Seminar

Named Entity Recognition & Domain Adaptation

Jenny, ZHANG

2014.11.26

Advance Data And Programming Technology

Founded in 2010, SJTU. Directed by Prof. Kenny Q. Zhu

Index

1. Introduction on Named Entity Recognition

2. Models for Solving NER

3. Domain Adaptation

4. Conclusion

Index

1. Introduction on Named Entity Recognition

2. Models for Solving NER

3. Domain Adaptation

4. Conclusion

Some NLP Problems

- **Information Extraction**
  - Named entities recognition (NER)
  - Relationships between entities
- **Finding Linguistic Structure**
  - Part-of-speech tagging
  - Parsing
- **Machine Translation**

Named Entity Recognition

- **Named Entity Recognition**(NER) is a subtask of information extraction that seeks to locate and classify elements in text into pre-defined categories: the names of PERSONs, ORGANIZATIONs and LOCATIONs.

- **INPUT:**

  - Jim bought 300 shares of Alibaba Group Holding Limited in September 2014.

- **OUTPUT:**

  - Jim(PER) bought 300 shares of Alibaba(ORG) Group(ORG) Holding(ORG) Limited(ORG) in September 2014.

Application

- More accurate Internet search engines
- General document organization
- Automatic indexing of books
- Preprocessing step to simplify tasks
- For more complex information extraction tasks

Problems in NE Task Definition

- Category definitions are intuitively quite clear, but there are many grey areas.
- Many of these grey area are caused by **metonymy**.
- **Organization vs. Location** : "**England** won the World Cup." vs. "The World Cup took place in **England**."
- **Location vs. Person**: "**Washington** is a state of the United States. " vs. "**Washington** was the first President of the United States."

Index

1. Introduction on Named Entity Recognition

2. Models for Solving NER

3. Domain Adaptation

4. Conclusion

Notation

- **INPUT:** Jim bought 300 shares of Alibaba Group Holding Limited.
- **OUTPUT:** PER O O O O ORG ORG ORG ORG

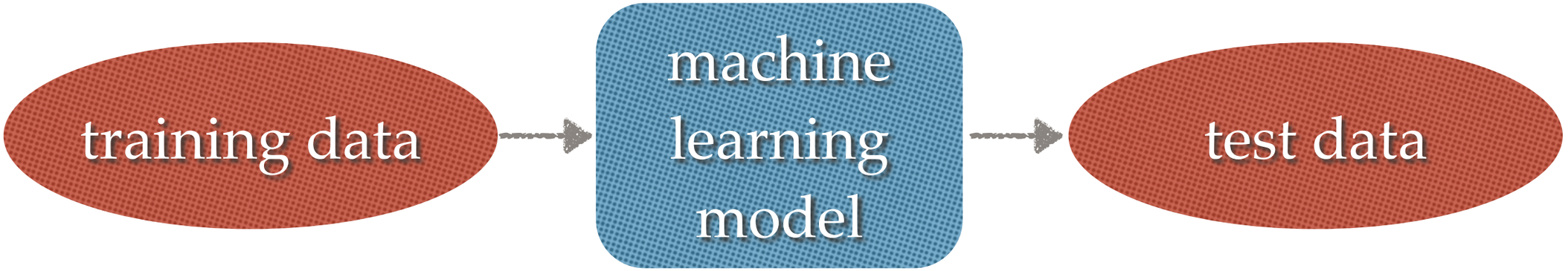
- An **input** sequence: $\mathbf{X} = (x_1, \ldots, x_n)$
- The **output** sequence : $\mathbf{Y} = (y_1, \ldots, y_n)$

$$y_i \in \{PER, LOC, ORG, O\}\ i = 1,2, \ldots, n$$

- Typical goal: Given **X**, predict **Y**,which satisfies:
- $$Y' = \underset{Y}{\operatorname{argmax}}(Y|X) \quad \text{OR} \quad Y' = \underset{Y}{\operatorname{argmax}}(X, Y)$$

