A Unified and Discriminative Model for Query Refinement

Jiafeng Guo[1], Gu Xu[2], Xueqi Cheng[1],Hang Li[2]
[1]*Institute of Computing Technology, CAS, China*
[2]*Microsoft Research Asia, China*

Outline

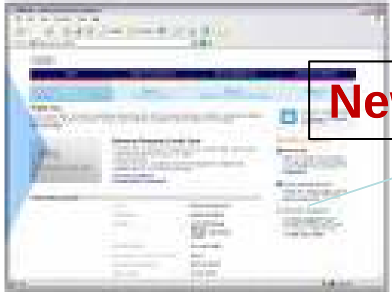
- Motivation
- Our Approach
- Experimental Results
- Conclusion

Outline

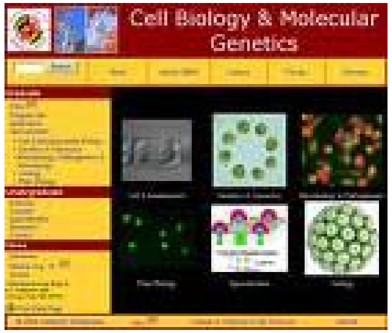
- Motivation
- Our Approach
- Experimental Results
- Conclusion

Introduction

Web Page



New York Times



Information Retrieval

Word Mismatch

Search Query
university of california
kelley blue book
prison break
best movie download
free music
free online games
ny times
……
……

Cont'

ill-formed queries

Words to be stemmed:
Stemming
data mine → data mining

Misspelled words:
Spelling Error Correction
sytem number→ system number

Mistakenly merged words:
Word Splitting
nypark→ ny park

Query
Refinement

Mistakenly split words:
Word Merging
on line game → online game

Acronym to be Expanded:
Acronym Expansion
nfs→ need for speed

Phrase to be quoted:
Phrase Segmentation
the office show → "the office" show

Previous Work

- Query Refinement:
  - Spelling error correction:
    - [1] Exploring distributional similarity based query spelling correction (Li et al. ACL ’06)
    - [2] Spelling correction as an iterative process that exploits the collective knowledge of web users (Cucerzan et al. EMNLP ‘04)
    - [3] Learning a spelling error model from search query logs (Ahmad et al. EMNLP ‘05)
    - [4] Improving query spelling correction using web search results (Chen et al. EMNLP

Existing — Separate Tasks, Generative Models → A unified framework, Discriminative Model — Our Goal

  - Query segmentation:
    - [6] Query segmentation for web search (Risvik et al. WWW ‘03)
    - [7] Learning noun phrase query segmentation (Bergsma et al. EMNLP ‘07)

Work	Task	Approach
[1][2][3]	spelling correction	generative
[1][3]	spelling correction	discriminative
[5]	word stemming	generative
[6]	phrase segmentation	generative
[7]	phrase segmentation	discriminative

Cont'

Why unified framework?

Cascaded Model ?

- Various query refinement tasks
  - ✓ Incorporate different tasks easily
- Mutual dependencies between tasks
  - ✓ Address tasks simultaneously to boost accuracy

✗ Ignore the dependencies between the tasks

✗ Accumulate errors through the processes

A case of Query Refinement

Original: Papers on machin learn

Spelling error correction

word stemming

Refined: Papers on "machine learning"

Phrase segmentation

Cont'

Why discriminative model?

