Accurate Linear-Time Chinese Word Segmentation via Embedding Matching

Jianqiang Ma & Erhard Hinrichs

Department of Linguistics & SFB833
University of Tübingen, Germany

Chinese Word Segmentation

- Input: 请在一米线外等候
- Output: 请 在 一米线 外 等候

EN: Please wait behind the one-meter-line

Erroneous: 请 在 一 米线 外 等候

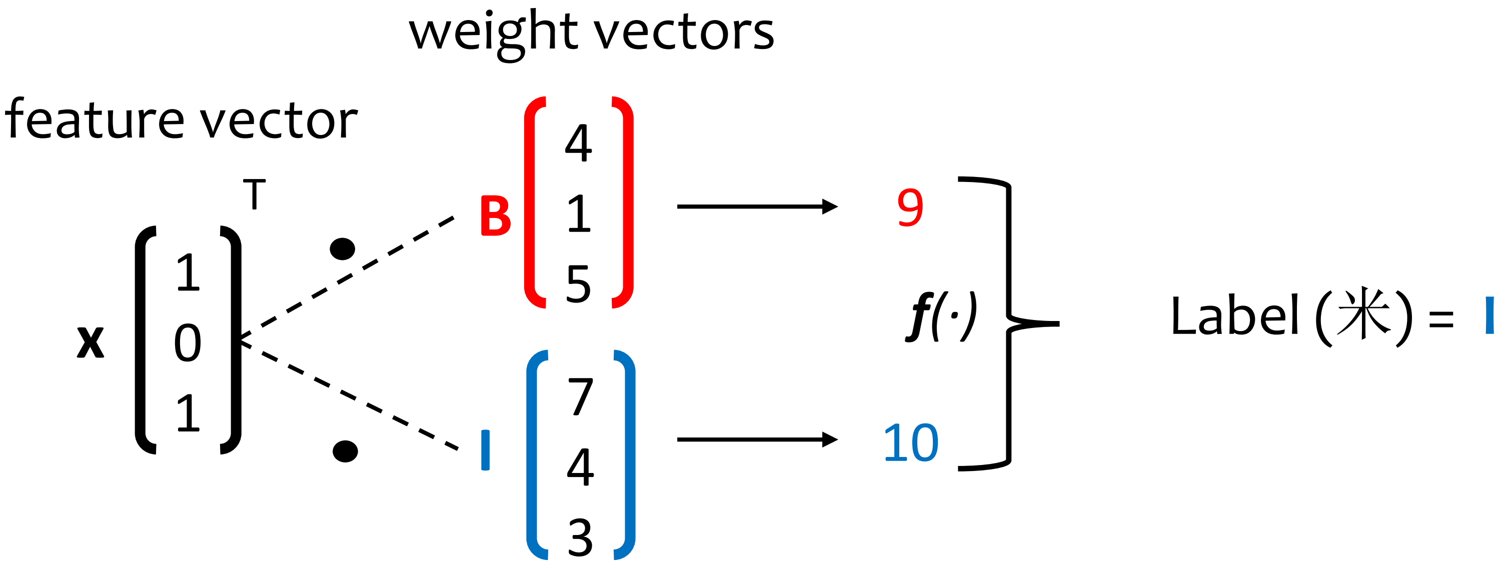
EN: Please wait outside rice-flour noodle.

Sequence Labeling for Word Segmentation

Input: 请在一米线外等候

Label: **B B B I I B B I** → Segmentation: 请 在 一米线 外 等候

B=begin, **I**= inside



One Size Fits All ?

One weight matrix
for *all* cases!

Linear model: $f(W^T \cdot x)$

One Size Fits All ?

One weight matrix
for *all* cases!

Linear model: $f(W^T \cdot x)$

Not Really!

