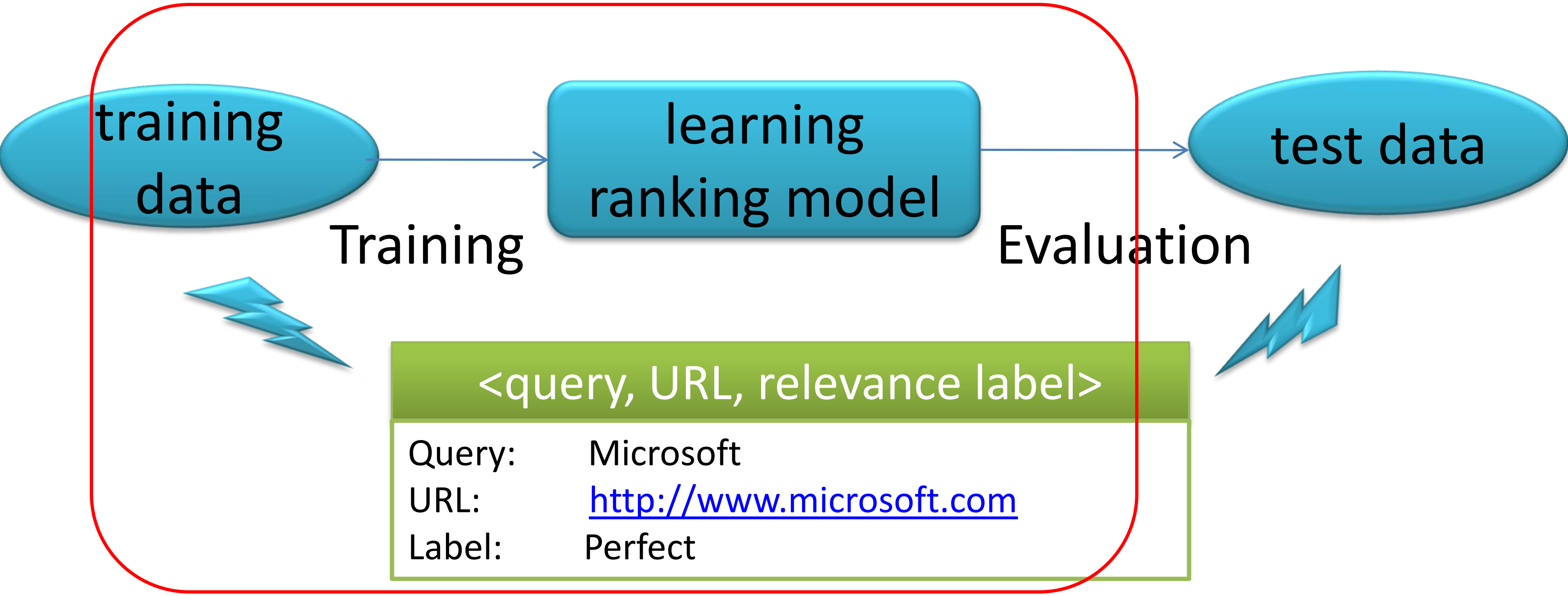
Microsoft Research

WSDM 2010
NYC

Improving Quality of Training Data for Learning to Rank Using Click-Through Data

Jingfang Xu, Chuanliang Chen, Gu Xu,
Hang Li, Elbio Abib

Learning to Rank



- Relevance label:
  - Assigned by human judges
  - Prone to contain errors
- This work: training data quality