- By nature a structured prediction problem
  - ✓ Enjoy all the merits of discriminative learning
  - ✗ A direct application of existing models would not work

Conditional Random Fields for Query Refinement (CRF-QR)

A case of Query Refinement

Original: Papers on machin learn

Spelling error correction

word stemming

Refined: Papers on "machine learning"

Phrase segmentation

Outline

- Motivation
- Our Approach
- Experimental Results
- Conclusion

Our Approach

A case of Query Refinement

Original: Papers on machin learn

Spelling error correction

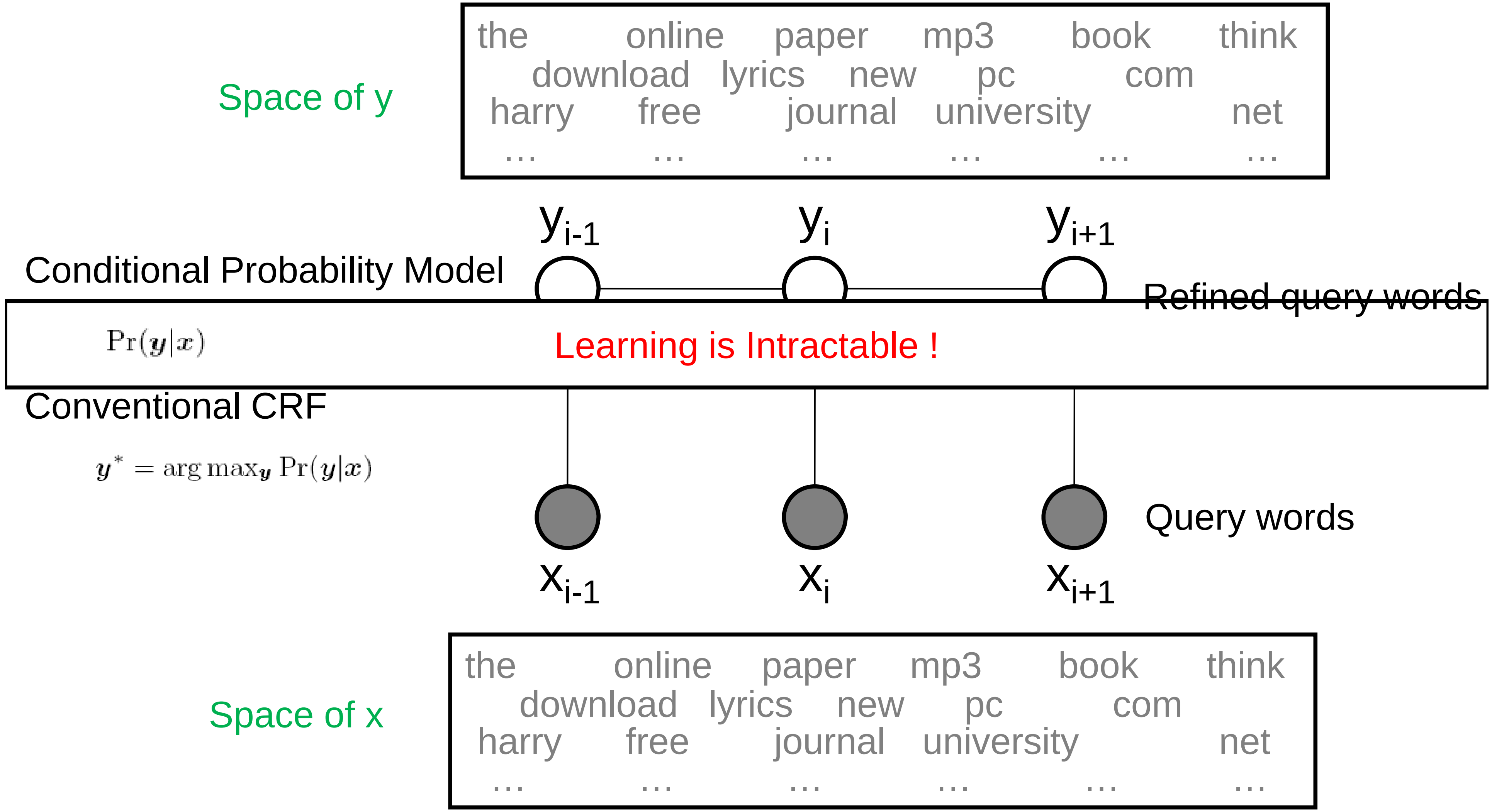
word stemming

Refined: Papers on “machine learning”