Motivation Examples

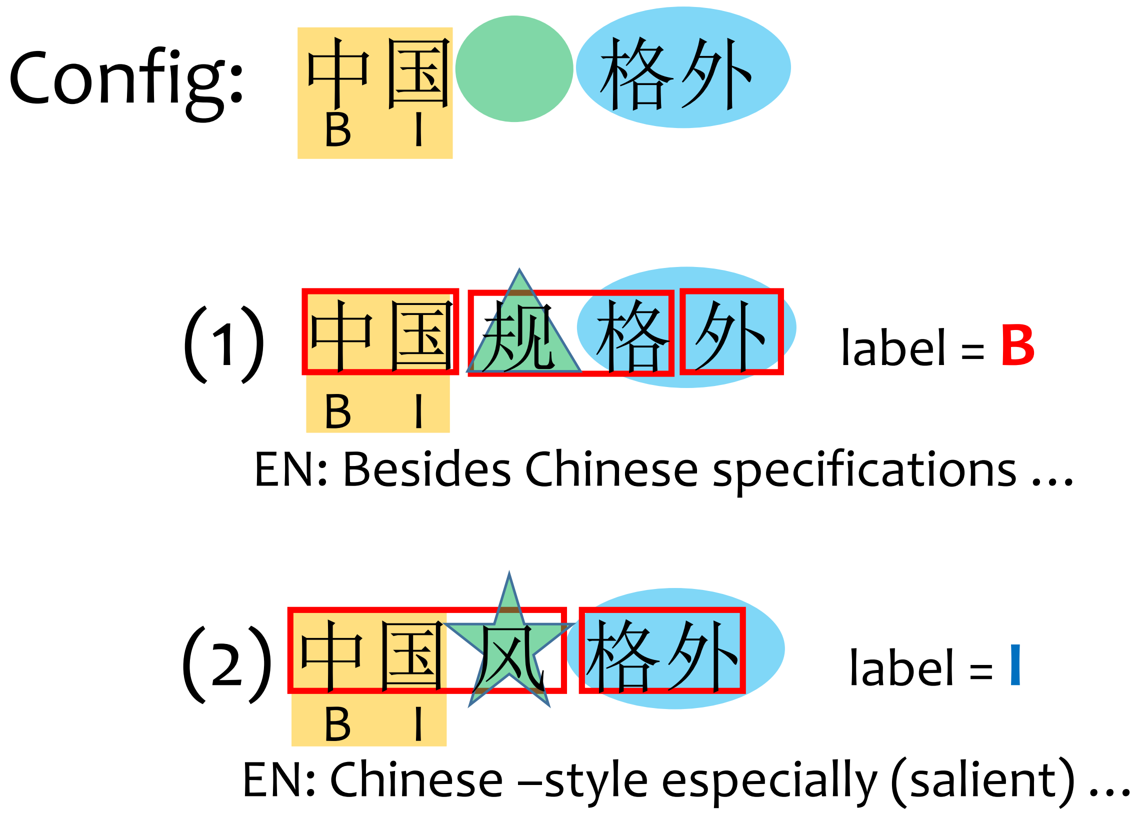
Configuration and Target Character

Config: 中国 格外
B I

(1) 中国规格外 label = **B**
B I
EN: Besides Chinese specifications …

(2) 中国风格外 label = **I**
B I
EN: Chinese –style especially (salient) …

Target Characters Impact Labels



Target character feature	Impact on Label	
	B	I
规 ‘rule’	+	-
风 ‘wind’	-	+

Conflicting Evidence: Same Target Character

Config: 中国 格外
B I

(2) 中国风格外 label = **I**
B I
EN: Chinese –style especially (salient) …

(3) 今天风很大 label = **B**
B I
EN: The wind is strong today

Target character Feature	Impact on Label	
	B	**I**
风 'wind'	–	+
风 'wind'	+	–

Conflicting Evidence: Same Configuration

Config: 中国 格外
B I

(1) 中国规格外 label = **B**
B I
EN: Besides Chinese specifications …

(2) 中国风格外 label = **I**
B I
EN: Chinese –style especially (salient) …

Configuration feature	Impact on Label	
	B	**I**
中国 B I	+	−
中国 B I	−	+

Character-Specific Tailoring of Labels

Config: 中国 格外
B I

Impact on *character-specific labels*:

规 'rule'-B label

Feature	规_**B**	规_**I**
中国 B I	+	–

(1) 中国规格外 label = B
B I
EN: Besides Chinese specifications ...