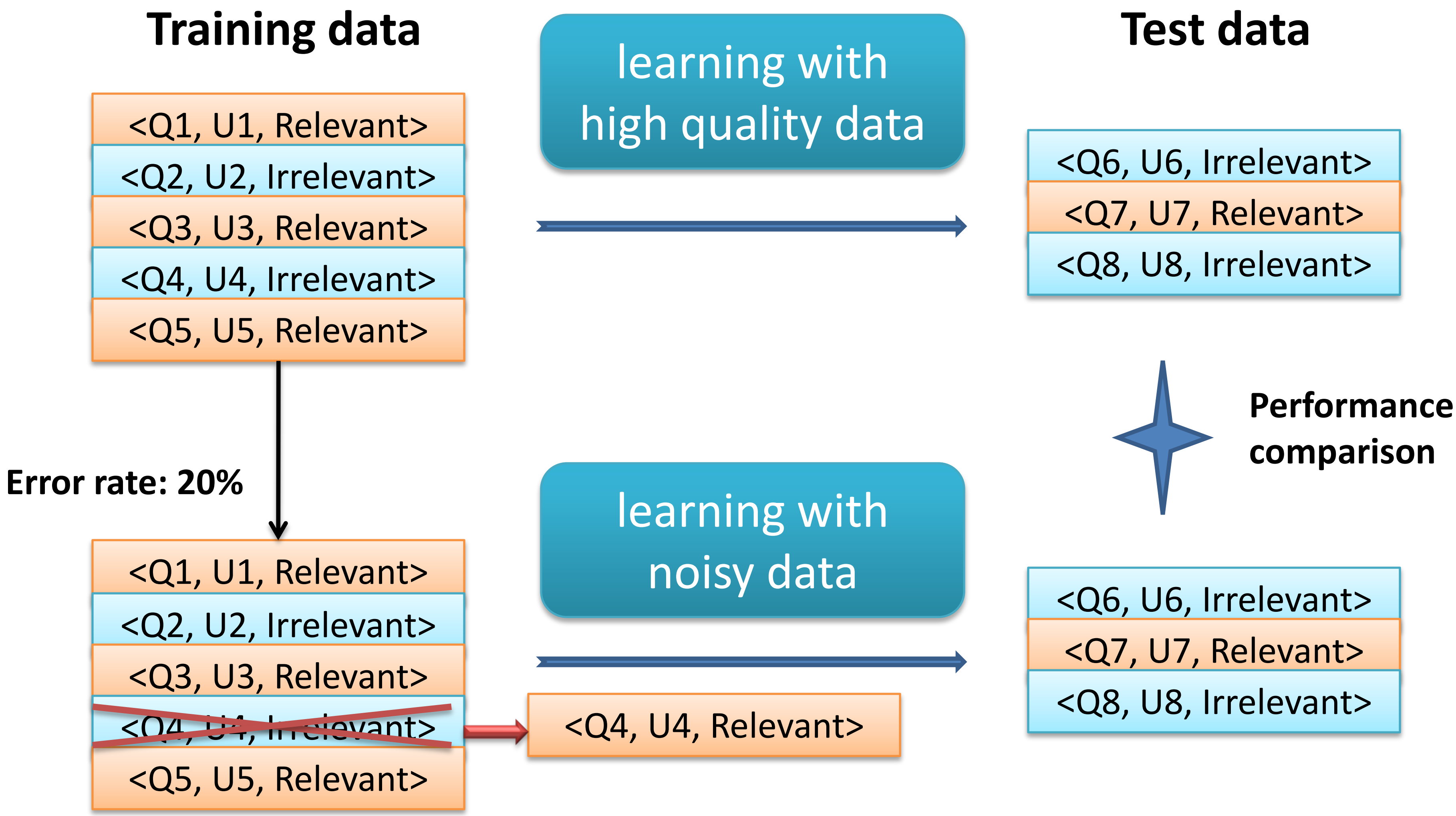
Talk Outline

Training Data Quality

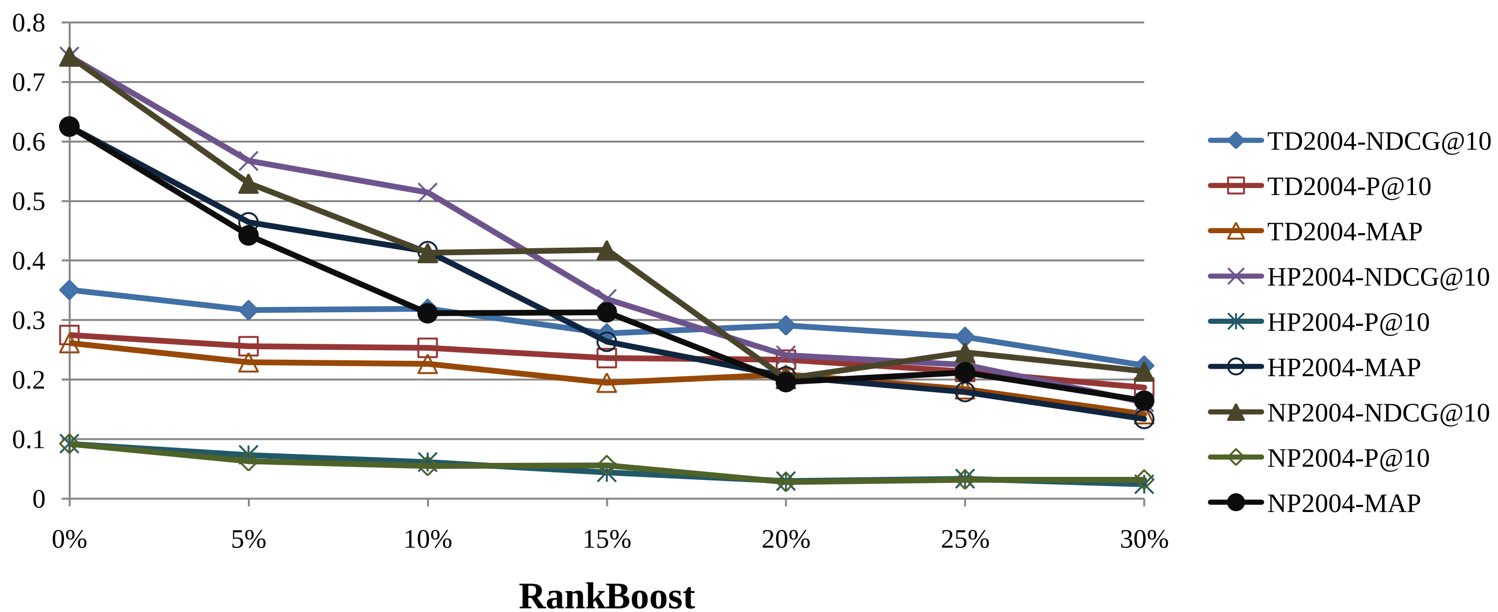
- How Training Data Quality Affects Learning to Rank
- Label Prediction Using Click-through Data
- Improving Training Data Quality Using Click-through Data