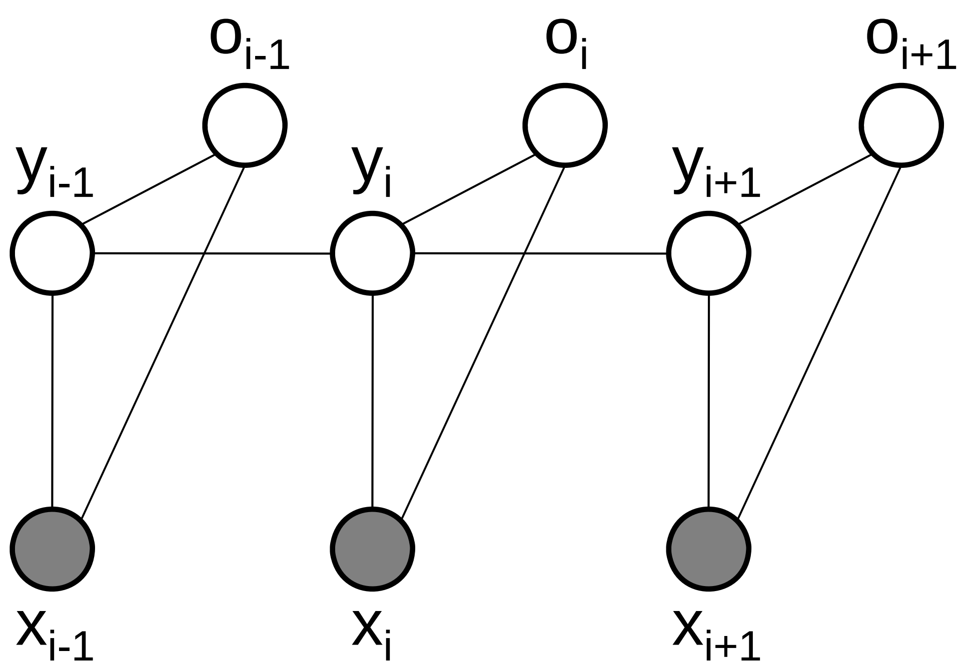
Phrase segmentation

Structured Prediction problem

Conventional CRF



CRF-QR Basic Model



Introducing Refinement Operations

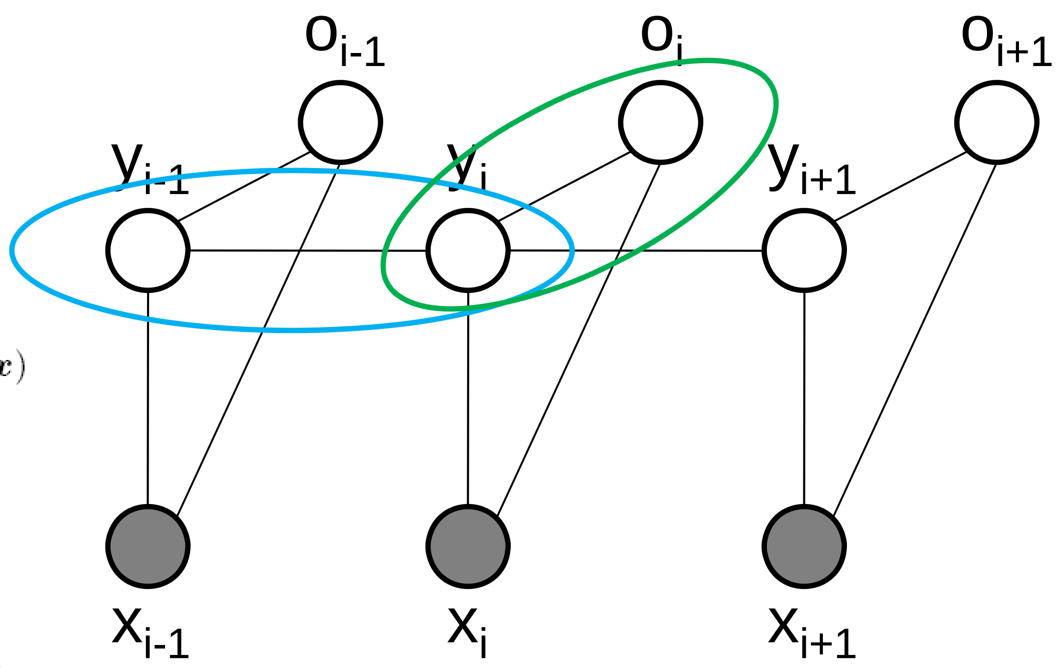
Refinement Operations

Task	Operation	Description
Spelling Error Correction	Deletion	Delete a letter in the word
	Insertion	Insert a letter into the word
	Substitution	Replace a letter in the word with another letter
	Transposition	Switch two letters in the word
Word Splitting	Splitting	Split one word into two words
Word Merging	Merging	Merge two words into one word
Phrase Segmentation	Begin	Mark a word as beginning of phrase
	Middle	Mark a word as middle of phrase
	End	Mark a word as end of phrase
	Out	Mark a word as out of phrase
Word Stemming	+s/-s	Add or Remove suffix `-s'
	+ed/-ed	Add or Remove suffix `-ed'
