

An Error-Driven Word-Character Hybrid Model for Joint Chinese Word Segmentation and POS Tagging

Canasai Kruengkrai^{†‡} Kiyotaka Uchimoto[‡] Jun'ichi Kazama[‡]
Yiou Wang[‡] Kentaro Torisawa[‡] Hitoshi Isahara^{†‡}

[†]Graduate School of Engineering, Kobe University

[‡]Language Infrastructure Group, MASTAR Project, NICT

ACL-IJCNLP 2009, Singapore

Outline

- 1 Introduction
- 2 Background
- 3 Policies for correct path selection
 - Motivation
 - Error-driven policy
- 4 Training method
 - Problem
 - Previous work
 - Our approach
- 5 Experiments
- 6 Conclusion

Chinese word segmentation and POS tagging

Input:

崇明是中国第三大島

word segmentation



崇明 是 中国 第三 大 島

POS tagging



Output:

崇明/noun 是/verb 中国/noun 第三/number 大/adjective 島/noun
(Chongming is China's third largest island)

Pipelined process

Chinese word segmentation and POS tagging

Input:

崇明是中国第三大島

word segmentation & POS tagging



Output:

崇明/**noun** 是/**verb** 中国/**noun** 第三/**number** 大/**adjective** 島/**noun**

(Chongming is China's third largest island)

Joint process

Chinese word segmentation and POS tagging

Input:

崇明是中国第三大島

word segmentation & POS tagging



Output:

崇明/**noun** 是/**verb** 中国/**noun** 第三/**number** 大/**adjective** 島/**noun**
(Chongming is China's third largest island)

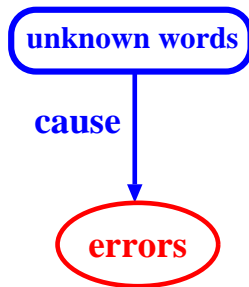
Joint process

Advantages

- Reduce **error propagation**
- Give improvements over the pipelined process

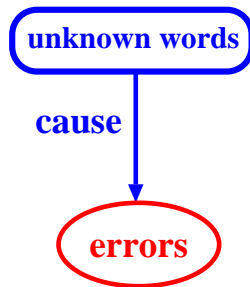
A serious problem: unknown words

- Unknown words = words that are not found in a training corpus or in a system's word dictionary



A serious problem: unknown words

- Unknown words = words that are not found in a training corpus or in a system's word dictionary
- To achieve the optimal performance, the system must **effectively handle unknown words.**



Previous work

Character-based approach

Xue [2003], Ng and Low [2004], Jiang *et al.* [2008]

Word-based (dictionary-based) approach

Uchimoto *et al.* [2001], Kudo *et al.* [2004]

Word-character hybrid approach

Nakagawa and Uchimoto [2007]

Previous work

Character-based approach

- Pros: simple, robust for processing unknown words
- Cons: lose useful information on words

Word-based (dictionary-based) approach

Uchimoto *et al.* [2001], Kudo *et al.* [2004]

Word-character hybrid approach

Nakagawa and Uchimoto [2007]

Previous work

Character-based approach

- Pros: simple, robust for processing unknown words
- Cons: lose useful information on words

Word-based (dictionary-based) approach

- Pros: precise for processing known words
- Cons: need a good dictionary, difficult to handle unknown words

Word-character hybrid approach

Nakagawa and Uchimoto [2007]

Previous work

Character-based approach

- Pros: simple, robust for processing unknown words
- Cons: lose useful information on words

Word-based (dictionary-based) approach

- Pros: precise for processing known words
- Cons: need a good dictionary, difficult to handle unknown words

Word-character hybrid approach

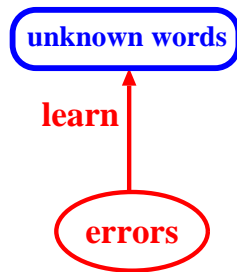
- Pros: good for processing both known and unknown words
- Cons: difficult to train, need good balance for learning known and unknown word statistics

Our solution

- Focus on the word-character hybrid approach

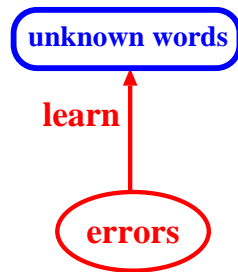
Our solution

- Focus on the word-character hybrid approach
- Propose a simple, unified way to train the model based on discriminative online learning
- Learn **unknown word characteristics** from errors



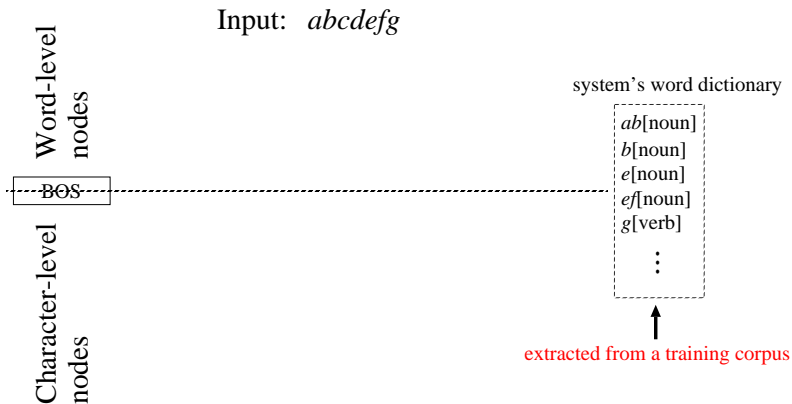
Our solution

- Focus on the word-character hybrid approach
- Propose a simple, unified way to train the model based on discriminative online learning
- Learn **unknown word characteristics** from errors
- Achieve superior performance compared with the best existing methods