How Training Data Quality Affects Learning to Rank

Simulation on LETOR Datasets



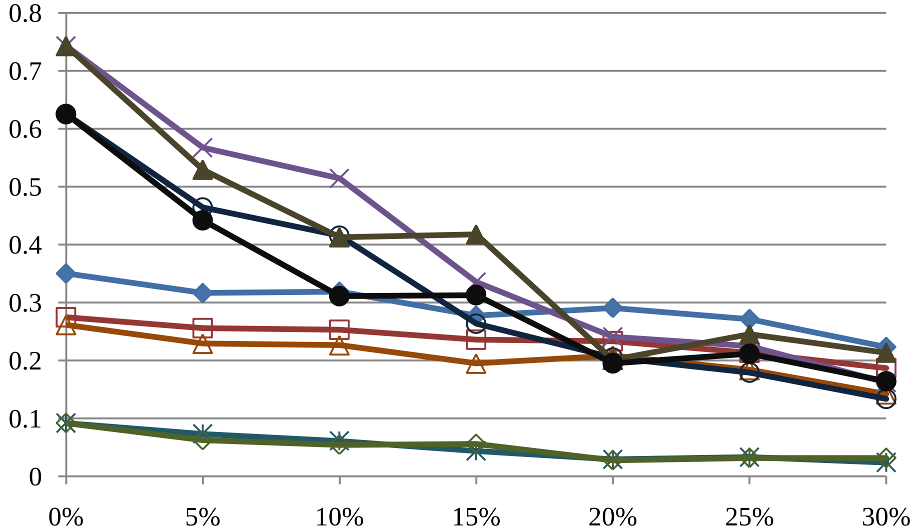
Performance Degrades When Error Rate Increases