风 'wind'-B label

Feature	风_**B**	风_**I**
中国 B I	–	+

(2) 中国风格外 label = I
B I
EN: Chinese –style especially ...

Outline

- Embedding-based Matching
  - Matching & why embedding
  - Architecture, prediction & learning
- Experiments
  - Recent embedding-based models
  - State-of-the-art
- Conclusion

The Matching Formulation

Word Segmentation as Matching: Character-Specific Label

- Input: 请在一米线外等候

  'Please wait behind the one-*meter*-line'

- **Character-specific Labels:**

  请_B 在_B 一_B 米_I 线_I …

Word Segmentation as Matching: Features

- Configuration Features:

Unigram	在, 一, 米, 线, 外
Bigram	在一, 一米, 米线, 线外
Char-specific label	在_B, 一_B

Input sent: 请在一米线外等候

- Compare *Match* (*config*$_{米}$, 米_B) with *Match*(*config*$_{米}$, 米_I)

Why Embedding?

- Sequence labeling as matching?
  - |character| ~ 10^3 - 10^4, |labels| ~ 10^4
  - Annotated data ~ 10^6 character tokens

Data sparseness!

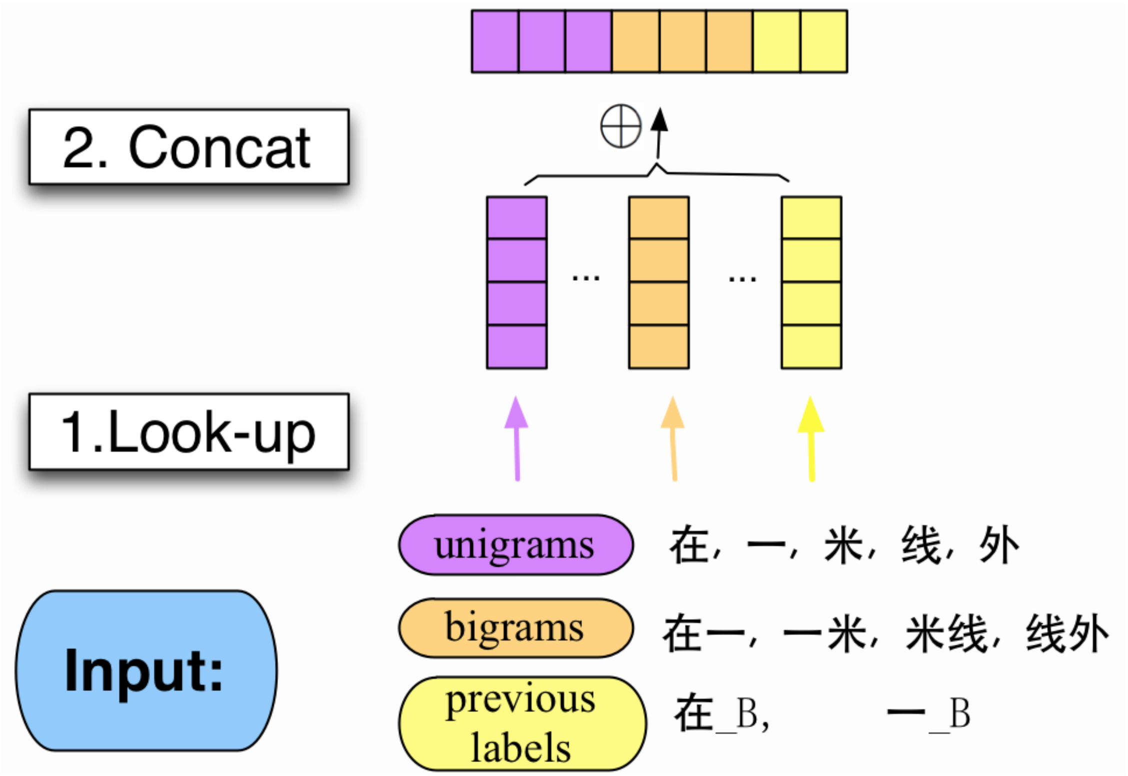
- Embeddings
  - Low-dimension vector representation of words and beyond
  - Similar items share similar vectors
  - Trained “by predict”

Why Embedding?

