



---

# Outline

- Background
  - Previous Work
  - Feature Combination Experiments
  - Viterbi Realignment Experiments
  - Conclusions and Future Work
-

---

# Background

- Conditional Random Fields (CRFs)
    - Discriminative probabilistic model
    - Used successfully in various domains such as part of speech tagging and named entity recognition
    - Directly defines a posterior probability of a sequence  $Y$  given an input sequence  $X$ 
      - e.g.  $P(Y|X)$
-

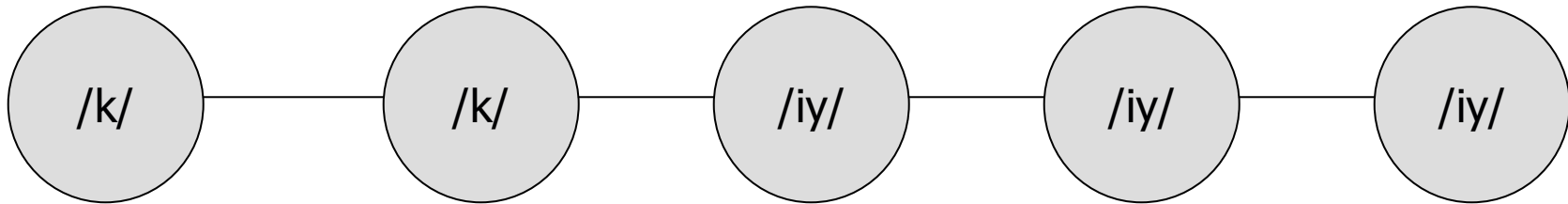
---

# Background

- CRFs for ASR
    - Phone Classification (Gunawardana et al., 2005)
      - Uses sufficient statistics to define feature functions
    - Different approach than NLP tasks using CRFs
      - Define binary feature functions to characterize observations
    - Our approach follows the latter method
      - Use neural networks to provide “soft binary” feature functions (e.g. posterior phone outputs)
-