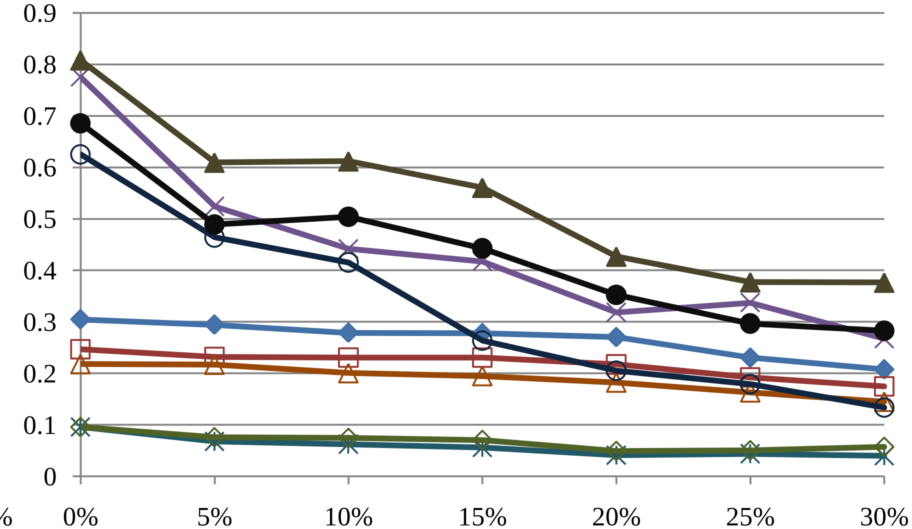
Relative decrease in MAP

Error Rate	TD2004	HP2004	NP2004
5%	12%	36%	29%
30%	46%	70%	74%

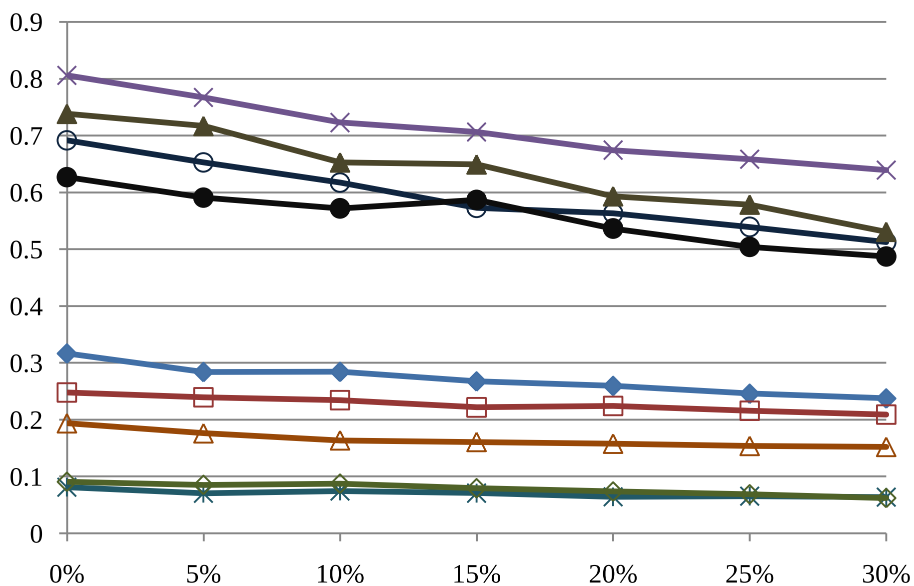
Similar Results on Other Algorithms



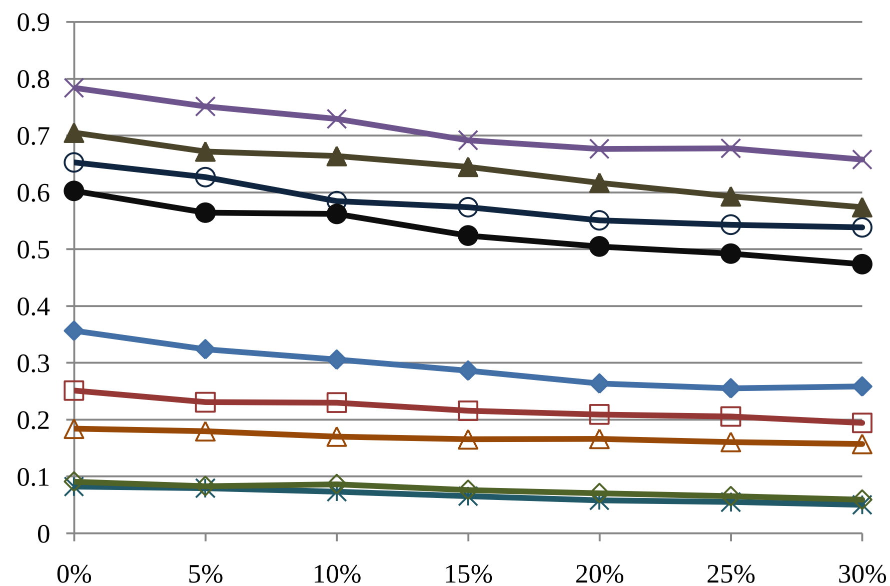
RankBoost



RankSVM



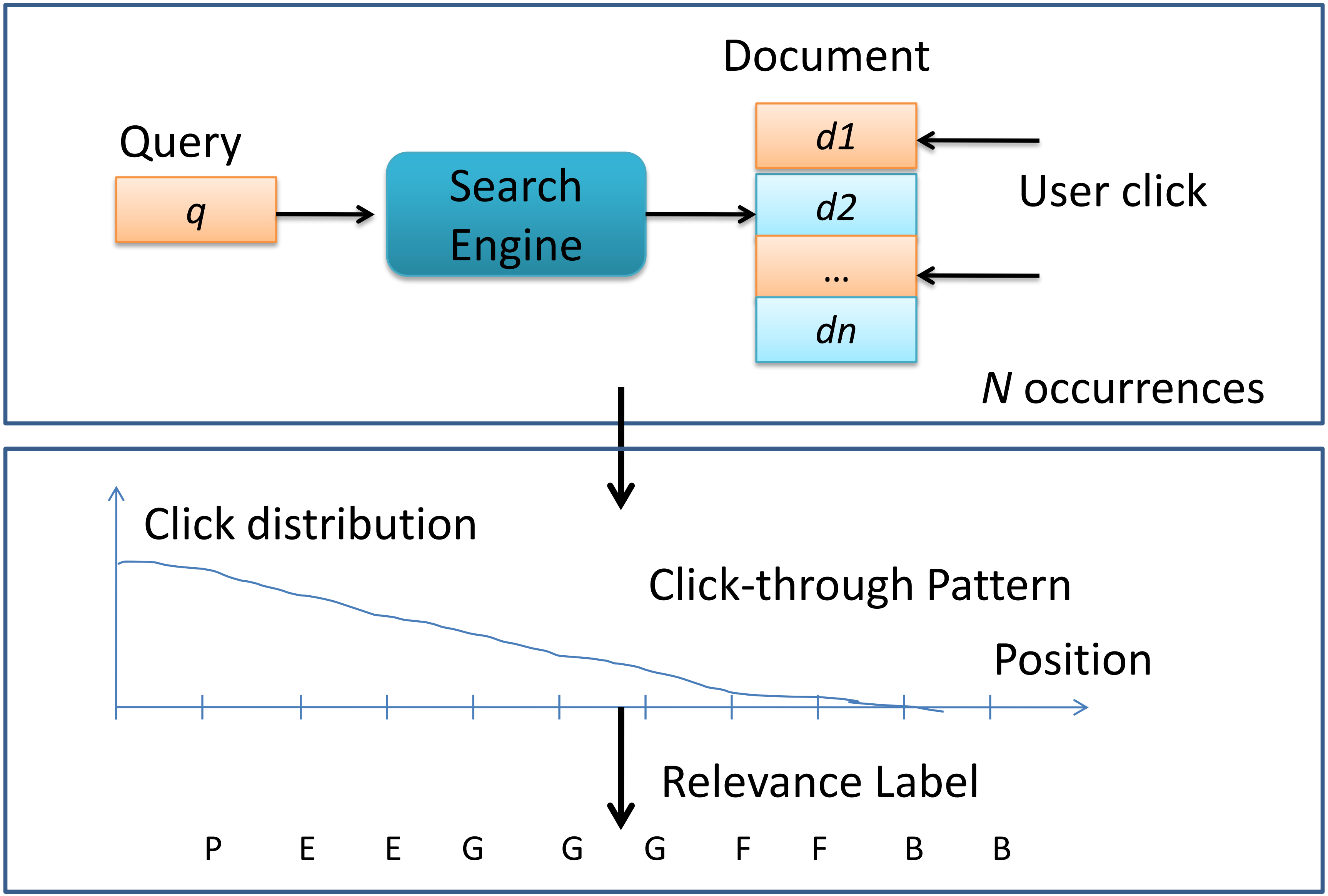
AdaRank



SVMMAP

Label Prediction Using Click-Through Data