	+ing/-ing	Add or Remove suffix `-ing'
Acronym Expansion	Expansion	Expand acronym

Conditional Function

$$y^* = \arg\max_{\boldsymbol{y}} \Pr(\boldsymbol{y}|\boldsymbol{x})$$

$$y^* o^* = \arg\max_{\boldsymbol{yo}} \Pr(\boldsymbol{y}, \boldsymbol{o}|\boldsymbol{x})$$



Conditional Function

$$\Pr(\boldsymbol{y}, \boldsymbol{o}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{i=1}^{n} \phi(y_{i-1}, y_i)\phi(y_i, o_i, \boldsymbol{x})$$

Potential Function

$$\begin{aligned} \phi(y_{i-1}, y_i) &= \exp(\sum_k \lambda_k f_k(y_{i-1}, y_i)) \\ \phi(y_i, o_i, \boldsymbol{x}) &= \exp(\sum_k \lambda_k h_k(y_i, o_i, \boldsymbol{x})) \end{aligned}$$

Basic CRF-QR model

$$\Pr(\boldsymbol{y}, \boldsymbol{o}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp(\sum_{i=1}^{n} (\sum_k \lambda_k f_k(y_{i-1}, y_i) + \sum_k \lambda_k h_k(y_i, o_i, \boldsymbol{x})))$$