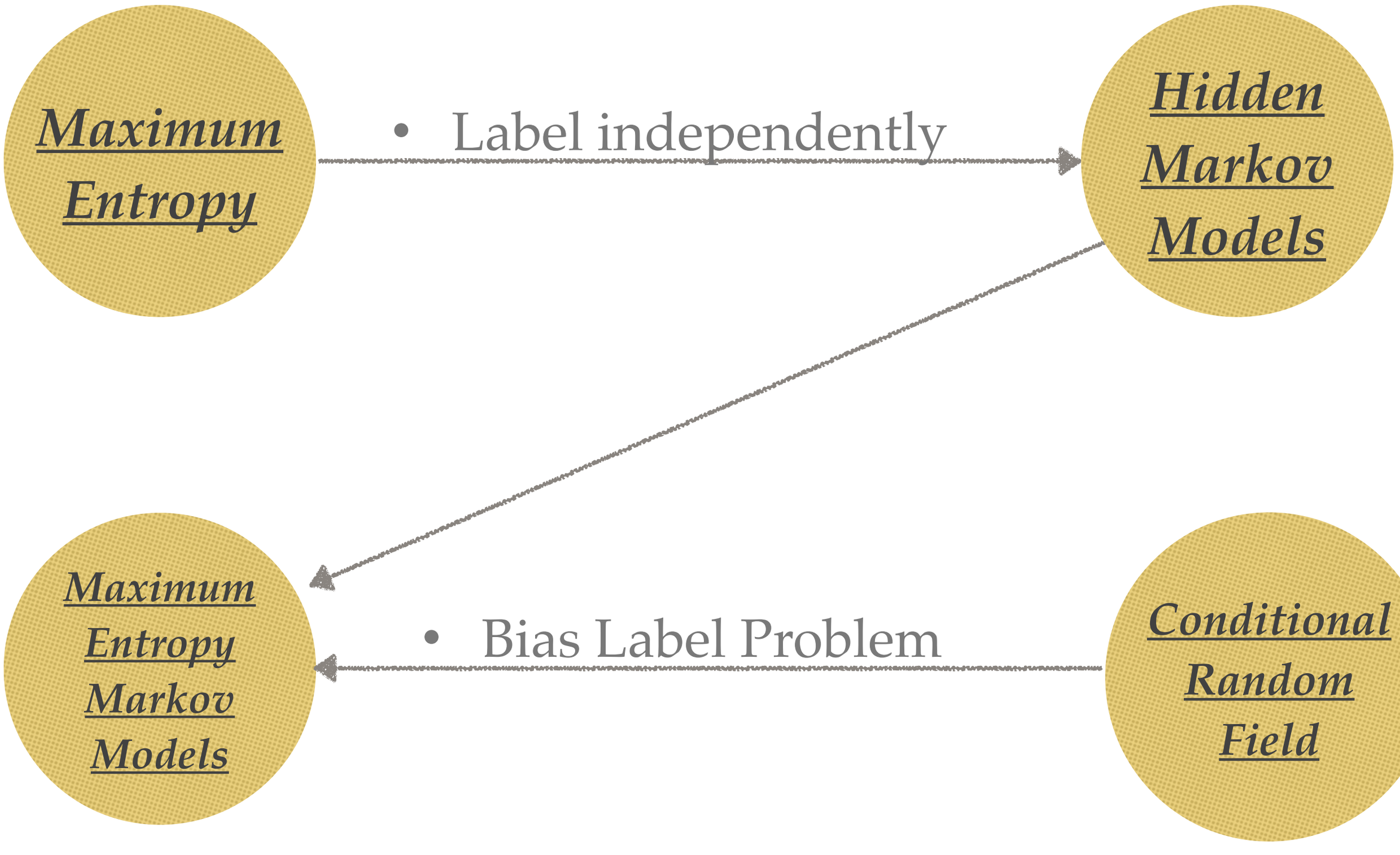
Current Tools

- The current tools: ***machine learning algorithms***

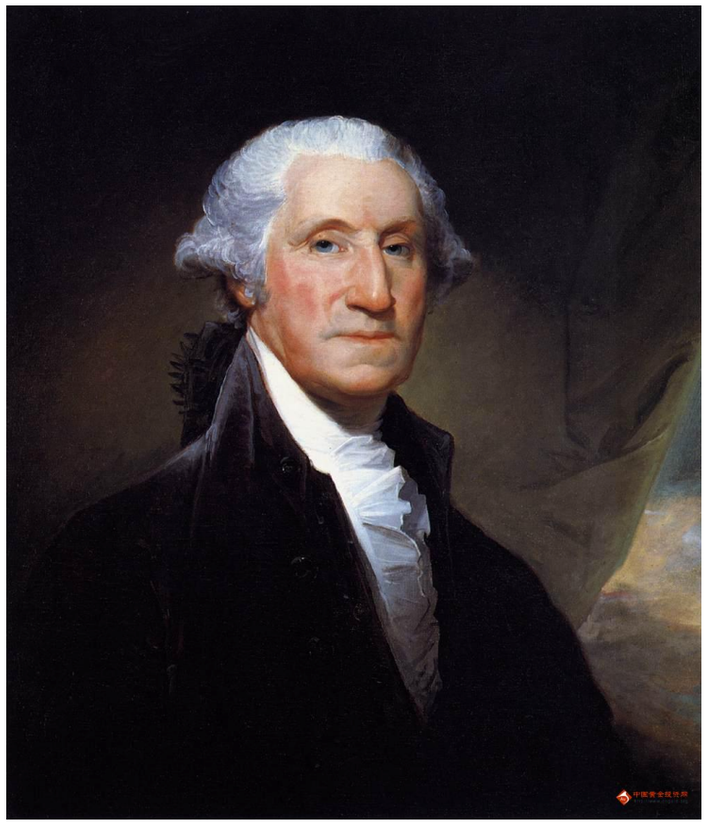


- ***Training and Testing:***
  - A multi-class classification problem: Logistic Regression, SVM

Models


Maximum Entropy
• Label independently
Hidden Markov Models
Maximum Entropy Markov Models
• Bias Label Problem
Conditional Random Field

First Solution: Maximum Entropy Classifier



Washington(PER)

Washington(LOC)

Victoria(PER)

Victoria(LOC)

$p(PER) = 0.6$

$p(Washington = PER) = ?$

Intuition:

$p(Washington = PER) = 0.3$

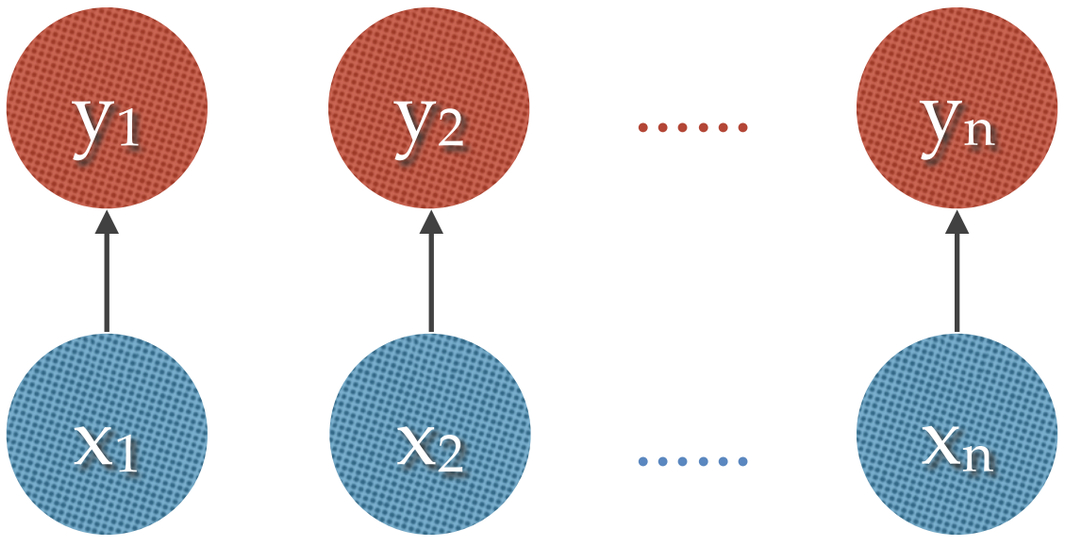
First Solution: MaxEnt Classifier

- ***Principle***:
  - model all that is known and assume nothing about that which is unknown
- ***Definition***:
  - Entropy

$$H(p) = -\sum_{x} p(x) \log p(x)$$

First Solution: Maximum Entropy Classifier

- ***Model***:
  - Conditional model $p(\mathbf{Y} \mid \mathbf{X})$.
- ***Dependency***:
  - The tag probabilities depend only on the current word



- ***Probability:***

$$p(Y|X) = p(y_1, \ldots, y_n|x_1, \ldots, x_n)$$

$$= \prod_{i=1}^{n} p(y_i|x_i)$$

- Berger, A. L., Pietra, Della, S. A., & Pietra, Della, V. J. (1996). **A Maximum Entropy Approach to Natural Language Processing**.