Relevance Label Prediction



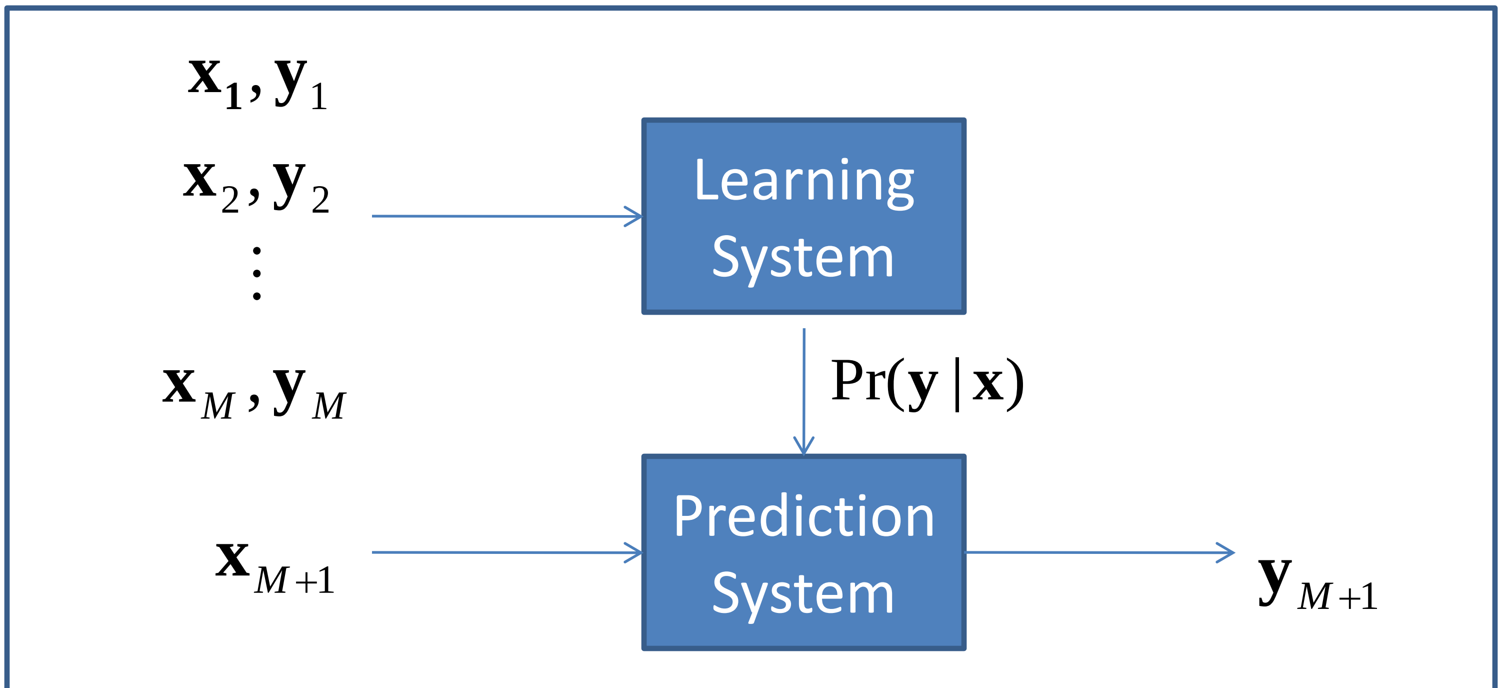
Relevance Label Prediction (cont')

$$\Pr(\mathbf{y}|\mathbf{x}) = p((y_1, y_2, \ldots, y_n) \,|\, (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n))$$

$\mathbf{x}$: click-through pattern

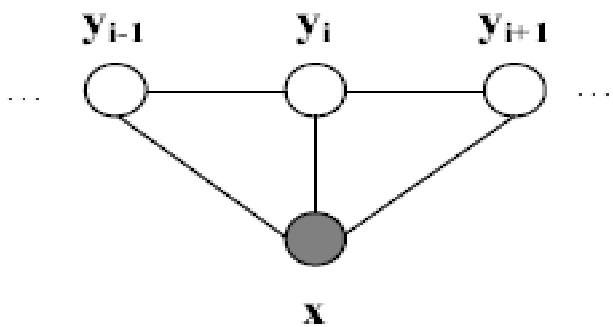
y: relevance label

Supervised Learning Problem
(Carterette & Jones 2007)