Function of Operations

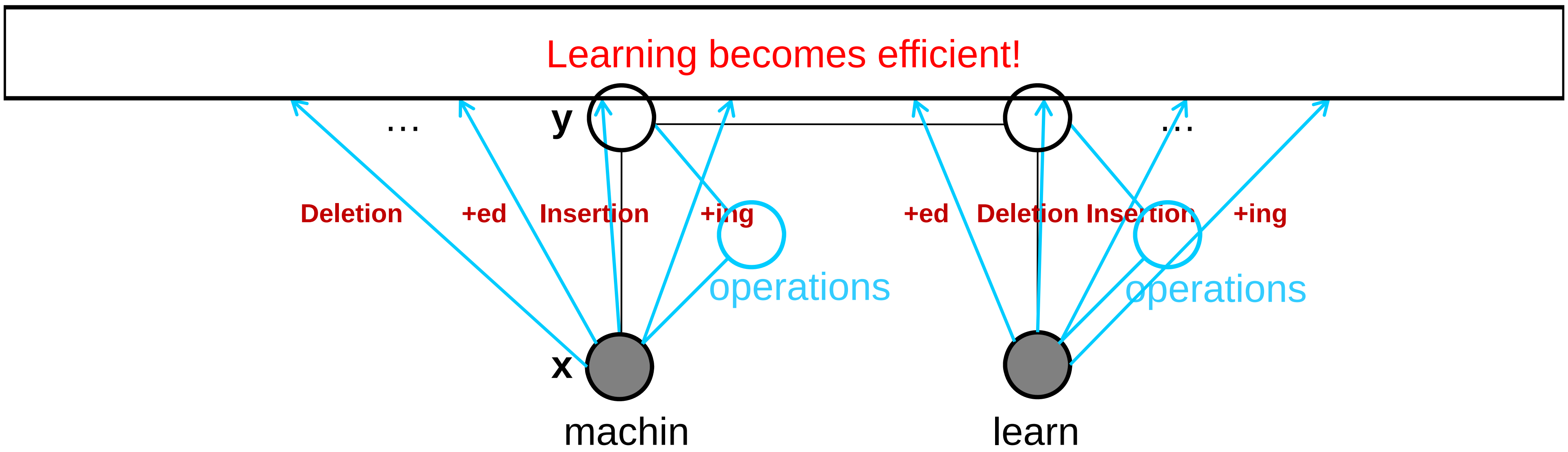
lean walk machined super soccer machining data

the learning paper mp3 book think macin clearn

machina lyrics learned machi new pc com lear

harry machine journal university net blearn course

… … … … … … …



1. o constrains the mapping from *x's to y's* (Reduce Space)
2. o indexes the mapping from *x's to y's* (Common Property)

Learning and Prediction

- Learning:
  - Labeled data (x, y, o)
  - Maximize the regularized log-likelihood function

$$\hat{\lambda} = \arg\max_{\lambda} \left\{ \sum_{i=1}^{N} \log(\Pr_{\lambda}(\boldsymbol{y}^{(i)}, \boldsymbol{o}^{(i)} | \boldsymbol{x}^{(i)})) - C\|\lambda\|_2 \right\}$$

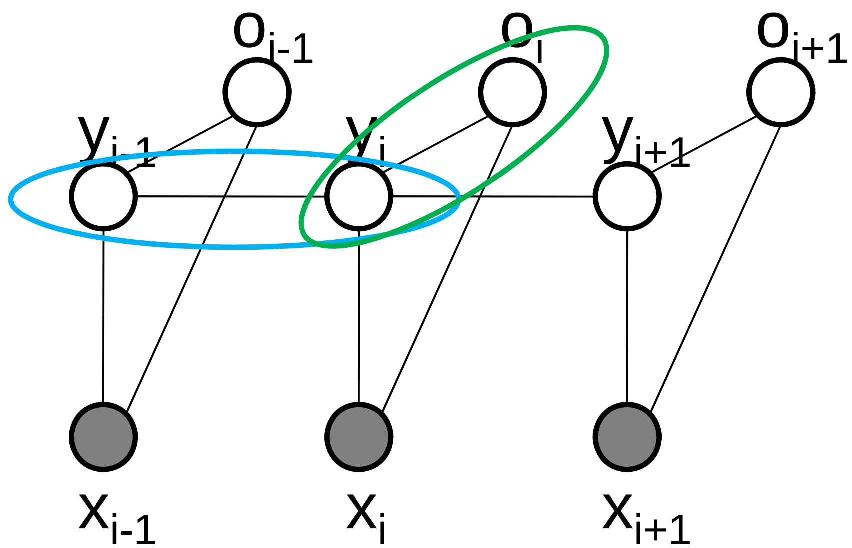
  - Quasi-Newton Method
  - Global optimal is guaranteed
- Prediction:
  - Viterbi algorithm

$$\boldsymbol{y}^* \boldsymbol{o}^* = \arg\max_{\boldsymbol{y}, \boldsymbol{o}} \Pr(\boldsymbol{y}, \boldsymbol{o} | \boldsymbol{x})$$