- In our model: embeddings are parameters *learned* from training data
- Jointly Learn embeddings for
  - Input features
  - Output character specific labels such as 米_B, 米_I

Embedding-Based Model

Embedding Matching Model: Input Embedding



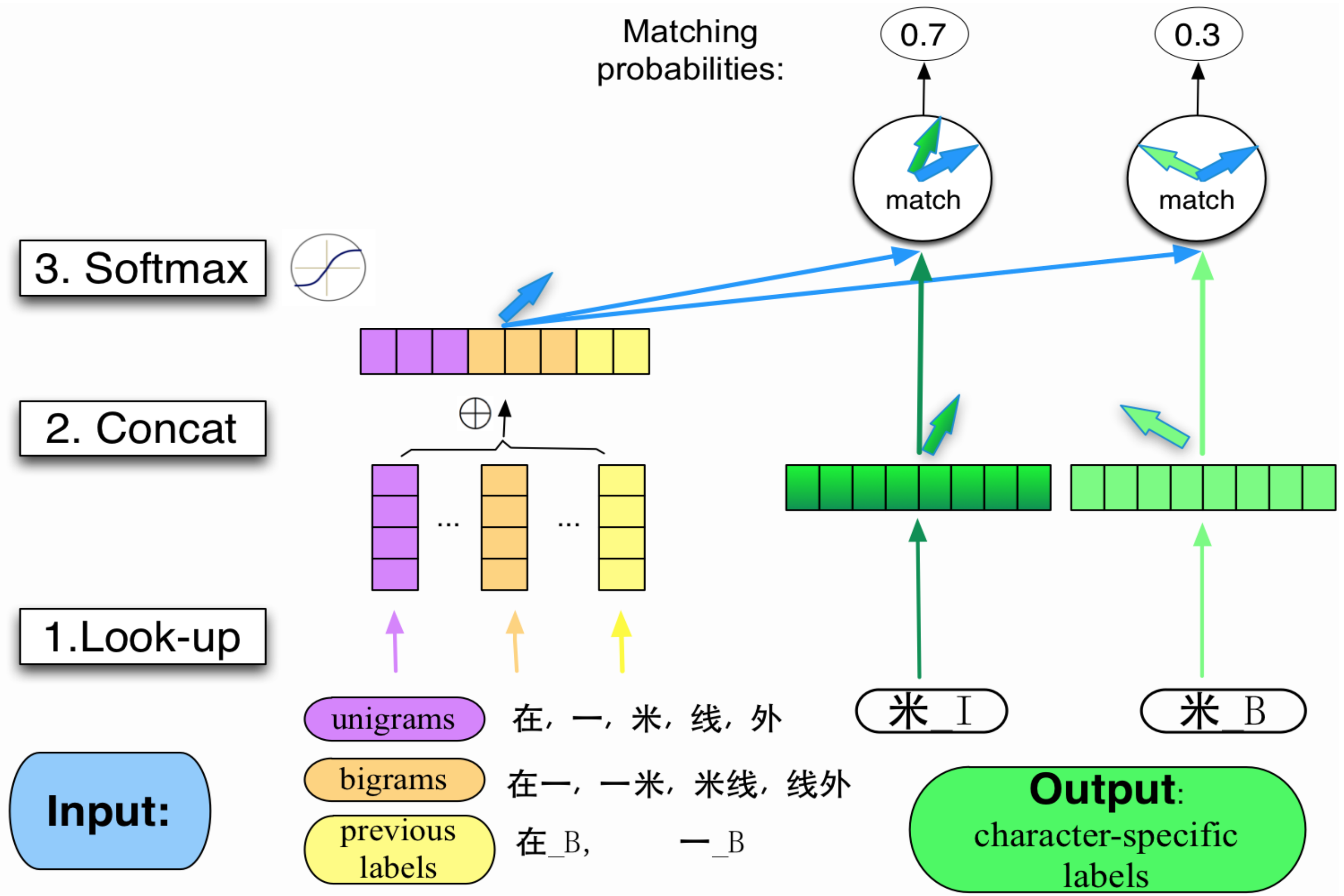
Input: 请在一米线外等候

'Please wait behind the one-meter-line'