Indicator / Feature Function

- ***Feature functions f(x,y):***

$$f_j(x_i, y_i) = \begin{cases} 1 & if\ y_i = PER\ and\ x_i = Washington \\ 0 & otherwise \end{cases}$$

$$f_j(x_i, y_i) = \begin{cases} 1 & if\ y_i = PER\ and\ the\ first\ letter\ of\ word\ x_i capitalized \\ 0 & otherwise \end{cases}$$

- ***Constraints:***

$$\tilde{p}(f) = p(f)$$

$$\sum_{x,y} \tilde{p}(x,y) f(x,y) = \sum_{x,y} \tilde{p}(x) p(y|x) f(x,y)$$

First Solution: MaxEnt Classifier

- Among the models p agree with the constraints, the maximum entropy philosophy dictates that we select the distribution which is most uniform.

$$p(Y|X) = p(y_1, \ldots, y_n|x_1, \ldots, x_n) = \prod_{i=1}^{n} p(y_i|x_i)$$

- A mathematical measure of the uniformity of a conditional distribution $p(y \mid x)$

$$H(p) = -\sum_{x,y} \tilde{p}(x)p(y|x)\log p(y|x)$$

First Solution: MaxEnt Classifier

- The constrained optimization problem is to:

$$find\ p^* = \underset{p}{\operatorname{argmax}}\, H(p)$$

$$= \underset{p}{\operatorname{argmax}} \left(-\sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x) \right)$$

- we seek to **maximize** $H(p)$ subject to the foll[illegible]ts:

Lagrange Multiplier

1. $p(y|x) \geq 0\ for\ all\ x, y$
2. $\sum_{x} p(y|x) = 1\ for\ all\ x$
3. $\sum_{x,y} \tilde{p}(x) p(y|x)\, f(x,y) = \sum_{x,y} \tilde{p}(x,y)\, f(x,y)$

First Solution: MaxEnt Classifier

- There is a unique, exponential family distribution that meets these criteria.

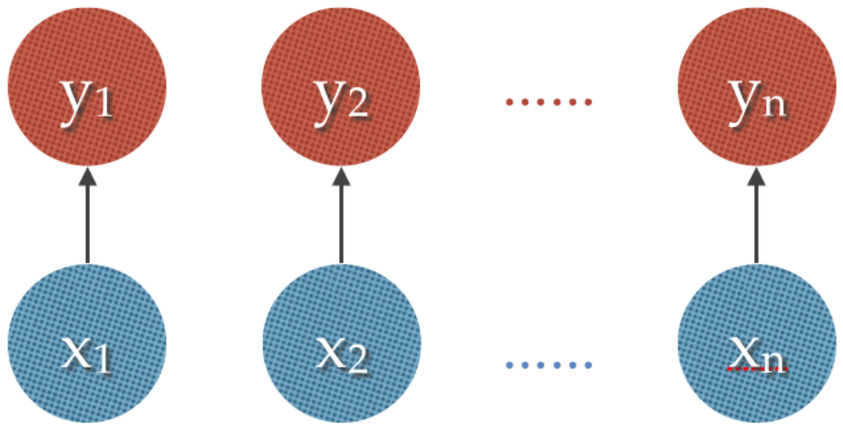
$$p_\lambda(y|x) = \frac{exp\left(\sum_j \lambda_j f_j(x,y)\right)}{Z_\lambda(x)}$$

- *where*

$$Z_\lambda(x) = \sum_y exp\left(\sum_j \lambda_j f_j(x,y)\right)$$

Weaknesses

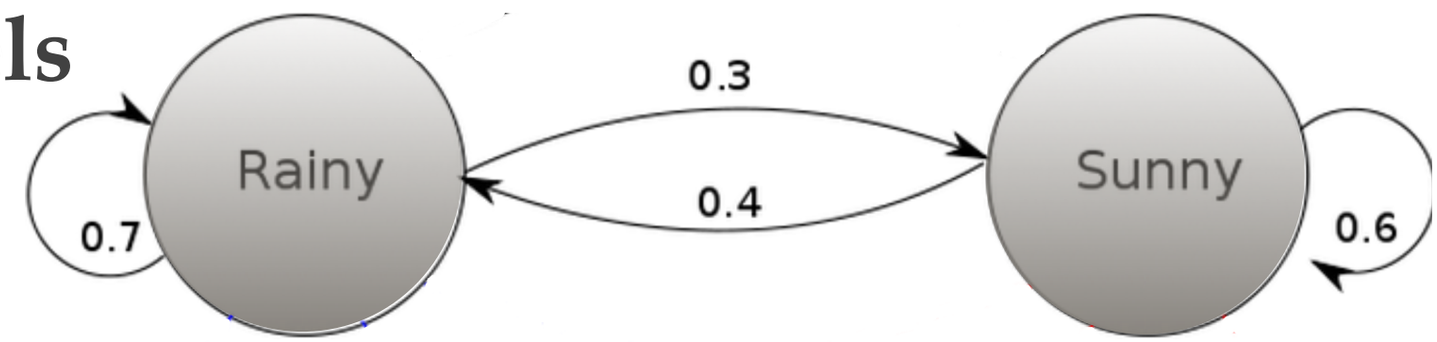
- It makes Output Labels at each position independently!