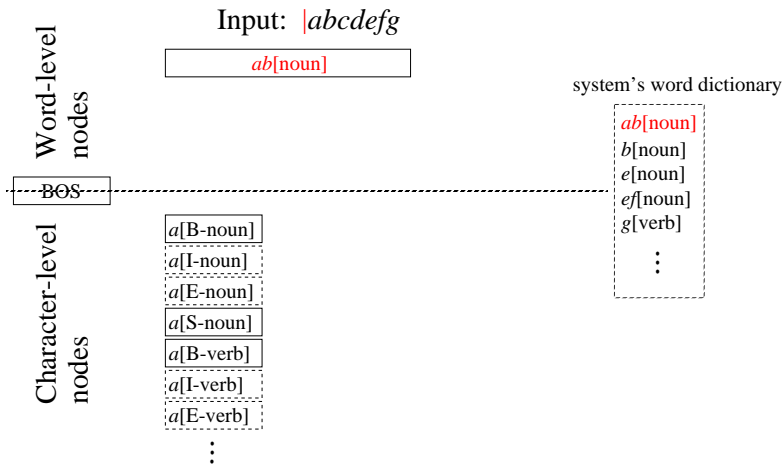
The word-character hybrid model

- **Key idea:** represent an input sentence with a lattice consisting of **word-level** and **character-level** nodes



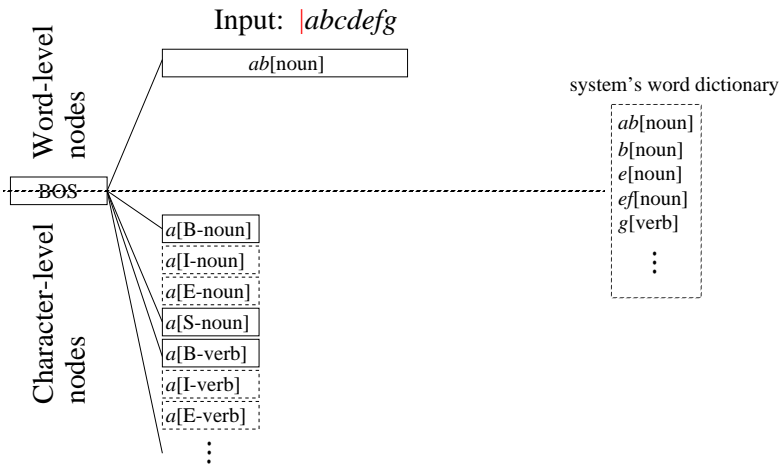
The word-character hybrid model

- **Key idea:** represent an input sentence with a lattice consisting of **word-level** and **character-level** nodes



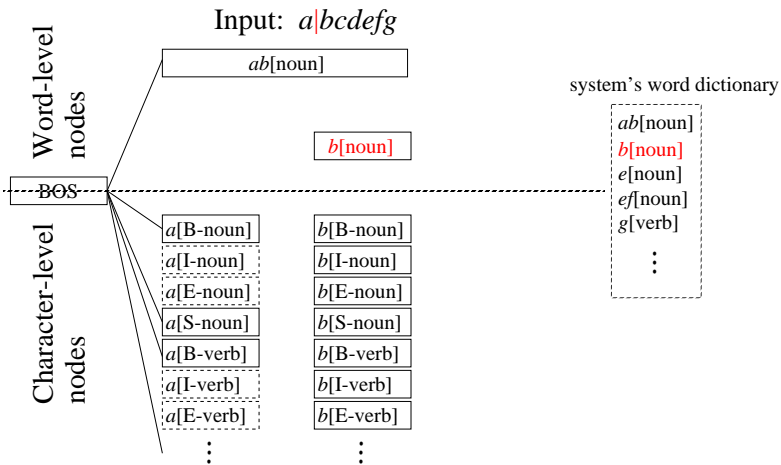
The word-character hybrid model

- **Key idea:** represent an input sentence with a lattice consisting of **word-level** and **character-level** nodes



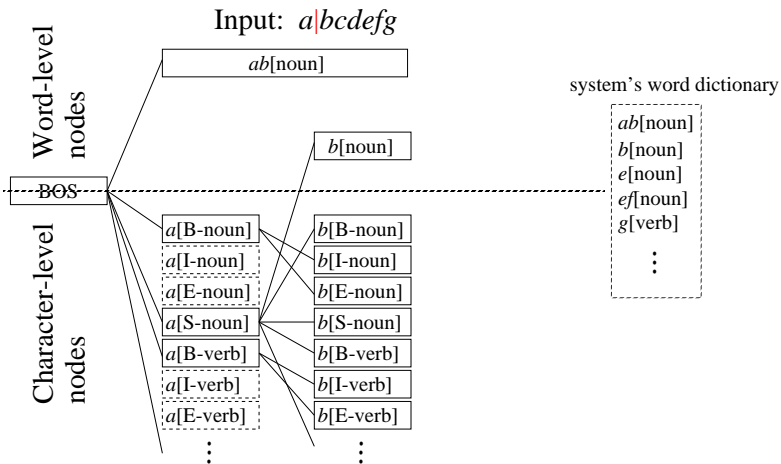
The word-character hybrid model

- **Key idea:** represent an input sentence with a lattice consisting of **word-level** and **character-level** nodes