Features

Feature Type 1:

$$f(y_{i-1}, y_i) = \log \Pr(y_i | y_{i-1})$$



Feature Type 2:

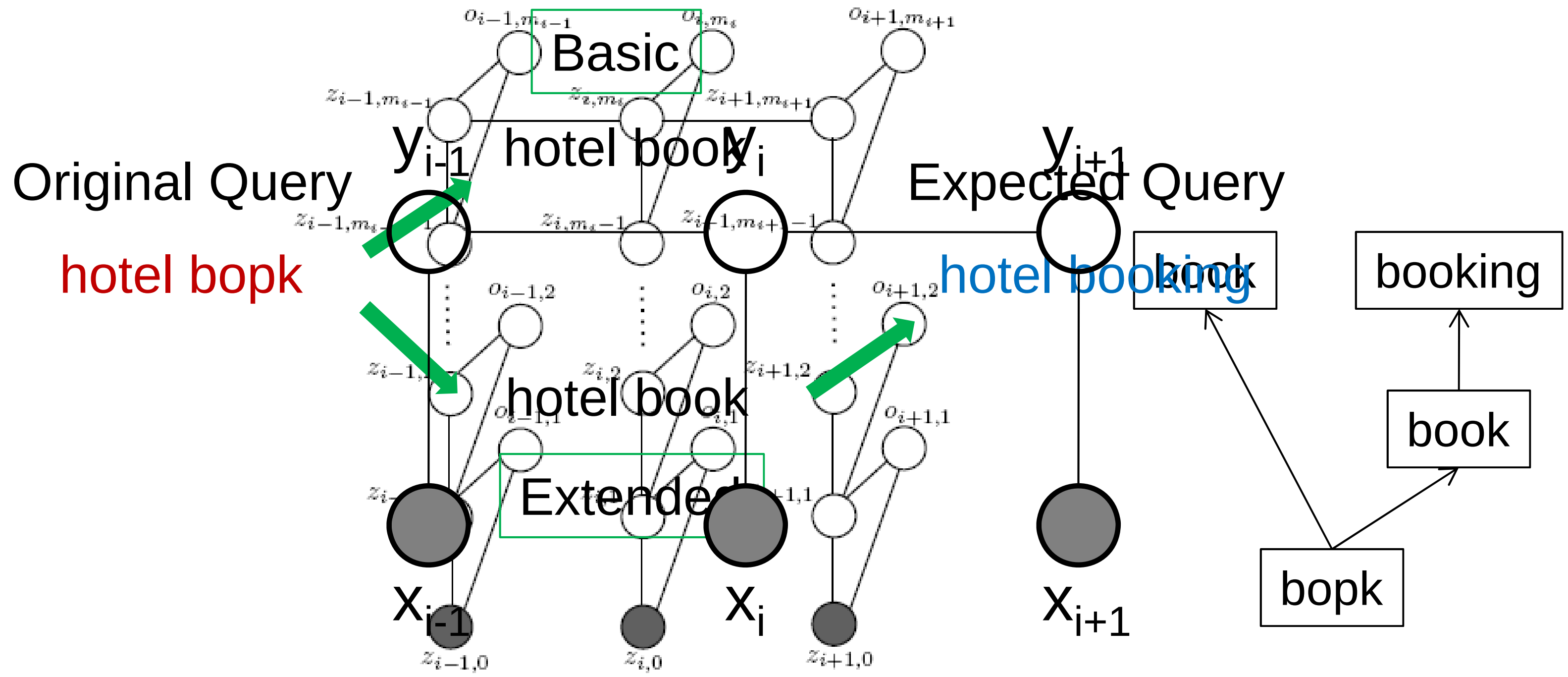
$$h(y_i, o_i, \boldsymbol{x}) = \begin{cases} 1, & \text{certain condition satisfied, given } o_i \\ 0, & \text{otherwise} \end{cases}$$