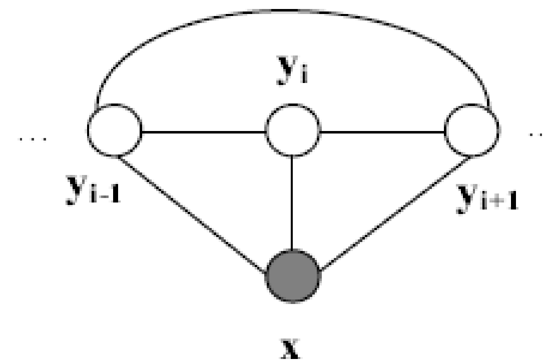


Two Dependency Models

Sequential dependency model



Full dependency model



- Dependency between labels of adjacent document pairs
- Dependency between labels of any document pairs

Two Dependency Models (cont')

Sequential dependency Model

Edge feature

$$\Pr_\theta(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\exp\Big(\sum_{i,k}\lambda_k^i f_k(y_{i-1}, y_i, \mathbf{x}) + \sum_{i,k}\mu_k^i g_k(y_i, \mathbf{x})\Big)$$

Vertex feature

$$Z(\mathbf{x}) = \sum_{\mathbf{y}}\exp\Big(\sum_{i,k}\lambda_k^i f_k(y_{i-1}, y_i, \mathbf{x}) + \sum_{i,k}\mu_k^i g_k(y_i, \mathbf{x})\Big)$$

Position dependent

Full dependency Model

$$\Pr_\theta(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\exp\Big(\sum_{i,j,k}\lambda_k^{i,j} f_k(y_i, y_j, \mathbf{x}) + \sum_{i,k}\mu_k^i g_k(y_i, \mathbf{x})\Big)$$

$$Z(\mathbf{x}) = \sum_{\mathbf{y}}\exp\Big(\sum_{i,j,k}\lambda_k^{i,j} f_k(y_i, y_j, \mathbf{x}) + \sum_{i,k}\mu_k^i g_k(y_i, \mathbf{x})\Big)$$

Learning

Learning = Maximum Likelihood Estimation

$$\theta^{\star} = \arg\max_{\theta} \mathcal{L}(\theta) = \arg\max_{\theta} \sum_{m=1}^{M} \log(\Pr_{\theta}(\mathbf{y}^m | \mathbf{x}^m))$$

Sequential dependency model	Full dependency model
Dynamic programming (L-BFGS)	Solution space is huge and thus calculation of Z(x) is difficult Approximate with Gibbs Sampling

Prediction

Find most likely label sequence

$$\mathbf{y}^{\star} = \arg\max_{\mathbf{y}} \Pr_{\theta^{\star}}(\mathbf{y}|\mathbf{x})$$

Sequential dependency model	Full dependency model
Viterbi algorithm	Solution space is huge Quadratic programming relaxation method (Ravikumar & Lafferty, 2006)

Major Features

Vertex Features	
ClickthroughRate (r_1, r_2)	Whether clickthrough rate of document is in range of $[r_1, r_2]$
DwellTime (t_1, t_2)	Whether time users spend on document is in range of $[t_1, t_2]$
LastClick (p_1, p_2)	Whether probability of document's being the last click of session is in range of $[p_1, p_2]$
Edge Features	
ClickthroughRateDiff (r_1, r_2)	Whether the diff between clickthrough rates of two documents is in range of $[r_1, r_2]$
DwellTimeDiff(t_1, t_2)	Whether the diff between times users spends on two documents is in range of $[t_1, t_2]$
LastClickDiff (p_1, p_2)	Whether the diff between probabilities of two documents' being the last click of a session is in range of $[p_1, p_2]$

Experiment on Label Prediction

Data Set

- Search log of a commercial search engine in Oct. 2008
- 1500 queries, 141 million impressions and 129 million clicks
- Query-document pairs judged by 3 well-trained judges
- 900 queries for training, 600 queries for testing

Baseline method

- Non-dependency model (Carterette & Jones 2007)
- $\Pr(\mathbf{y}|\mathbf{x}) = p(y_1|x_1)p(y_2|x_2)\ldots p(y_n|x_n)$

Evaluation measure

- Correlation between predicted labels and human labels

Experimental Result on Label Prediction

Comparison between Three Methods

Model	Correlation	Improvement
NDM	0.64	-
SDM	0.69	+7.8% *
FDM	0.74	+15.6%*

NDM: non-dependency model

SDM: sequential dependency model

FDM: full dependency model

- SDM and FDM outperform NDM
  - Considering conditional dependency is necessary
- FDM outperforms SDM
  - Increasing scope of dependency is necessary

Improving Training Data Quality Using Click-through Data

Labeling Error Creation

Judgment Error

Random error

- Caused by careless miss
- Equally change to other labels

Real error

- Caused by misunderstanding/ low proficiency
- More likely change to close labels
- Estimated from Mturk (low quality judgment)