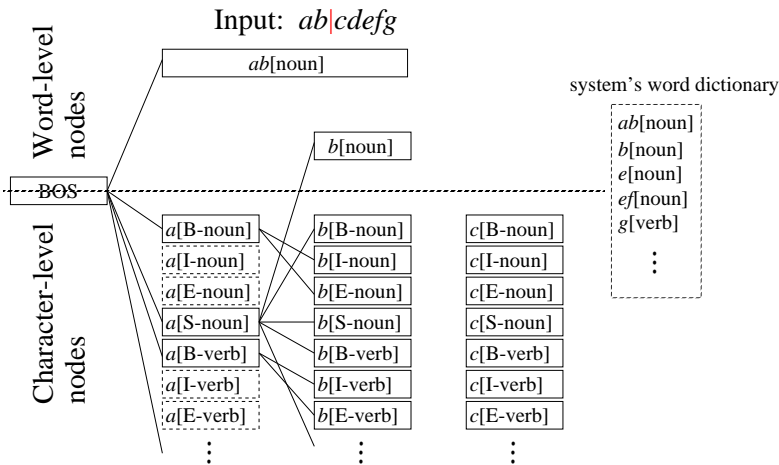
The word-character hybrid model

- **Key idea:** represent an input sentence with a lattice consisting of **word-level** and **character-level** nodes



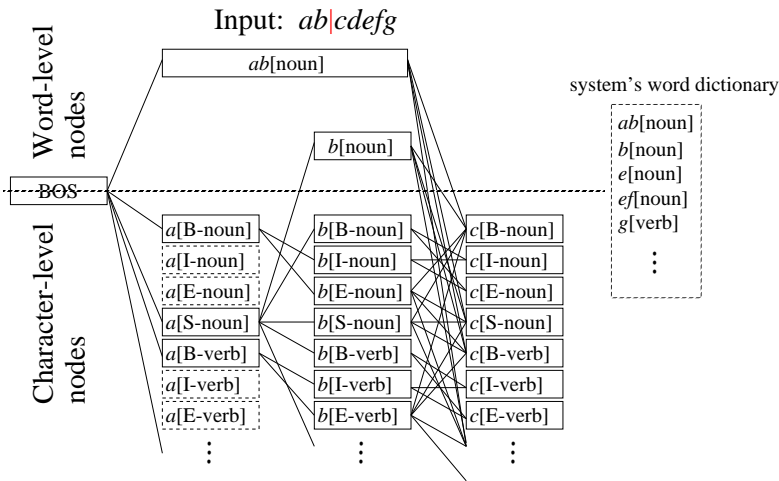
The word-character hybrid model

- **Key idea:** represent an input sentence with a lattice consisting of **word-level** and **character-level** nodes



The word-character hybrid model

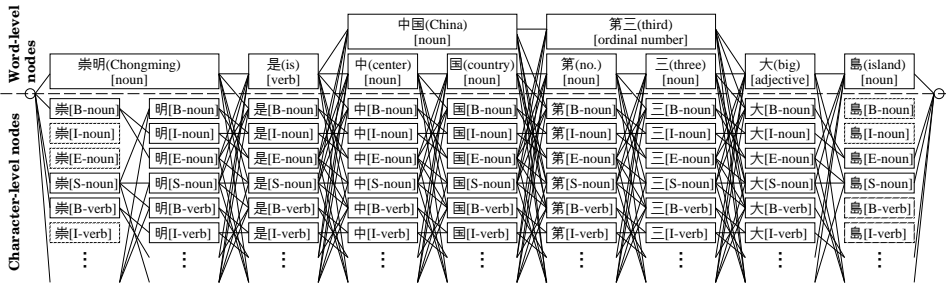
- **Key idea:** represent an input sentence with a lattice consisting of **word-level** and **character-level** nodes



The word-character hybrid model

Sentence:

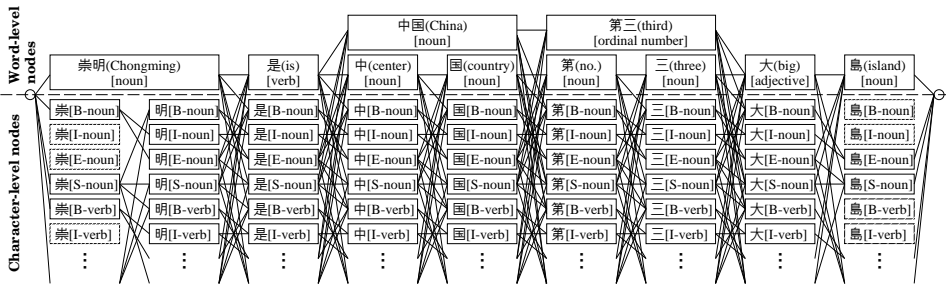
崇明是中国第三大島



How can we select a correct path for training?