Lexicon-based feature Word-based feature Query-based feature

Position-based feature Corpus-based feature

CRF-QR Extended model

multiple refinement tasks needed



$$\Pr(\boldsymbol{y}, \vec{o}, \vec{z}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{i=1}^{n} (\phi(y_{i-1}, y_i) \prod_{j_i=1}^{m_i} \phi(z_{i,j_i}, o_{i,j_i}, z_{i,j_i-1}))$$

Outline

- Motivation
- Our Approach
- Experimental Results
- Conclusion

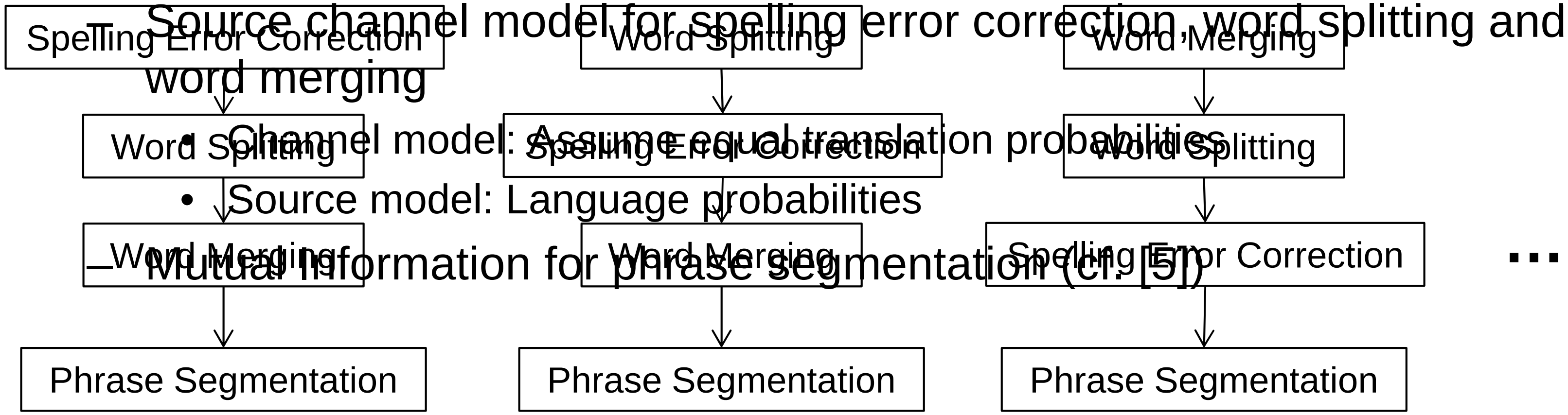
Experimental Result

- Data Set
  - Random select 10,000 queries
  - Average length: 2.8 words
  - Four human annotators
  - Four refinement types:
    - Spelling error correction
    - Word merging
    - Word splitting
    - Phrase segmentation
  - Training 7000 Testing 3000

Refinement Task	Num. of Refined Queries
Spelling Correction	733
Word Splitting	221
Word Merging	323
Phrase Segmentation	5,876

Baseline Method

- Cascaded approach
  - Build one sub-model for each task
  - Same structure and feature set for each sub-model
  - Sequentially connect the sub-models in different orders
- Generative approach
  - Source channel model for spelling error correction, word splitting and word merging
    - Channel model: Assume equal translation probabilities
    - Source model: Language probabilities
  - Mutual information for phrase segmentation (cf. [5])



Experiment on Query Refinement

Comparisons between CRF-QR and Baselines on Query Refinement at Query level (%)