Label: 请_B 在_B 一_B 米_I 线_I …

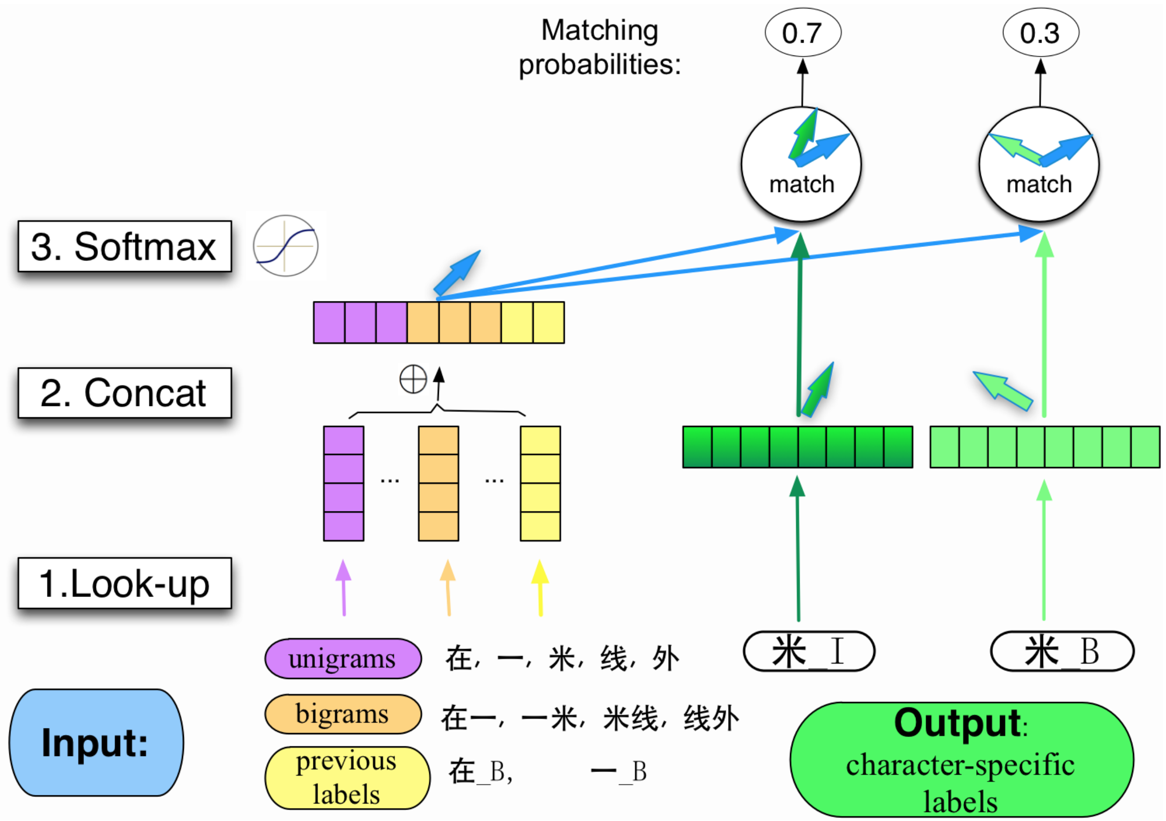
- Dim of each feature embedding: **50**
- Dim of concatenated embedding: **550**

Embedding Matching Model: Full Picture



Sentence: 请在一米线外等候 ‘Please wait behind the one-meter-line’