Annotated sentence:

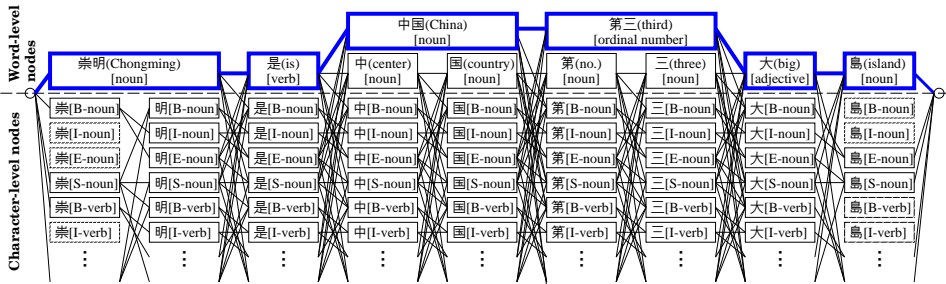
崇明/[noun] 是/[verb] 中国/[noun] 第三/[ordinal number] 大/[adjective] 島/[noun]



How can we select a correct path for training?

Annotated sentence:

崇明/[noun] 是/[verb] 中国/[noun] 第三/[ordinal number] 大/[adjective] 島/[noun]

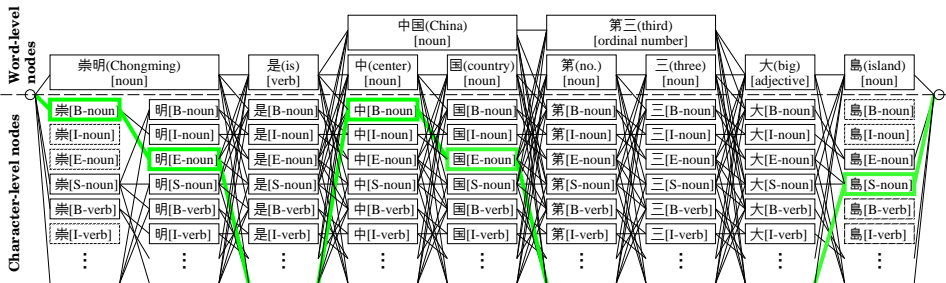


In training, if we select the **correct path** corresponding to the annotated sentence, it will only consist of **word-level nodes** that do not allow learning for unknown words.

How can we select a correct path for training?

Annotated sentence:

崇明/[noun] 是/[verb] 中国/[noun] 第三/[ordinal number] 大/[adjective] 島/[noun]

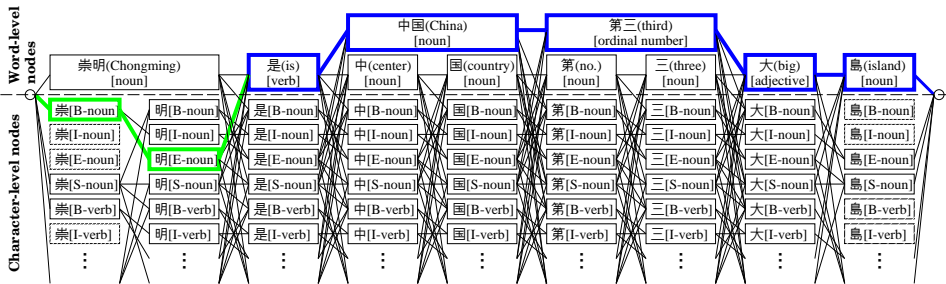


However, if we use **character-level nodes** to represent the annotated sentence, we lose useful features on word surfaces.

How can we select a correct path for training?

Annotated sentence:

崇明/[noun] 是/[verb] 中国/[noun] 第三/[ordinal number] 大/[adjective] 島/[noun]

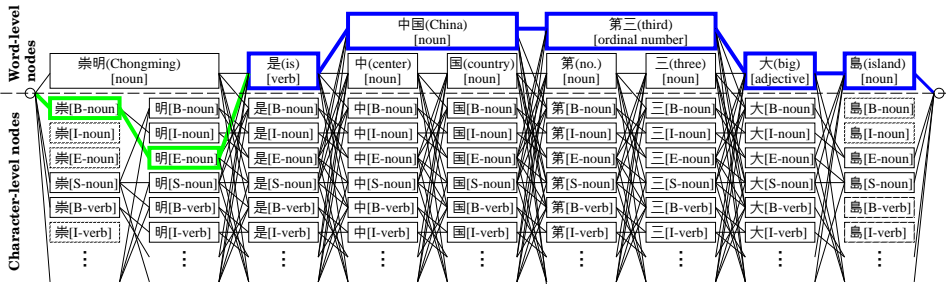


We need to select some **word-level nodes** for learning statistics of **known words** and some **character-level nodes** for learning those of **unknown words**.

How can we select a correct path for training?