Second Solution: Hidden Markov Model

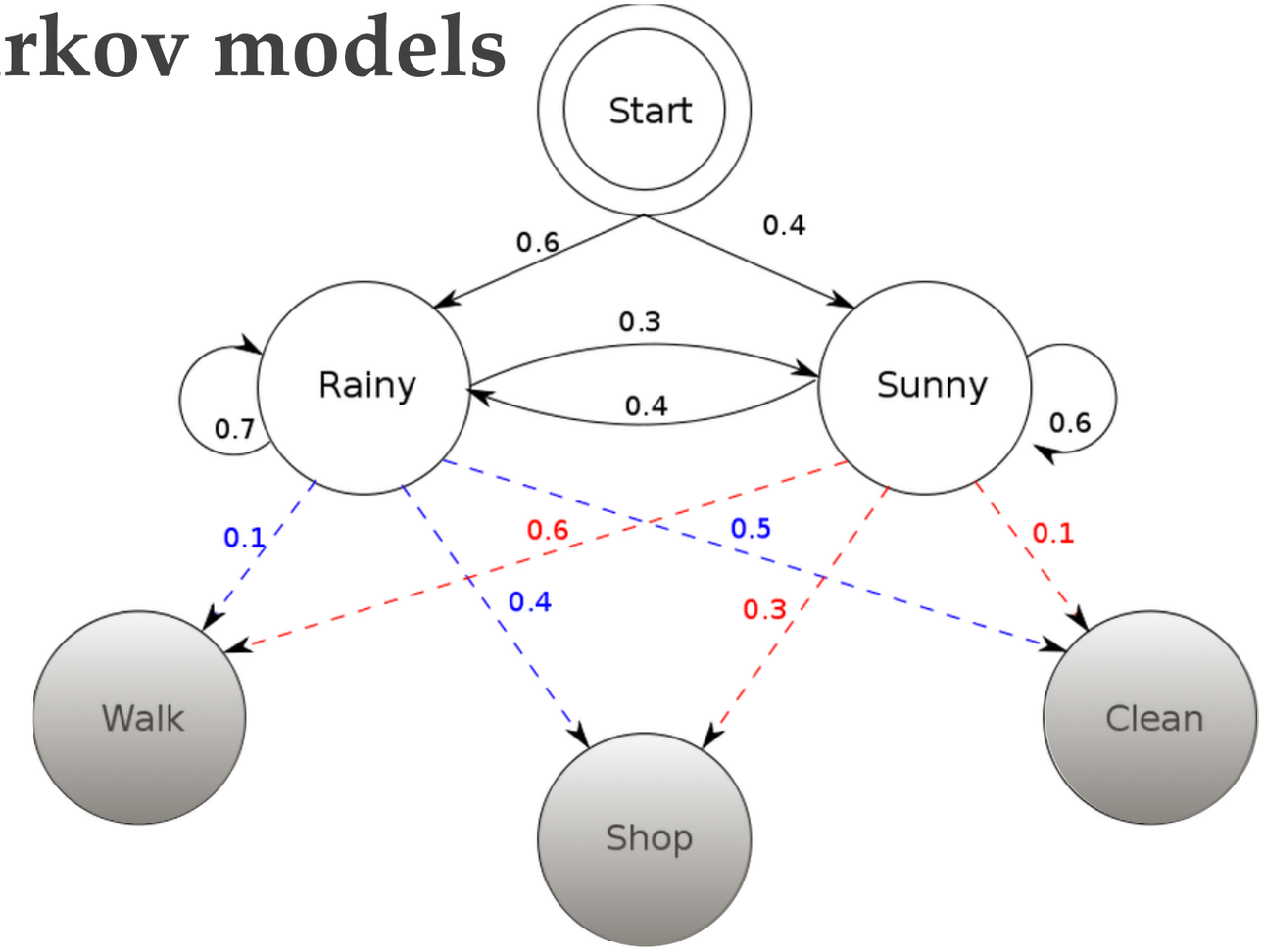
In simpler Markov models

(like a Markov chain)



In Hidden Markov models

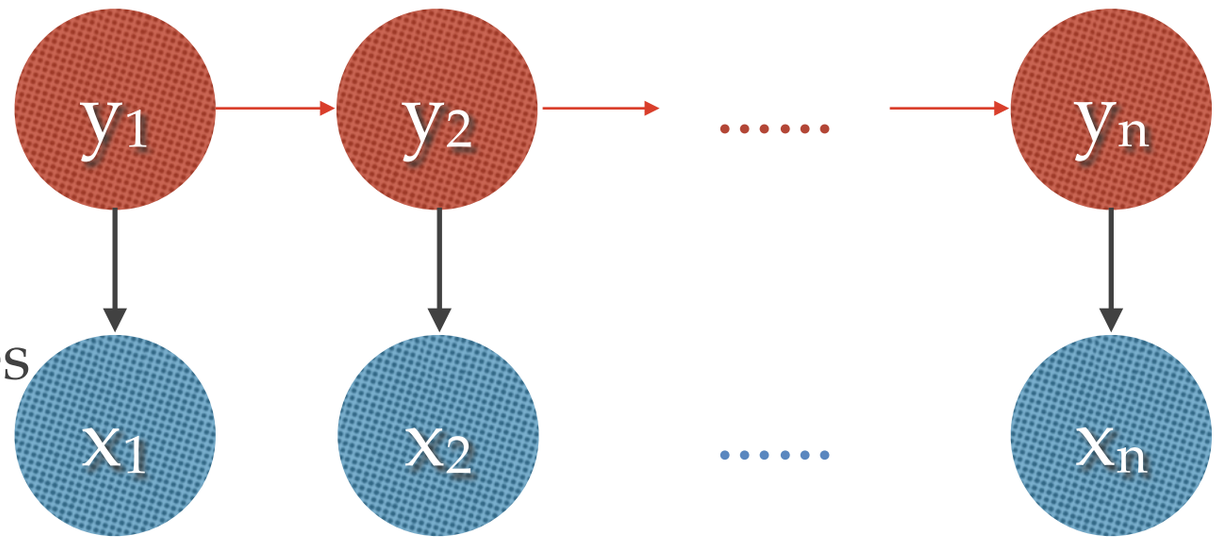
Alice



Bob

Second Solution: HMM

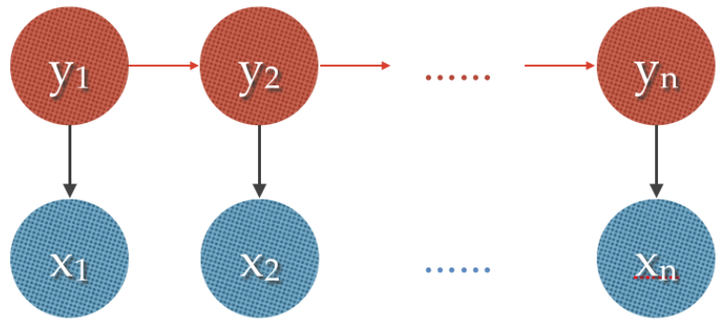
- ***Model***:
  - Assign a joint probability $p(\mathbf{X},\mathbf{Y})$ to paired observation and label sequences
  - The parameters trained to maximize the joint likelihood of train examples



- ***Dependency***:
  - The tag probabilities depend on the current word and the previous tag
- ***Probability:***

$$p(X,Y) = p(y_1)p(x_1|y_1)\prod_{k=2}^{n} p(y_k|y_{k-1})p(x_k|y_k)$$

Second Solution: HMM



- ***Model***:

$$Y' = \underset{y_1, y_2, \ldots, y_n}{\operatorname{argmax}}\, p(y_1)p(x_1|y_1) \prod_{k=2}^{n} p(y_k|y_{k-1})p(x_k|y_k)$$