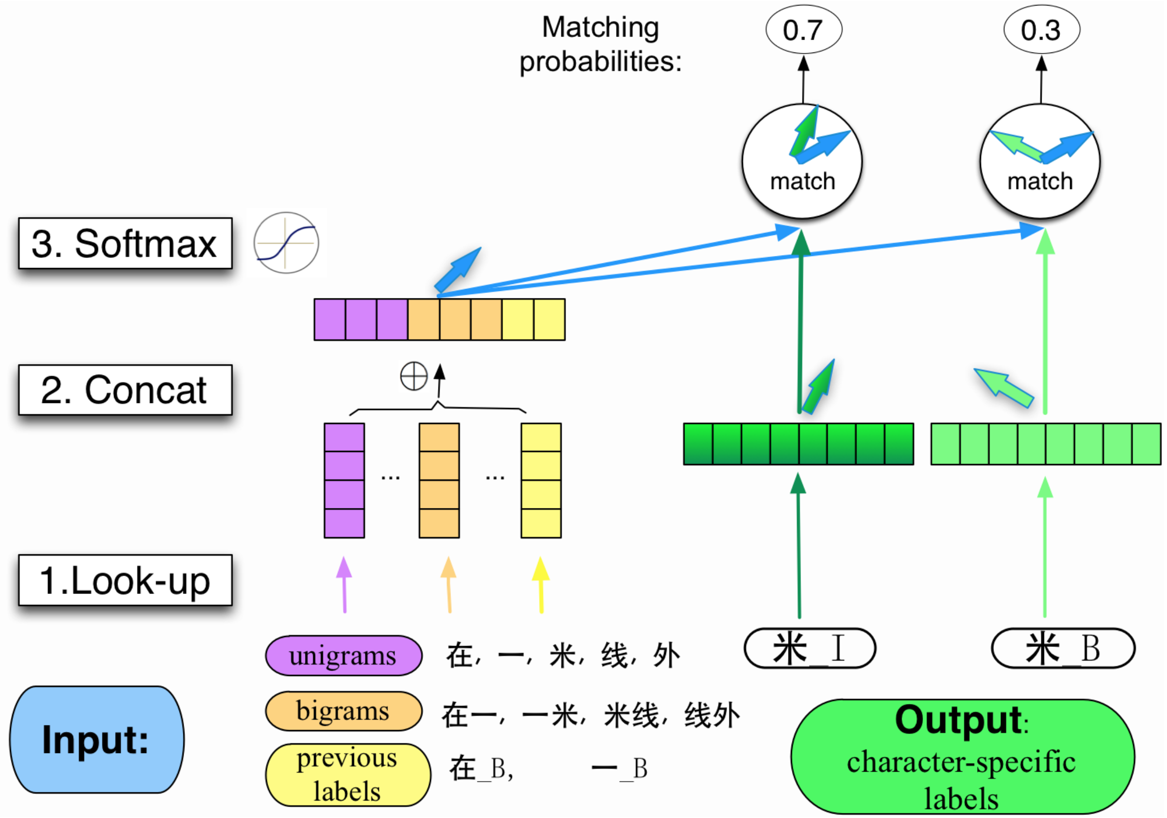
Model Characteristics



Sentence: 请在一米线外等候
'Please wait behind the one-meter-line'

- Input & output:
  - Embeddings to be learned
  - Learned jointly
  - Char-specific label: feature & output
- No hidden layer(s)
  - Embedding without complex NN
- Positive/negative cases

Sequence Prediction & Learning



- Greedy Search
- Learning
  - Objective: cross entropy + L2
  - Stochastic gradient descent
- Linear-time in search/learning

Experiments

Experiments: Data & Evaluation Metrics

- Data: **PKU** & **MSR** corpora
  - Bakeoff-2005
  - Standard split of training and test set

	PKU	**MSR**
Word types	5.5×10^4	8.8×10^4
Word tokens	1.1×10^6	2.4×10^6
Character types	5×10^3	5×10^3
Character tokens	1.8×10^6	4.1×10^6