Annotated sentence:

崇明/[noun] 是/[verb] 中国/[noun] 第三/[ordinal number] 大/[adjective] 島/[noun]



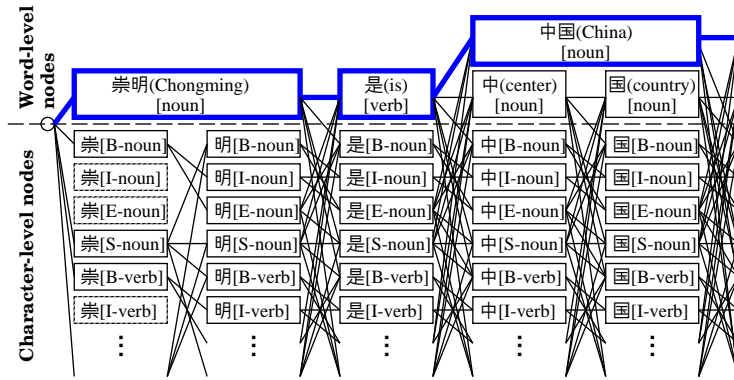
We need to select some **word-level nodes** for learning statistics of **known words** and some **character-level nodes** for learning those of **unknown words**. **What is an optimal way?**

Policies for Correct Path Selection

Policies for correct path selection

Annotated sentence:

崇明/[noun] 是/[verb] 中国/[noun] 第三/[ordinal number] 大/[adjective] 島/[noun]



Start by using all **word-level nodes**

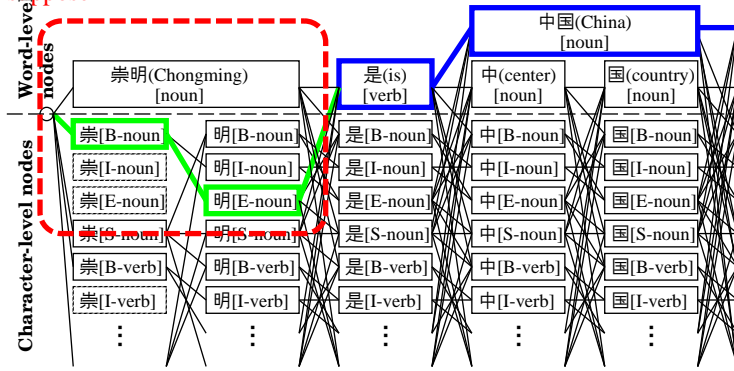
Policies for correct path selection: baseline

Annotated sentence:

崇明/[noun] 是/[verb] 中国/[noun] 第三/[ordinal number] 大/[adjective] 島/[noun]

occurs only once in the training corpus

suppose



Use character-level nodes instead of word-level nodes for rare words
← our baseline policy

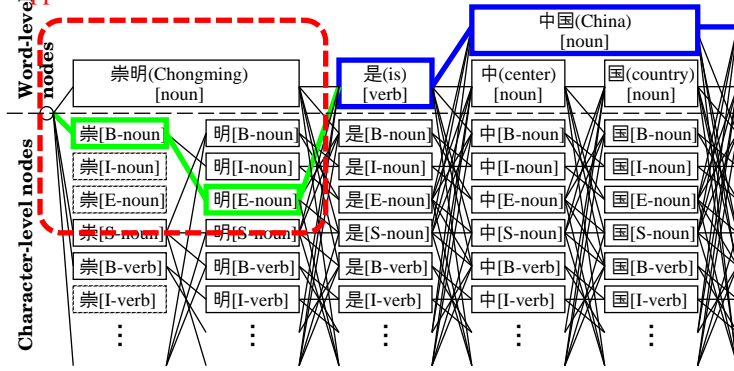
Policies for correct path selection: baseline

Annotated sentence:

崇明/[noun] 是/[verb] 中国/[noun] 第三/[ordinal number] 大/[adjective] 島/[noun]

occurs only once in the training corpus

suppose



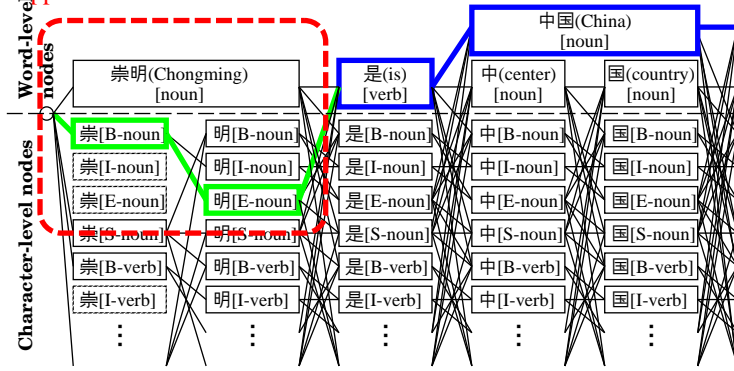
Our baseline policy derives from Baayen and Sproat's assumption [1996]: the characteristics of **rare words** in a training corpus resemble those of **unknown words**.