Confusion Matrix estimated from Mturk

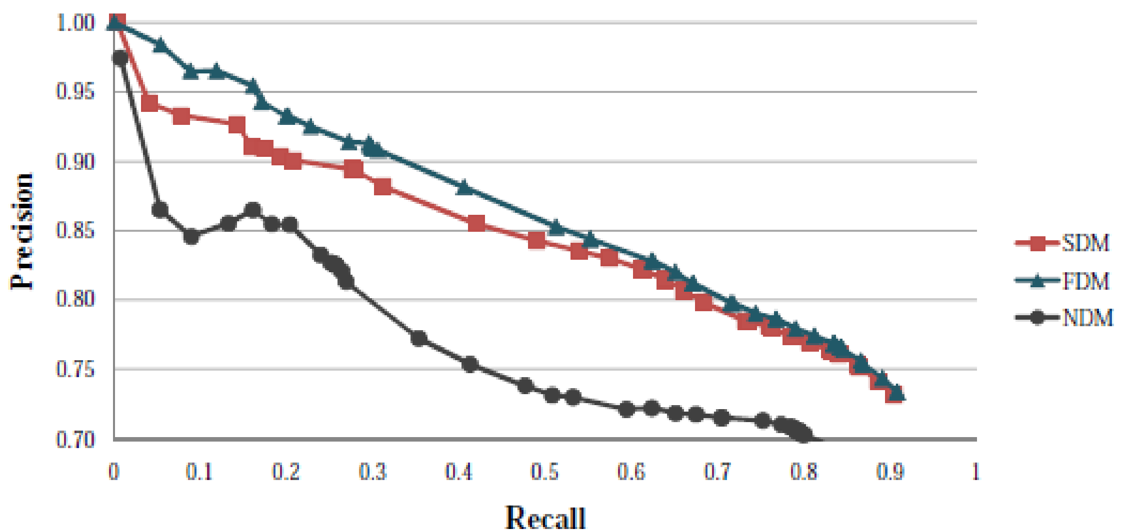
	Perfect	Excellent	Good	Fair	Bad
Perfect	55%	19%	15%	9%	2%
Excellent	10%	26%	26%	26%	12%
Good	7%	11%	25%	34%	23%
Fair	4%	9%	31%	23%	33%
Bad	5%	3%	8%	20%	64%

Labeling Error Detection

Detection Method

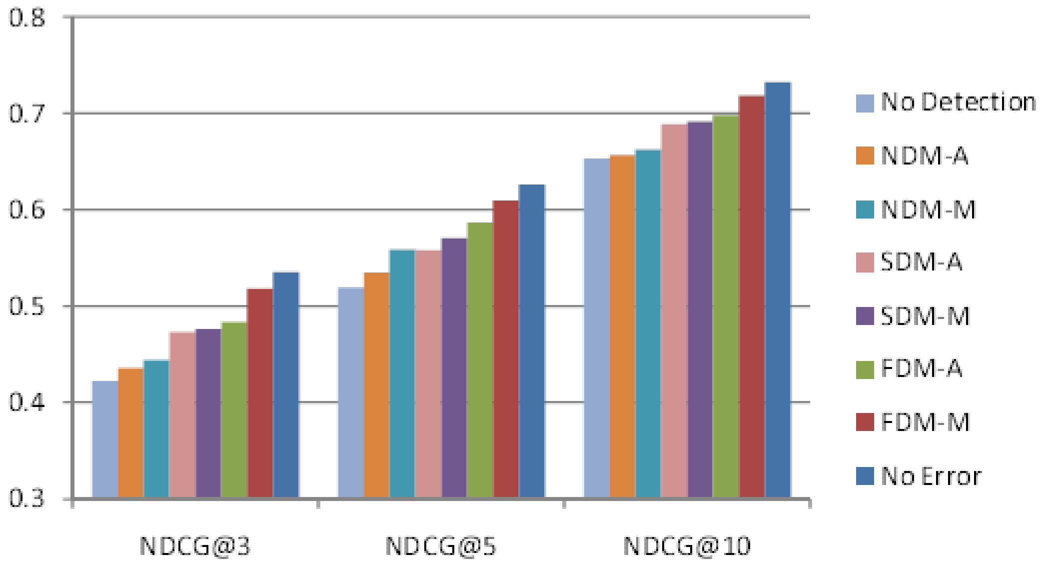
Predict labels using click-through data
If P(predicted label|current label)>threshold,
then detect as error