- Metrics: precision(**P**), recall (**R**) & balanced f-score (**F**)

$$F = \frac{2 \times P \times R}{P + R}$$

  - Recall for out-of-vocabulary words ($\mathbf{R_{oov}}$)

Comparison with Previous Embedding Models

Models	PKU Corpus				MSR Corpus			
	P	**R**	**F**	**R_{oov}**	**P**	**R**	**F**	**R_{oov}**
Zheng et al.(2013)	92.8	92.0	92.4	63.3	92.9	93.6	93.3	55.7
+ *pre-training*†	93.5	92.2	92.8	69.0	94.2	93.7	93.9	64.1
Mansur et al. (2013)	93.6	92.8	93.2	57.9	92.3	92.2	92.2	53.7
+ *pre-training*†	94.0	93.9	94.0	69.5	93.1	93.1	93.1	59.7
Pei et al. (2014)	93.7	93.4	93.5	64.2	94.6	94.2	94.4	61.4
+ *pre-training*†	94.4	93.6	94.0	69.0	95.2	94.6	94.9	64.8
+ *pre-training & bigram*†	-	-	**95.2**	-	-	-	**97.2**	-
This work (closed-set)	**95.5**	**94.6**	95.1	**76.0**	**96.6**	**96.5**	96.6	**87.2**

Numbers in percentage. Results with † (dagger) used extra corpora for (pre-)training

Comparison with the State-of-the-Art

Mode	PKU	MSR	Method
Best05 closed-set	95.0	96.4	*Character-based CRF*
Zhang et al. (2006)	95.1	97.1	*CRF + dictionary matching*
Zhang & Clark (2007)	94.5	97.2	*Word-based structured perceptron*
Wang et al. (2012)	94.1	97.2	*Word-based LM + character CRF*
Sun et al. (2009)	95.2	97.3	*Latent variable model, character + word*
Sun et al. (2012)	95.4	**97.4**	*Adaptive training, new features*
Zhang et al. (2013)	**96.1**	**97.4**	*Semi-supervised co-training, new features*
This work (closed set)	95.1	96.6	*Embedding matching*

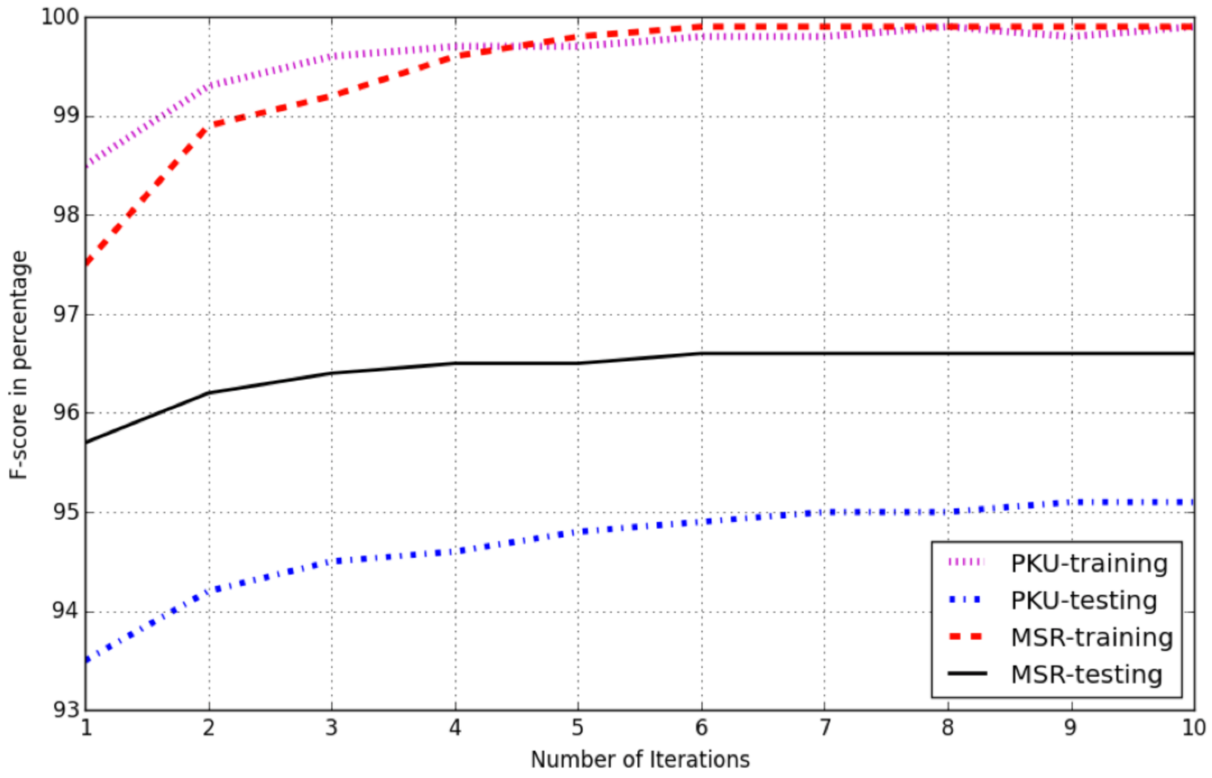
Numbers in percentage.