Policies for correct path selection: error-driven

Annotated sentence:

崇明/[noun] 是/[verb] 中国/[noun] 第三/[ordinal number] 大/[adjective] 島/[noun]

崇明 is an erroneous word
suppose

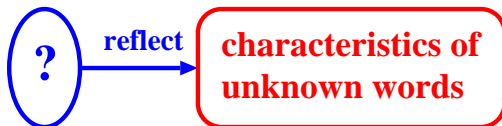


Use character-level nodes instead of word-level nodes for particular erroneous words in a training corpus ← **our error-driven policy**

Error-driven policy

Motivation

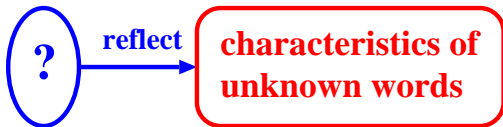
- We need to find words that reflect the **characteristics of unknown words.**



Error-driven policy

Motivation

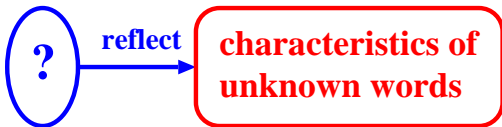
- We need to find words that reflect the **characteristics of unknown words.**
- We found that using a set of rare words is not enough because many errors still remain, **mainly caused by unknown words.**



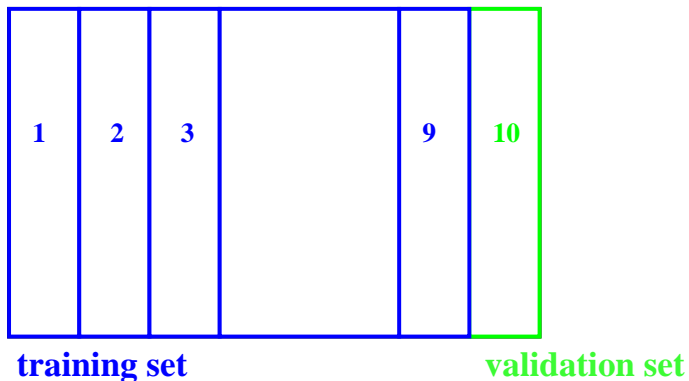
Error-driven policy

Motivation

- We need to find words that reflect the **characteristics of unknown words**.
- We found that using a set of rare words is not enough because many errors still remain, **mainly caused by unknown words**.
- So, we need to find an **additional set of words** that may capture the characteristics of unknown words.

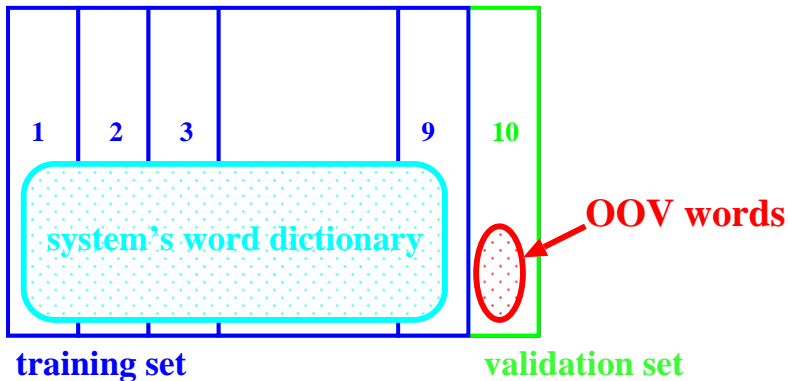


Error-driven policy



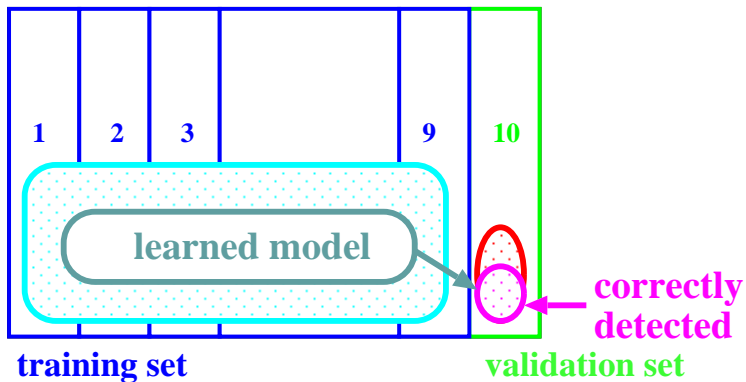
Let's start by performing **10-fold cross validation** on the training corpus

Error-driven policy



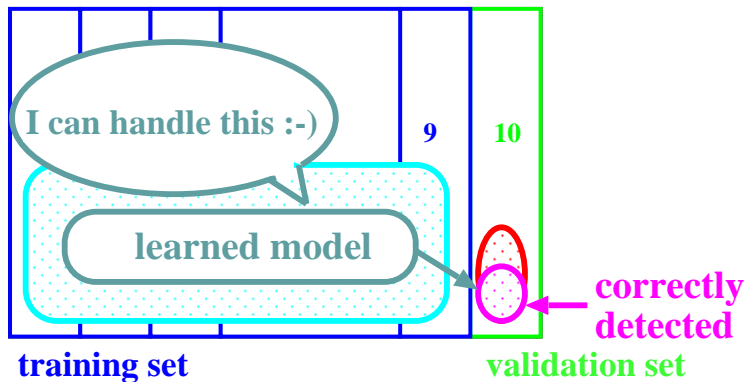
A set of out-of-vocabulary (**OOV**) words in each validation set is a possible candidate that may reflect the **characteristics of unknown words**