---

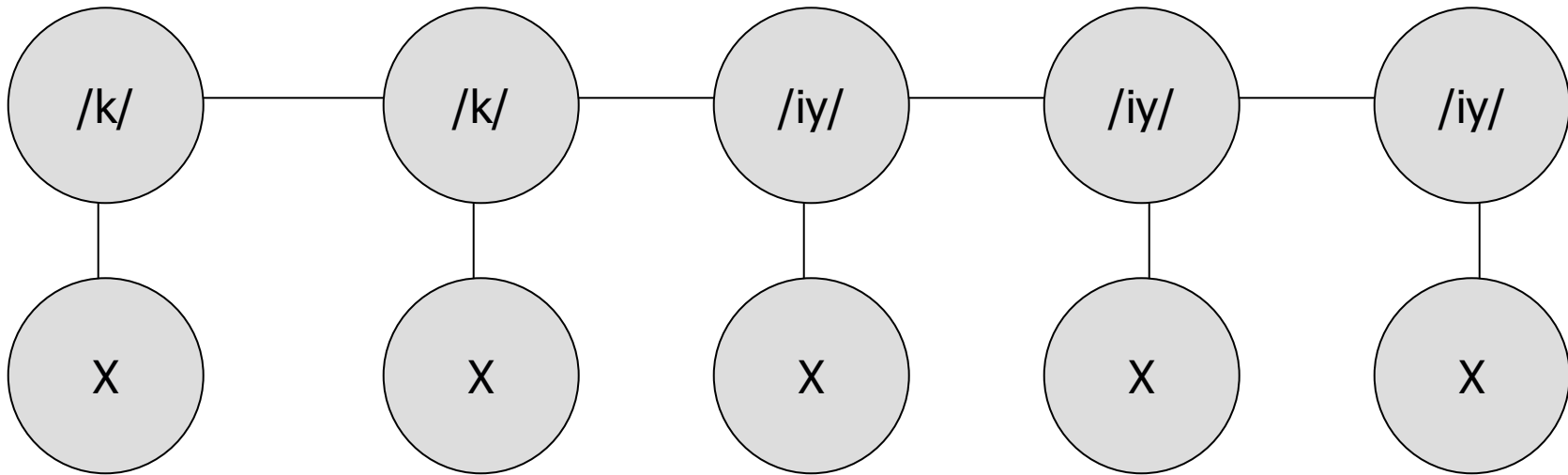
# Conditional Random Fields



- Based on the framework of Markov Random Fields

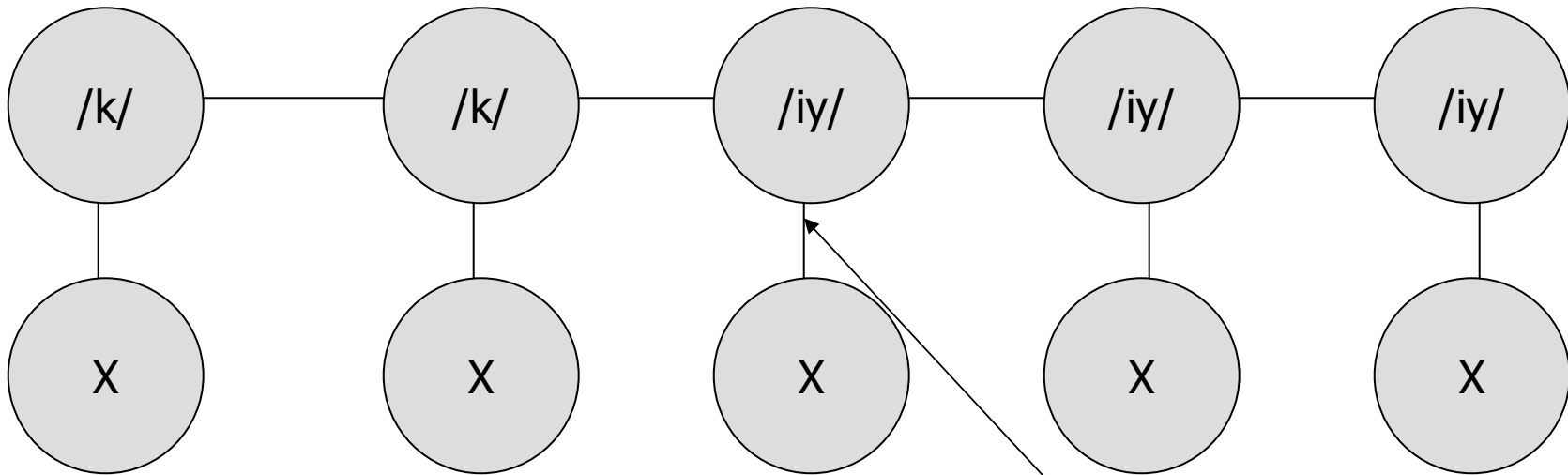
---

# Conditional Random Fields



- Based on the framework of Markov Random Fields
    - A CRF iff the graph of the label sequence is an MRF when conditioned on a set of input observations (Lafferty et al., 2001)
-

# Conditional Random Fields

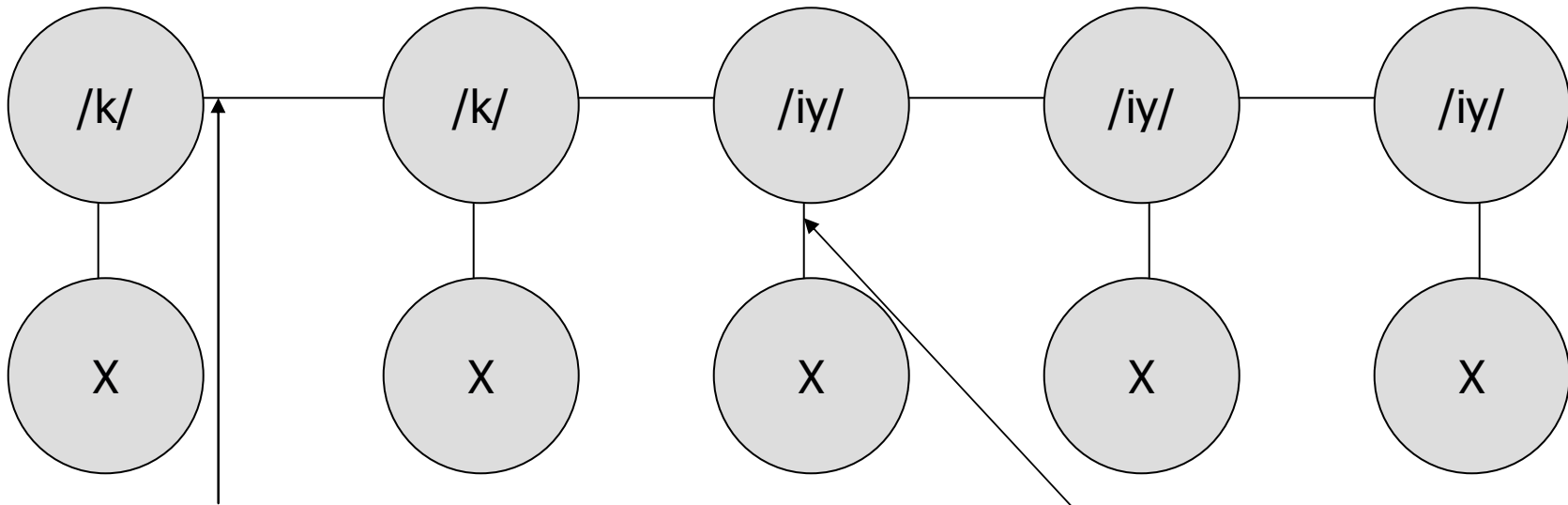


- Based on the framework of Markov Random Fields

- A CRF iff the graph of the  $\theta_i$  when conditioned on the  $\theta_j$

State functions help determine the identity of the state

# Conditional Random Fields



- **Basic Field**

Transition functions add associations between transitions from one label to another