Comparison with the State-of-the-Art

Mode	PKU	MSR	Method
Best05 closed-set	95.0	96.4	*Character-based CRF*
Zhang et al. (2006)	95.1	97.1	*CRF + dictionary matching*
Zhang & Clark (2007)	94.5	97.2	*Word-based structured perceptron*
Wang et al. (2012)	94.1	97.2	*Word-based LM + character CRF*
Sun et al. (2009)	95.2	97.3	*Latent variable model, character + word*
Sun et al. (2012)	95.4	**97.4**	*Adaptive training, new features*
Zhang et al. (2013)	**96.1**	**97.4**	*Semi-supervised co-training, new features*
This work (closed set)	95.1	96.6	*Embedding matching*

Numbers in percentage.

Learning Curve and Hyper-Parameters



Size of feature embed'	$N_1 = 50$
Size of output embed'	$N_2 = 550$
Window size	$h = 5$
Initial learning rate	$\alpha = 0.1$
Regularization	$\lambda = 0.001$
Hybrid matching	$f_{top} = 8\%$

Conclusion & Future

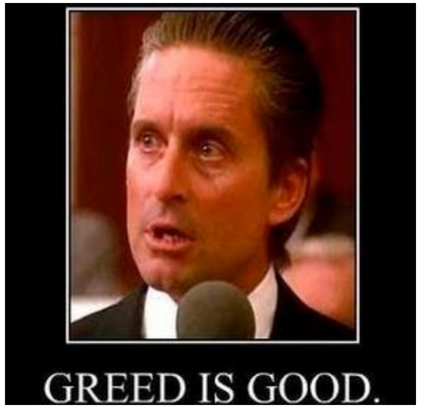
- Conclusion
  - Embedding matching model for Chinese word segmentation
  - First greedy segmenter comparable to the state-of-the-art
- Future Work
  - Better leverage external resources
  - Deep architecture
- Implementation publicly available: https://zenodo.org/record/17645

Take-Home Message

- Better matching than one-size-fit-all

- Embeddings empower fine-grained modeling such as matching

- Simplicity & greed(y search) is good



Thank you! 谢谢！

Questions?

Greedy & Exact Search-Based Models

Model	F-score/PKU	Training Time
Greey Search	0.975	1 X
Exact Search	0.944	7.8 X

- Each model is tailored to specific search errors
- Search is important only when model is inaccurate
- Zhang and Clark (2011): beam search > exact-search

Feature Impact

Feature	F-score	Feature	F-score
All features	**95.1**	uni-&bi-gram	94.6
w/o action	94.6	only action	93.3
w/o unigram	94.8	only unigram	92.1
w/o bigram	94.4	only bigram	94.2

Results on PKU corpus. Number in percentage.