Experimental Result

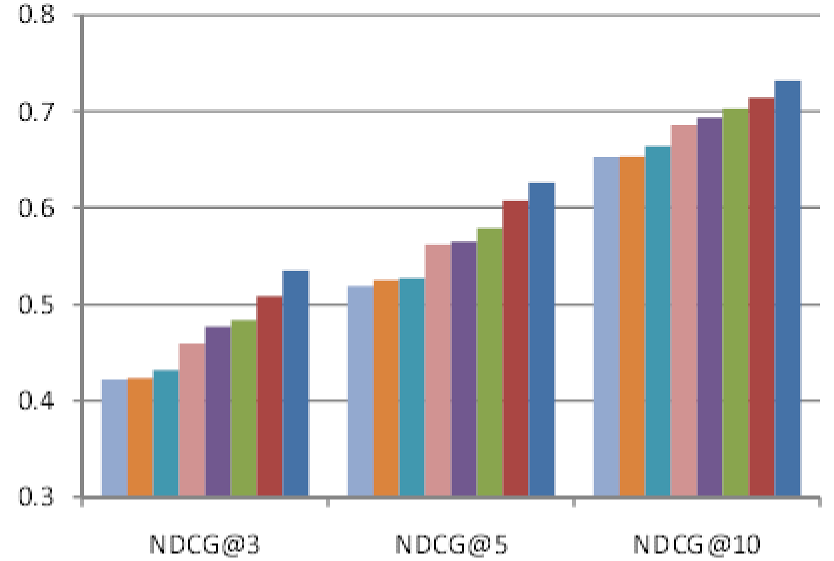


- FDM and SDM outperform NDM
- FDM outperforms SDM

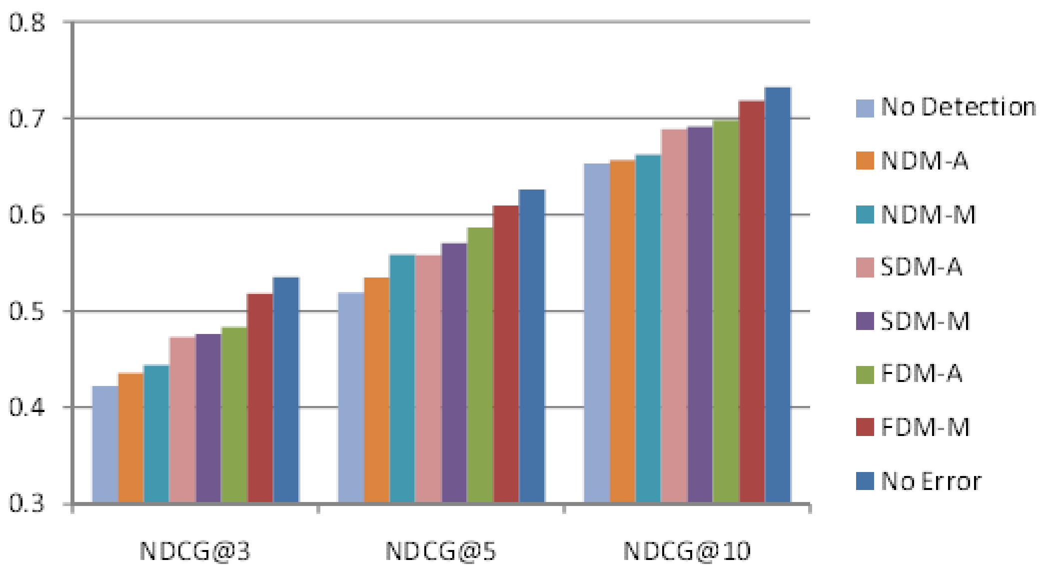
Experimental Results on Labeling Error Detection



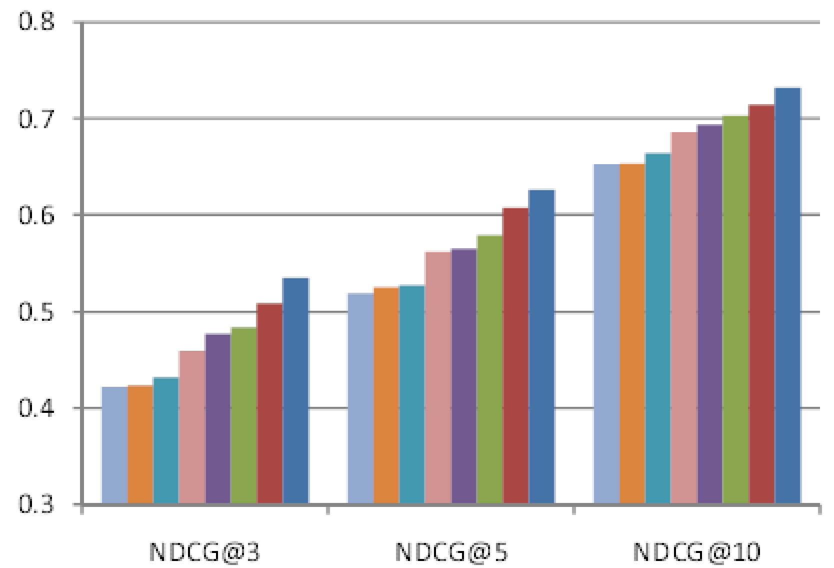
RankSVM (detection precision = 0.7)



RankSVM (0.8)



SVM-MAP (detection precision = 0.7)



SVM-MAP (0.8)

Conclusion

- Labeling errors in training data significantly degrade performance of learning to rank
- Automatically predicting relevance labels using click-through data
  - Sequential dependency model
  - Full dependency model
- Error correction significantly improves performance of learning to rank

Thank you!