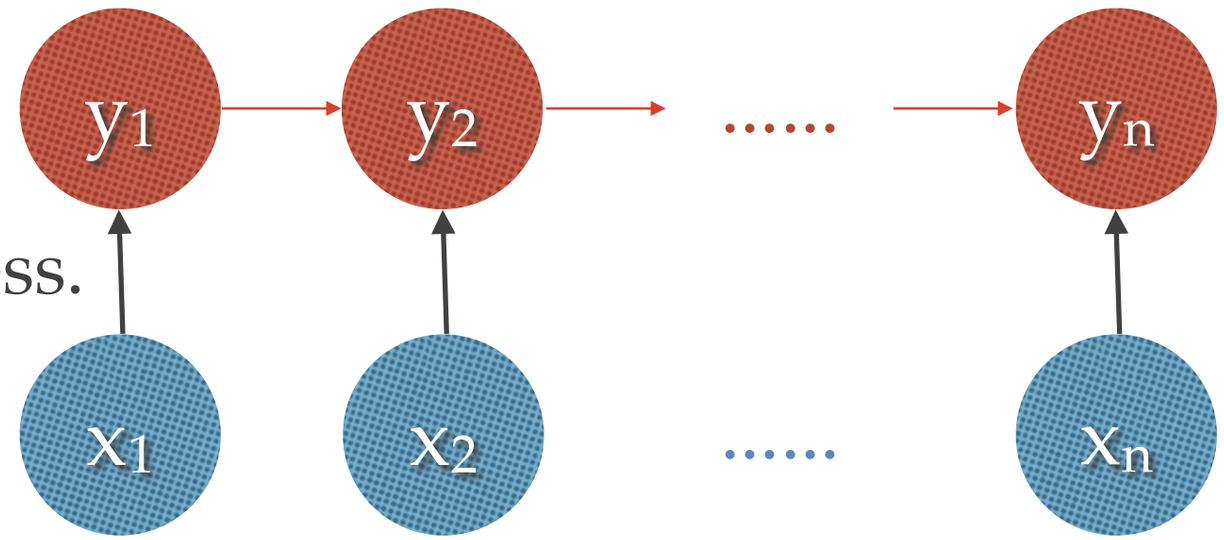
- ***Optimal Labeling Computation***:
  - **Efficient dynamic programming (DP) algorithms that solve these problems are the Forward, Viterbi, and Baum-Welch algorithms respectively.**

Weaknesses

- HMM – Tag and observed word both depend only on previous tag
- Need to account for dependency of tag on observed word
- Need to extract “features” from word & use
- Lack of many overlapping features
- the set of all possible observations is not reasonably enumerable.

Third Solution: Maximum Entropy Markov Model

- ***Model***:
  - Defines a discriminative process.
  - Conditional model $p(\mathbf{Y} \mid \mathbf{X})$.



- ***Dependency:***
  - The tag probabilities depend on the current word and the previous tag
- ***Probability:***

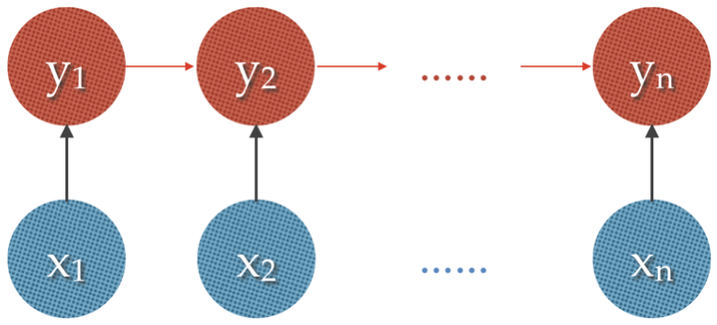
$$p(Y|X) = \prod_{k=1}^{n} p(y_k|y_{k-1}, x_k)$$

- Mccallum, A., Freitag, D., & Pereira, F. (2001). Maximum Entropy Markov Models for Information Extraction and Segmentation.

Third Solution: MEMM

- ***Model:***

$$Y' = \underset{y_1,y_2,\ldots,y_n}{\operatorname{argmax}} \prod_{i=1}^{n} p(y_i|y_{i-1}, x_i)$$



where

$$p(y_i|y_{i-1}, x_i) = \frac{\exp\left(\sum_j \lambda_j f_j(y_{i-1}, x_i, y_i)\right)}{\sum_\gamma \exp\left(\sum_j \lambda_j f_j(y_{i-1}, x_i, \gamma)\right)}$$

- ***Parameter Estimation:***

$$\lambda_1, \lambda_2, \ldots, \lambda_M = \underset{\lambda_1,\lambda_2,\ldots,\lambda_M}{\operatorname{argmax}} \sum_{i=1}^{n} log p_{\lambda_1^M}(y_i|y_{i-1}, x_i)$$

  - **Generalized Iterative Scaling algorithm**

- ***Optimal Labeling Computation:***

$$Y' = \underset{y_1,y_2,\ldots,y_n}{\operatorname{argmax}} \prod_{i=1}^{n} p(y_i|y_{i-1}, x_i)$$