	Pre.	Rec.	F1	Acc.
CRF-QR	**54.48**	**40.75**	**46.63**	**56.27**
Cascaded1	53.38	39.71	45.54	55.57
Cascaded2	53.38	39.71	45.54	55.57
Cascaded3	53.38	39.71	45.54	55.57
Cascaded4	53.45	39.76	45.60	55.60
Cascaded5	53.45	39.76	45.60	55.60
Cascaded6	53.45	39.76	45.60	55.60
Generative	30.46	32.95	31.66	39.10

Relative Improvement: F1 Score 2.26% Accuracy 1.21%

Cont’

Comparisons between CRF-QR and Baselines on Query Refinement Tasks (%)

	Spelling Correction			Word Splitting			Word Merging			Phrase Segmentation		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
CRF-QR	77.16	71.84	**74.40**	76.47	76.47	**76.47**	88.89	82.35	**85.50**	69.21	50.78	**58.58**
Cascaded1	73.91	68.39	71.04	74.19	67.65	70.77	86.67	76.47	81.25	69.00	50.13	58.07
Cascaded2	73.91	68.39	71.04	74.19	67.65	70.77	86.67	76.47	81.25	69.00	50.13	58.07
Cascaded3	74.68	67.43	70.87	70.59	70.59	70.59	86.67	76.47	81.25	69.01	50.16	58.09
Cascaded4	75.16	67.43	71.09	70.59	70.59	70.59	86.89	77.94	82.17	69.01	50.16	58.09
Cascaded5	74.38	68.39	71.26	74.19	67.65	70.77	86.89	77.94	82.17	69.00	50.13	58.07
Cascaded6	75.16	67.43	71.09	70.59	70.59	70.59	86.89	77.94	82.17	69.01	50.16	58.09
Generative	30.86	92.57	46.29	39.06	59.52	47.17	34.44	84.93	49.01	57.36	53.47	55.35

CRF-QR performs best!

Case Study

- Why CRF-QR can outperform the Baseline methods?
  - Cascaded approach suffers from the neglect of mutual dependencies between tasks
    - E.g. nypark hitel → ny "park hotel"
  - Cascaded approach accumulate errors
    - E.g. bankin las vegas → banking "las vegas" (bank in "las vegas")
  - Generative approach produces more incorrect results
    - E.g. pick up stix → pick up six door to door → "door to" door

Error Analysis

- (1) Errors were mainly made by one of the refinement tasks
  - E.g. parnell roberts → pernell roberts
  - Adding new features
  - Increasing data size for language model training
- (2) Competition between refinement tasks
  - E.g. skate board dudes → "skate board" dudes (skateboard dudes)
  - Adding new features
  - Increasing training data size
- (3) Some queries were difficult to refine even for humans
  - E.g. ohio buckeye card → "ohio buckeye" card (ohio "buckeye card")

Experiment on Relevance Search

Measure: NDCG

Results on Relevance Search with Entire Query Set (NDCG@3)

	Before	After
Human	0.265	0.304 (+14.7%)
CRF-QR	0.265	0.288 (+8.7%)

Results on Relevance Search with Refined Queries (NDCG@3)

	Refined	Before	After
Human	2023	0.254	0.312 (+22.8%)
CRF-QR	1546	0.258	0.304 (+17.7%)

Cont'

Results on Relevance Search by Query Refinement Tasks (NDCG@3)

		Refined	Before	After
Spelling Correction	Human	208	0.093	0.339
	Unified	163	0.078	0.322
Word Splitting	Human	61	0.190	0.333
	Unified	51	0.180	0.294
Word Merging	Human	120	0.198	0.305
	Unified	111	0.207	0.278
Phrase Segmentation	Human	1881	0.281	0.308
	Unified	1351	0.276	0.288

Outline

- Motivation
- Our Approach
- Experimental Results
- Conclusion

Conclusion

- Query Refinement
  - Automatically reformulate ill-formed queries
  - Better represent users' search needs
- CRF-QR model
  - Unified
  - Discriminative
- Experimental results
  - Query Refinement
  - Relevance Search

Thank You!