}}, \mathtt{x} \rangle$$

  - $\lambda_{\mathtt{y}}$ is an adjustable parameter that weights each component's contribution: $\lambda_{\hat{\mathtt{y}}} = 1 - \lambda_{\mathtt{y}}$

Multi-Component Perceptron

- **The algorithm learns $\mathbf{V}$ and $\mathbf{M}$ independently**

$\texttt{Multi-Component\_Perceptron}((x_i, y_i)^n, (x_i, \hat{y}_i)^m, \mathbf{V}, \mathbf{M})$

$\mathbf{V} \leftarrow 0$

$\mathbf{M} \leftarrow 0$

for $t = 1$ **to** $i = T$

do $\texttt{Multiclass\_Perceptron}((x_i, y_i)^n, \mathbf{V})$

$\quad\texttt{Multiclass\_Perceptron}((x_i, y_i)^n, \mathbf{M})$

$\quad\texttt{Multiclass\_Perceptron}((x_i, y_i)^m, \mathbf{M})$

Outline

- Pattern classification for WSD
  - Features
  - Flat multiclass averaged perceptron
- Multi-component WSD
  - Generating external training data
  - Multi-component perceptron
- Experiments and results

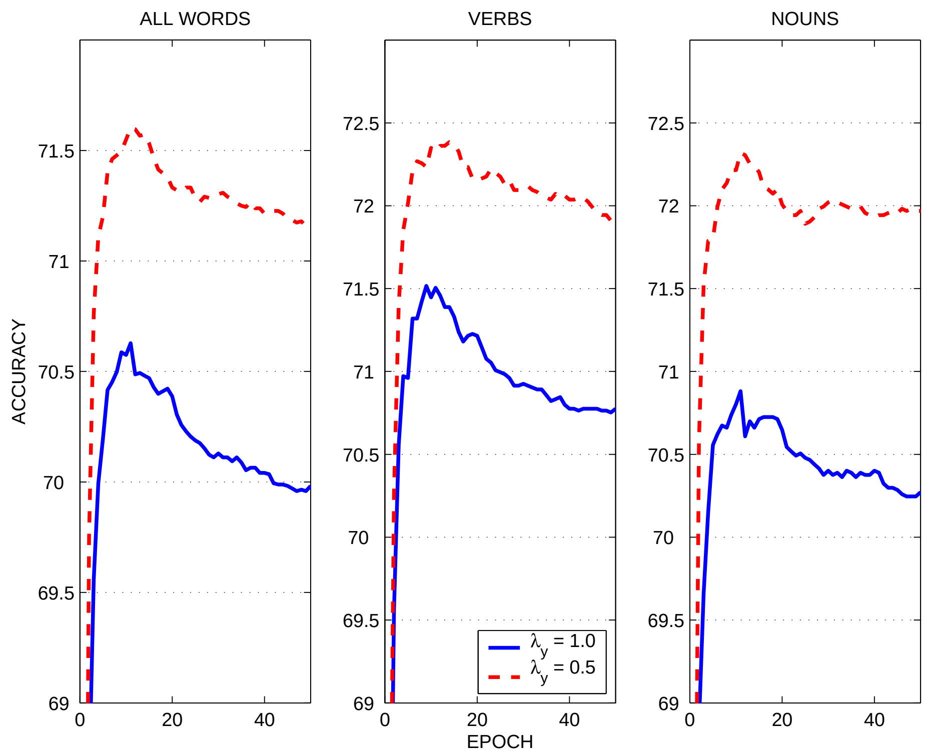
Experiments and results

- One classifier trained for each test word
- Adjectives: standard perceptron, only set $\mathtt{T}$
- Verbs/nouns: multicomponent perceptron, set $\mathtt{T}$ and λ_{y}
- Cross-validation experiments on the training data for each test word:
  1. choose the value for λ_{y}; $\lambda_{\mathrm{y}} = 1$ use only the "flat" perceptron, or $\lambda_{\mathrm{y}} = 0.5$ use both component equally weighted
  2. choose the number of iterations $\mathtt{T}$
- Average $\mathtt{T}$ value = 13.9
- For 37 out of 52 nouns/verbs $\lambda_{\mathrm{y}} = 0.5$; the two-component model is more accurate than the flat perceptron

English Lexical Sample Results

Measure	Precision	Recall	Attempted %
Fine all POS	71.1	71.1	100
Coarse all POS	78.1	78.1	100
Fine verbs	72.5	72.5	100
Coarse verbs	80.0	80.0	100
Fine nouns	71.3	71.3	100
Coarse nouns	77.4	77.4	100
Fine adjectives	49.7	49.7	100
Coarse adjectives	63.5	63.5	100

Flat vs. Multi-component: cross validation on train



Conclusion

- **Advantages of the multi-component perceptron trained on neighbors' data**
  - **Neighbors: one "supersense" for each sense, same amount of additional data per sense**
  - **Simpler model: smaller variance more homogeneous external data**
  - **Efficiency: fast and efficient training**
  - **Architecture: simple, easy to add any number of (weighted) "components"**