  - **Viterbi algorithm to find the highest probability tag sequence**

Weaknesses

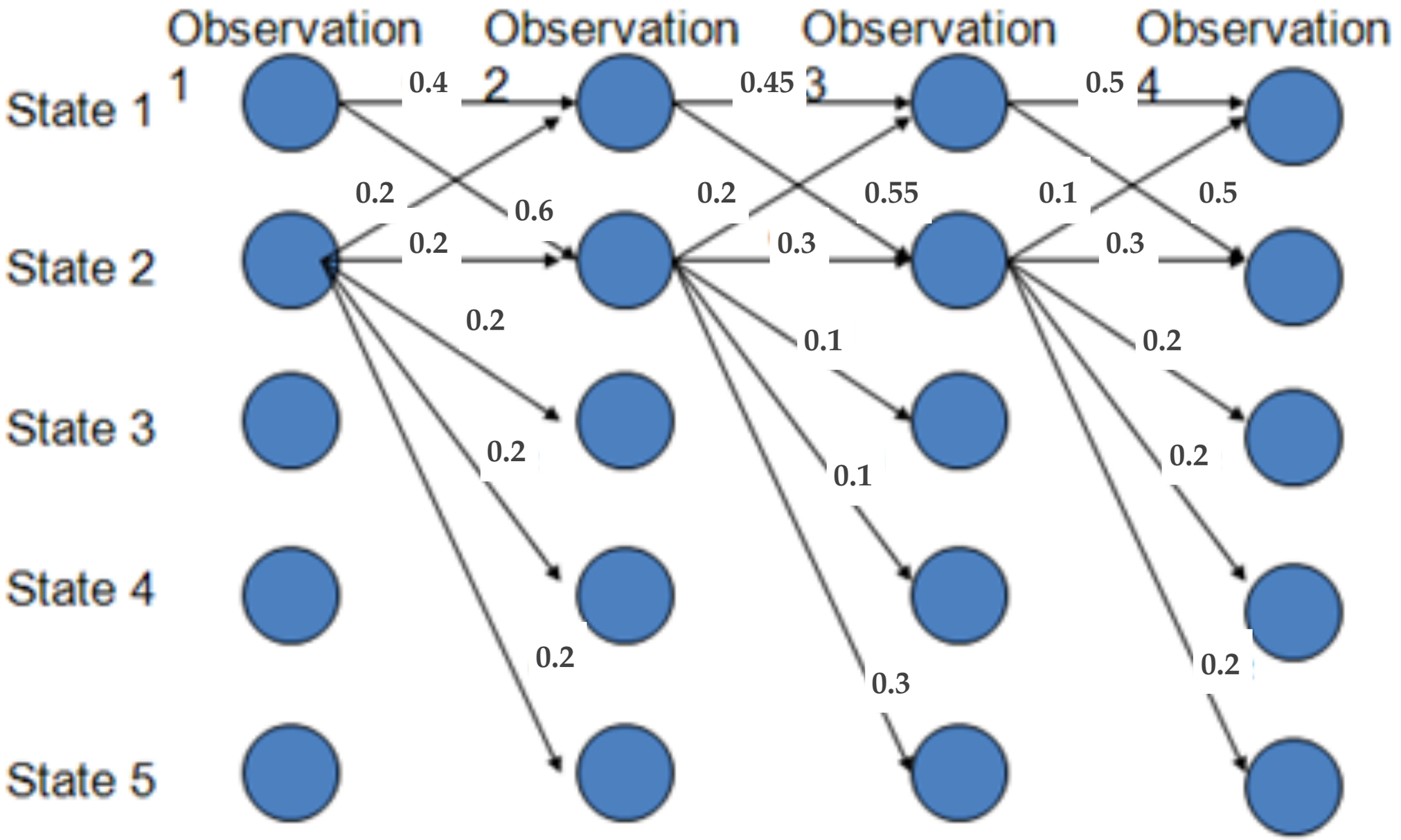
- Bias Label Problem

p(1-1-1-1) = 0.4*0.45*0.5=0.09

p(2-2-2-2) = 0.2*0.3*0.3=0.018

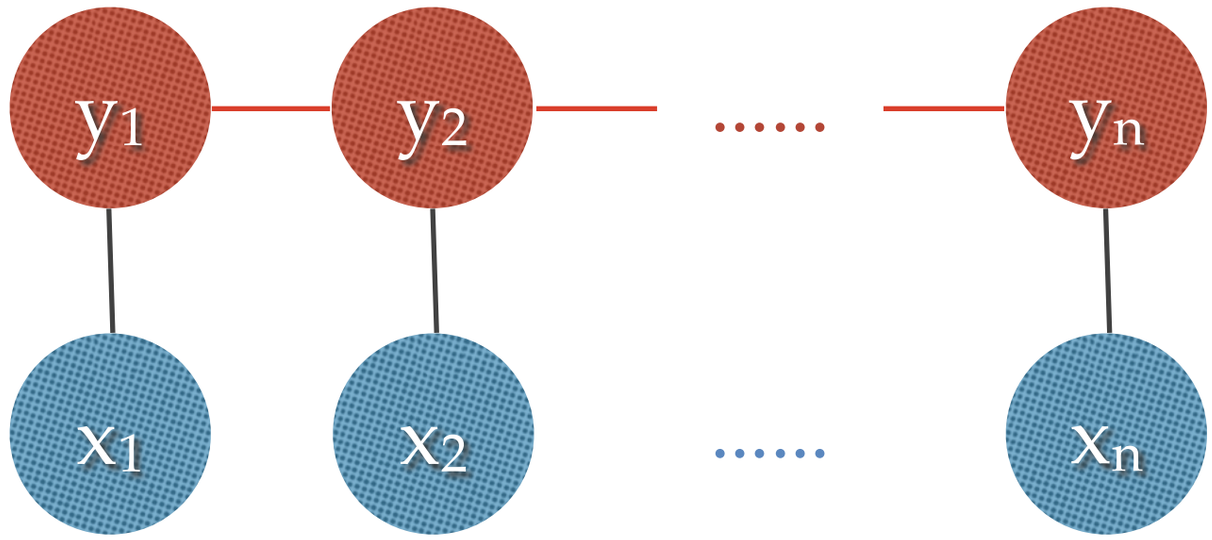
p(1-2-1-2) = 0.6*0.2*0.5=0.06

p(1-1-2-2) = 0.4*0.55*0.3=0.066



Fourth Solution: Conditional Random Field

- ***Model:***
  - Conditionally-trained
  - Undirected graphical model



A standard linear-chain CRF structure

- ***Probability:***

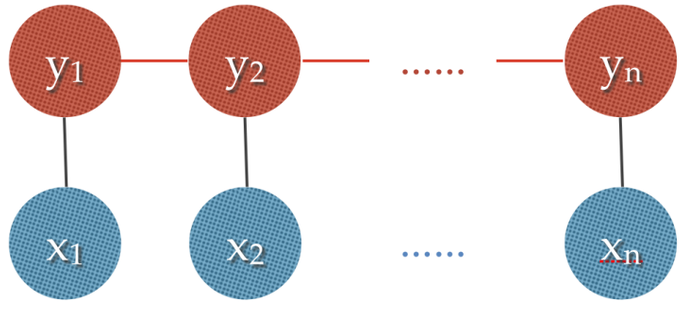
$$p_\lambda(Y|X) \propto exp\left(\sum_{j=1}^{M}\sum_{i=1}^{n}\lambda_j f_j(X, i, y_i, y_{i-1})\right)$$

where there are M feature functions

Fourth Solution: CRF

- ***Model***:

$$Y' = \underset{y_1, y_2, \ldots, y_n}{\operatorname{argmax}} \, exp\left(\sum_{j=1}^{M} \sum_{i=1}^{n} \lambda_j f_j(X, i, y_i, y_{i-1}) \right)$$



A standard linear-chain CRF structure

- ***Parameter Estimation:***
  - **Improved Iterative Scaling algorithm**
- ***Optimal Labeling Computation:***
  - **Viterbi algorithm to find the highest probability tag sequence**

Weakness

- High complexity
- Expensive training cost

Current Status of NER

- Quote from Wikipedia
  - "State-of-the-art NER systems produce near-human performance. For example, the best system entering MUC-7 scored **93.39%** of f-measure while human annotators scored **97.60%** and **96.95%**".
- Wow, that is so **cool**! At the end, we **finally** solved something!

Truth: The NER problem is still **not solved**. Why?

Index

1. Introduction on Named Entity Recognition

2. Models for Solving NER

3. Domain Adaptation

4. Conclusion

The problem:domain over-fitting

- The issues of supervised machine learning algorithms:
  - **Need Labeled Data**
- What people have done for NER: Labeled large amount of data on news corpus.
- However, it is still not enough. The Web contains all kind of data
  - Blogs, Novels, Biomedical Documents, . . .
  - Many domains!
- We might do a good job on **news** domain, but not on **other** domains...