- **A**

when conditioned on the i

## Markov Random

te functions help determine the identity of the state

---

# Conditional Random Fields

$$P(Y | X) = \frac{\exp \sum_k \left( \sum_i \lambda_i s_i(x, y_k) + \sum_j \mu_j t_j(x, y_k, y_{k-1}) \right)}{Z(x)}$$

- CRF defined by a weighted sum of state and transition functions
    - Both types of functions can be defined to incorporate observed inputs
    - Weights are trained by maximizing the likelihood function
-

---

## Previous Work

- Implemented CRF models on data from phonetic attribute detectors
    - Performed phone recognition
    - Compared results to Tandem/HMM system on same data
  - Experimental Data
    - TIMIT corpus of read speech
-

---

# Attribute Selection

- Attribute Detectors
    - ICSI QuickNet Neural Networks
  - Two different types of attributes
    - Phonological feature detectors
      - Place, Manner, Voicing, Vowel Height, Backness, etc.
      - N-ary features in eight different classes
    - Phone detectors
      - Neural networks output based on the phone labels
    - Trained using PLP 12+deltas
-

---

# Experimental Setup

- CRF code
    - Built on the Java CRF toolkit from Sourceforge
    - <http://crf.sourceforge.net>
    - Performs maximum log-likelihood training
    - Uses Limited Memory BGFS algorithm to perform minimization of the log-likelihood gradient
-

---

# Experimental Setup

$$s_{/t/,f}(y, x) = \begin{cases} NN_f(x), & \text{if } y = /t/ \\ 0, & \text{otherwise} \end{cases}$$

- *Feature functions* built using the neural net output
    - Each attribute/label combination gives one feature function
-

---

# Experimental Setup

- Baseline system for comparison
    - Tandem/HMM baseline (Hermansky et al., 2000)
    - Use outputs from neural networks as inputs to gaussian-based HMM system
    - Built using HTK HMM toolkit
  - Linear inputs
    - Better performance for Tandem with linear outputs from neural network
    - Decorrelated using a Karhunen-Loeve (KL) transform
-

## Initial Results (Morris & Fosler-Lussier, 06)

<b>Model</b>	<b>Params</b>	<b>Phone Accuracy</b>
Tandem [1] (phones)	20,000+	60.82%
Tandem [3] (phones) 4mix	420,000+	68.07%
<b>CRF [1] (phones)</b>	<b>5280</b>	<b>67.32%</b>
Tandem [1] (feas)	14,000+	61.85%
Tandem [3] (feas) 4mix	360,000+	68.30%
<b>CRF [1] (feas)</b>	<b>4464</b>	<b>65.45%</b>
Tandem [1] (phones/feas)	34,000+	61.72%
Tandem [3] (phones/feas)	774,000+	68.46%
<b>CRF (phones/feas)</b>	<b>7392</b>	<b>68.43%</b>

---

# Feature Combinations

- CRF model supposedly robust to highly correlated features
    - Makes no assumptions about feature independence
  - Tested this claim with combinations of correlated features
    - Phone class outputs + Phono. Feature outputs
    - Posterior outputs + transformed linear outputs
  - Also tested whether linear, decorrelated outputs improve CRF performance
-

---

## Feature Combinations - Results

<b>Model</b>	<b>Phone Accuracy</b>
CRF (phone posteriors)	67.32%
CRF (phone linear KL)	66.80%
<b>CRF (phone post+linear KL)</b>	<b>68.13%</b>
CRF (phono. feature post.)	65.45%
CRF (phono. feature linear KL)	66.37%
<b>CRF (phono. feature post+linear KL)</b>	<b>67.36%</b>

---

---

# Viterbi Realignment

- Hypothesis: Poor CRF results could be due to using only pre-defined boundaries
    - HMM allows boundaries to shift during training
    - Basic CRF training process does not
  - Modify training to allow for better boundaries
    - Train CRF with fixed boundaries
    - Force align training labels using CRF
    - Adapt CRF weights using new boundaries
-

## Viterbi Realignment - Results

<b>Model</b>	<b>Accuracy</b>
CRF (phone posteriors)	67.32%
<b>CRF (phone posteriors – realigned)</b>	<b>69.92%</b>
Tandem[3] 4mix (phones)	68.07%
Tandem[3] 16mix (phones)	69.34%
CRF (phono. fea. linear KL)	66.37%
<b>CRF (phono. fea. lin-KL – realigned)</b>	<b>68.99%</b>
Tandem[3] 4mix (phono fea.)	68.30%
Tandem[3] 16mix (phono fea.)	69.13%
CRF (phones+feas)	68.43%
<b>CRF (phones+feas – realigned)</b>	<b>70.63%</b>
Tandem[3] 16mix (phones+feas)	69.40%

---

# Conclusions

- Using correlated features in the CRF model did not degrade performance
    - Extra features improved performance for the CRF model across the board
  - Viterbi realignment training significantly improved CRF results
    - Improvement did not occur when best HMM-aligned transcript was used for training
-

---

## Future Work

- Recently implemented stochastic gradient training for CRFs
    - Faster training, improved results
  - Work currently being done to extend the model to word recognition
  - Also examining the use of transition functions that use the observation data
-