Domain Adaptation

- Many NLP tasks are cast into classification problems
- Lack of training data in new domains
- Domain overfitting

NER Task	Train → Test	F1
to find PER, LOC, ORG from news text	NYT → NYT	0.855
	Reuters → NYT	0.641
to find gene/protein from biomedical literature	mouse → mouse	0.541
	fly → mouse	0.281

Another Example: Visual data

- people, faces
- chair
- tables
- monitor
- book
- scene: office, lab
- action: sitting, talking

From the movie “Transcendence”

Training and Learning

Train…Learn…Test

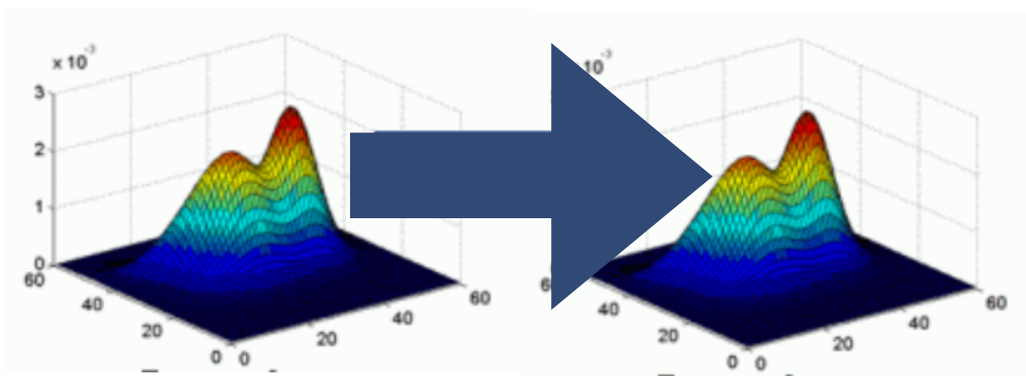
PubFig: Public Figures Face Database

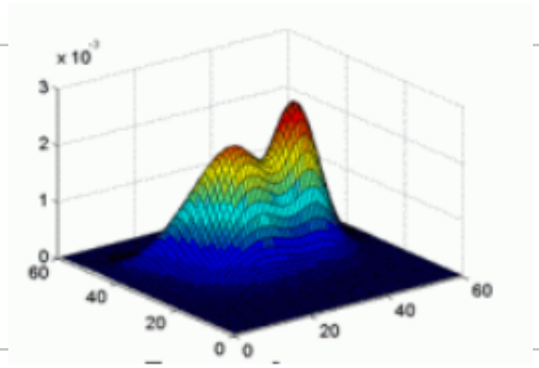
IMAGENET

Indoor Scene Recognition

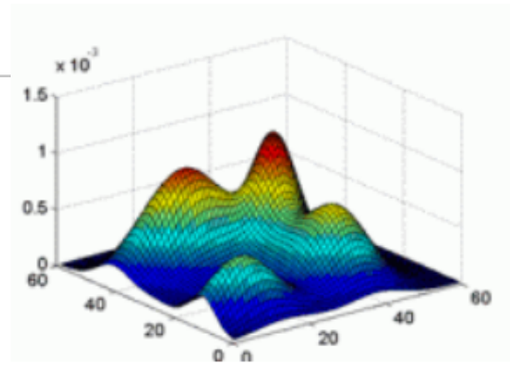
Hollywood Human Actions dataset

- people, faces
- chair
- tables
- monitor
- book
- scene: office, lab
- action: sitting, talking





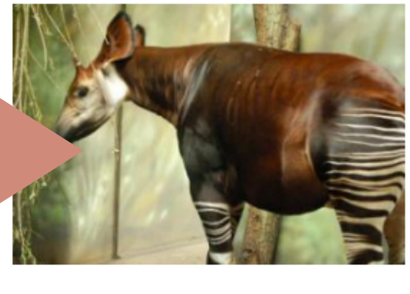
Real Scenarios



[Saenko et al. , ECCV 2010]

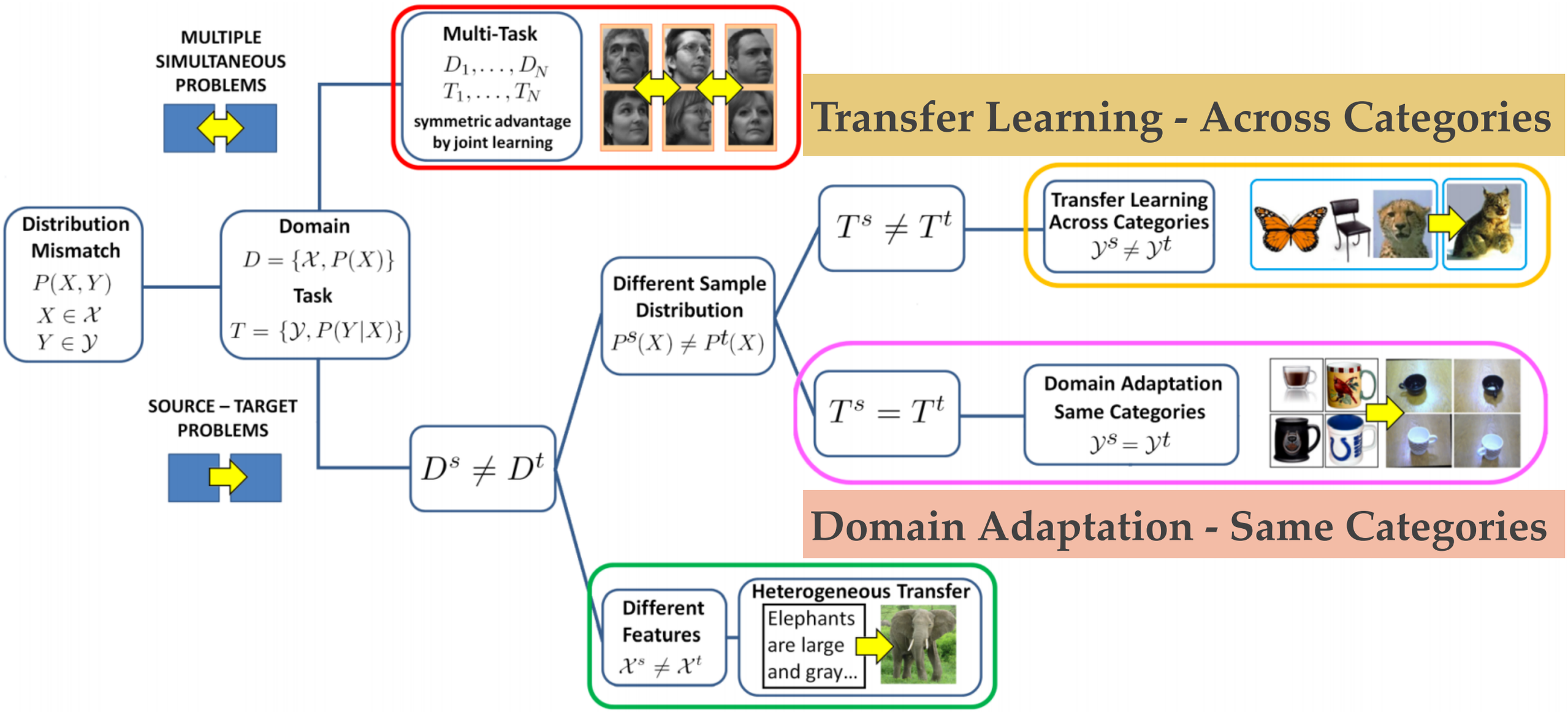
[Torralba, Efros, CVPR 2011]

[Rematas et al, VisDA ICCV 2013]



[Tommasi et al, CVPR 2010]

Domain Adaptation v.s *Transfer Learning*



Terminology

- ***Source Domain***: the domain we know a lot
- ***Target Domain***: the domain we do not know (or know very little) we want to evaluate on target domain

Possible Solution?

1. Don't care (**The current solution**)
   - Bad performance
2. Annotate more data
   - Annotate data for **the new domain**? Need create data for each new domain
3. Build a generic corpus?
   - Wikipedia
   - Good, still not cover all possible solutions. For example, NER for a company.
   - Not sure about the performance
4. Special design algorithms (for each domain)
5. Good, but need to redesign for every domain
6. ***Our Focus***: General purpose adaptation algorithms

Feature-based approaches

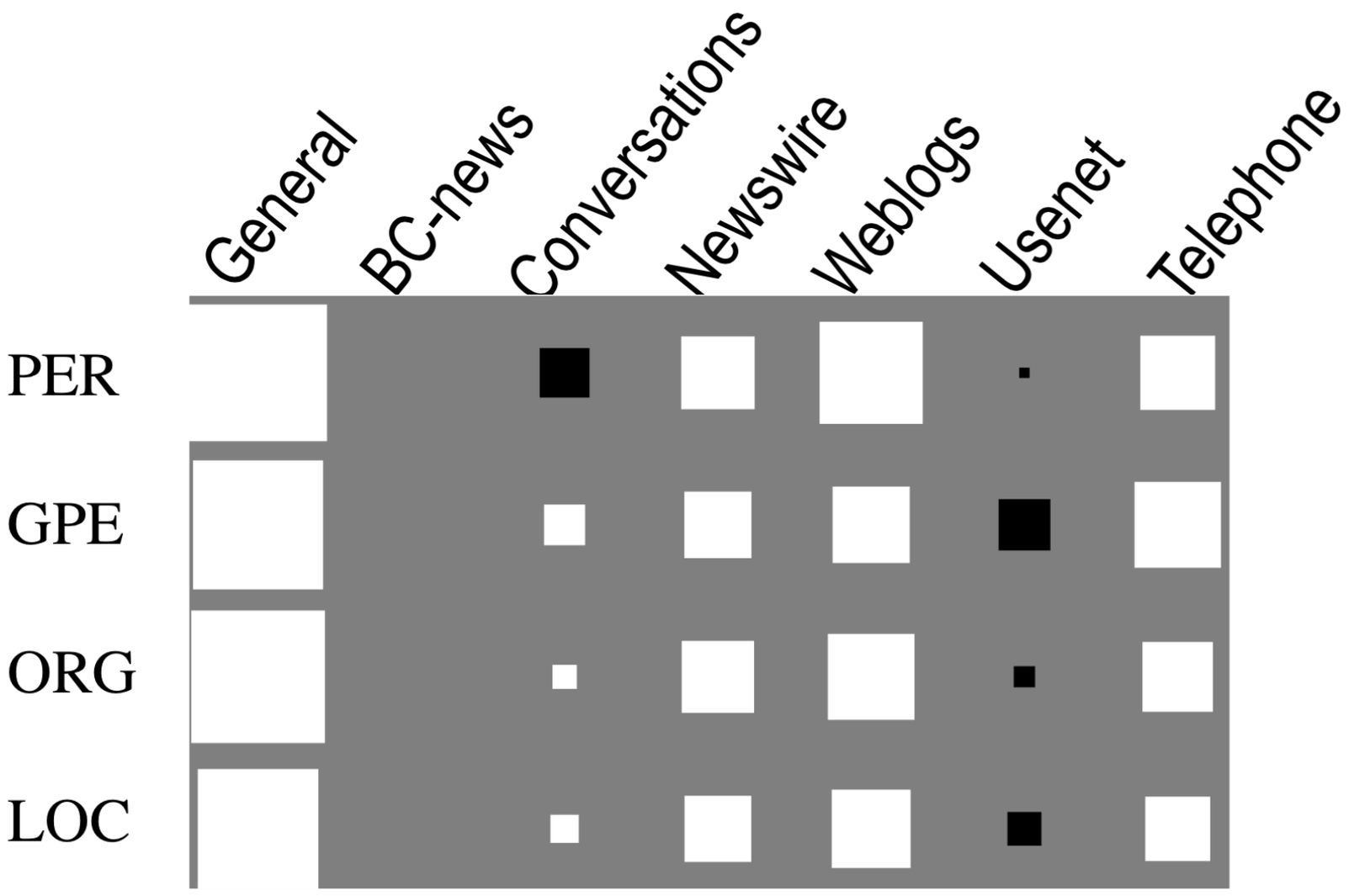


Figure 1: Hinton diagram for feature /Aa+/ at current position.

Hal Daume´ III (2007). Frustratingly Easy Domain Adaptation. In Proc. of ACL

Feature Augmentation approaches

- ***Idea***: take each original feature and make three versions of it
  - **Src-specific**, **tgt-specific**, and **general**
  - *Src data* has only **src-specific** and **general** features;
  - *tgt data* has only **tgt-specific** and **general** features.

In feature-vector lingo:

$\Phi(x) \rightarrow \langle \Phi(x), \Phi(x), 0 \rangle$ (for source domain)

$\Phi(x) \rightarrow \langle \Phi(x), 0, \Phi(x) \rangle$ (for target domain)

- ***Advantage***:
  - Build just one model. No need to build multiple models and then choose the "weights" by cross validation.
  - Easy to extend to multiple source domains

Index

1. Introduction on Named Entity Recognition

2. Models for Solving NER

3. Domain Adaptation

4. Conclusion

Inclusion

- ***NER*** Task is easy to solve, but still remaining **Cross Domain** problems
- ***Sequence Classifier Models*** can solve same domain problems quite well.
- However, an important problem(***Domain adaptation***): We only have limited amount of labeled data for news data but there are so many other domains.

- ***Existing Solutions:***
  - Feature Augmentation
  - Instance Weighting

- ***Many open problems***
  - Better techniques
  - How to combine those techniques
  - Multiple domains adaptation

END

THANK YOU !