Error-driven policy



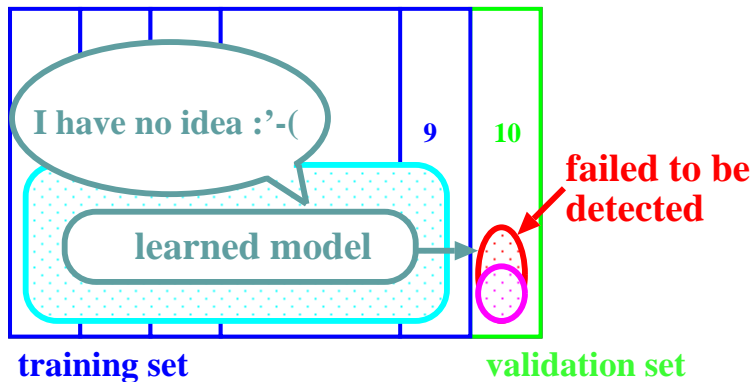
However, some of OOV words are **correctly detected**

Error-driven policy



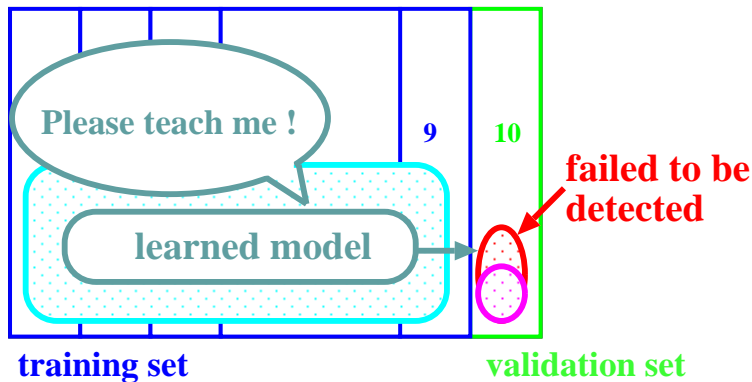
However, some of OOV words are **correctly detected** \Leftarrow the system already captures their characteristics

Error-driven policy



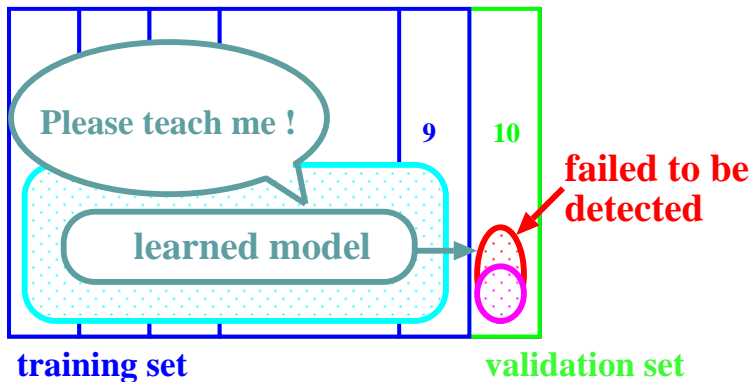
Some of OOV words cannot be correctly recognized

Error-driven policy



Let the system learn the characteristics of unknown words from the OOV words failed to be detected

Error-driven policy



Let the system learn the characteristics of unknown words from the
OOV words failed to be detected \Leftarrow **unidentified unknown words**

Error-driven policy: an example

Test sentence:

ABCDEMNX~~Y~~

An unsegmented test sentence in the validation set

Error-driven policy: an example

Correct annotation:

AB/[noun] C/[verb] DE/[noun] MN/[noun] X/[adjective] Y/[noun]

We know its correct annotation

Error-driven policy: an example

Correct annotation:

AB/[noun] C/[verb] DE/[noun] MN/[noun] X/[adjective] Y/[noun]

in-vocabulary (IV) words

Three words can be found in the system's word dictionary

Error-driven policy: an example

Correct annotation:

AB/[noun] C/[verb] DE/[noun] MN/[noun] X/[adjective] Y/[noun]

out-of-vocabulary (OOV) words

Three words cannot be found in the system's word dictionary

Error-driven policy: an example

Correct annotation:

AB/[noun] C/[verb] DE/[noun] MN/[noun] X/[adjective] Y/[noun]
OOV OOV OOV

System output:

A/[noun] B/[noun] C/[verb] DE/[noun] MN/[verb] X/[adjective] Y/[noun]

Compare with the predicted sentence by the system

Error-driven policy: an example

Correct annotation:

AB/[noun] C/[verb] DE/[noun] MN/[noun] X/[adjective] Y/[noun]
 OOV OOV OOV

System output:

A/[noun] B/[noun] C/[verb] DE/[noun] MN/[verb] X/[adjective] Y/[noun]

The system can correctly recognize four words, including one OOV word

Error-driven policy: an example

unidentified

Correct annotation:

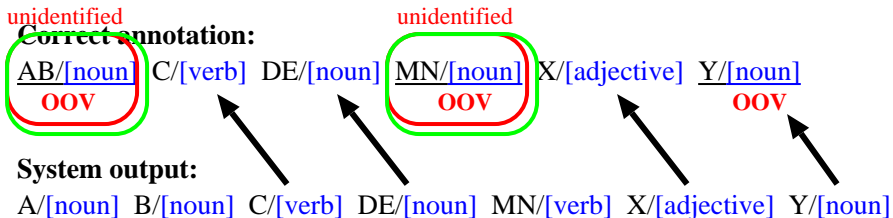
AB/[noun] C/[verb] DE/[noun] MN/[noun] X/[adjective] Y/[noun]
 OOV OOV

System output:

A/[noun] B/[noun] C/[verb] DE/[noun] MN/[verb] X/[adjective] Y/[noun]

The system cannot recognize two OOV words

Error-driven policy: an example



We use them for estimating **unknown word statistics** in the training step

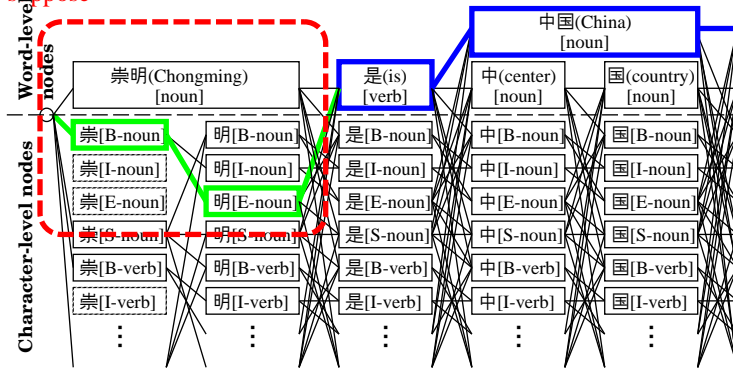
Policies for correct path selection: error-driven

Annotated sentence:

崇明/[noun] 是/[verb] 中国/[noun] 第三/[ordinal number] 大/[adjective] 島/[noun]

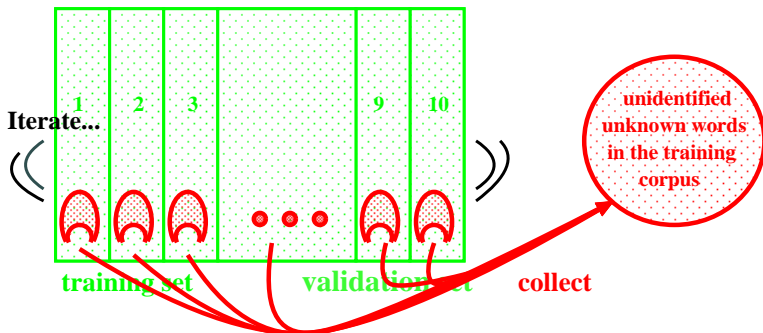
is an unidentified unknown word

suppose



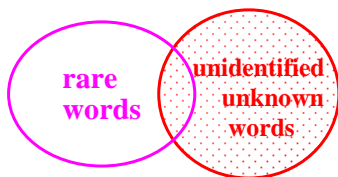
Use character-level nodes instead of word-level nodes for **unidentified unknown words** in the training corpus

Error-driven policy



After ten cross validation runs, we get a set of the unidentified unknown words derived from the **whole training corpus**

Error-driven policy

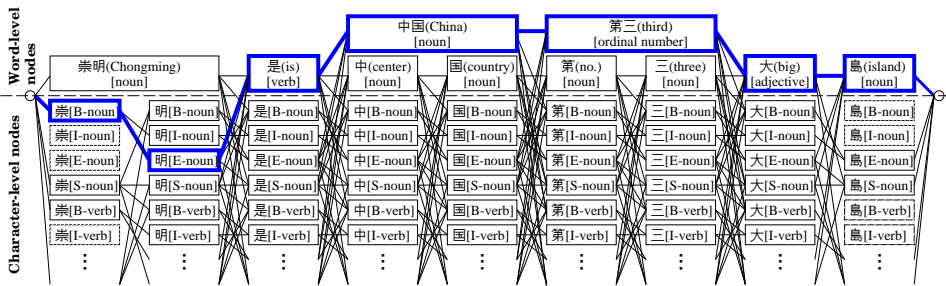


Artificial unknown words for learning

Combine **unidentified unknown words** in cross validation and **rare words** to obtain the optimal set of artificial unknown words

Problem

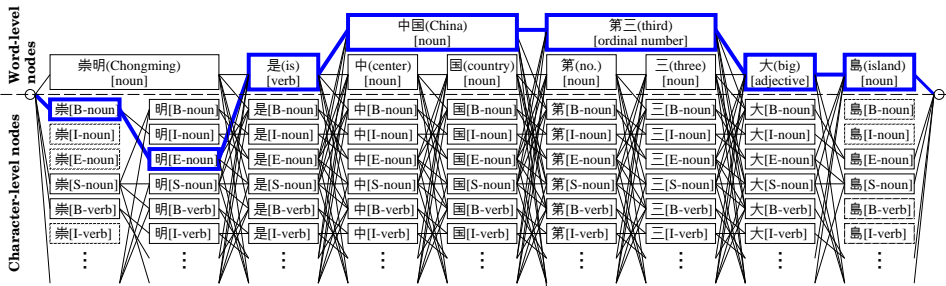
Training method



How can we train the model with large and complex lattice structures?

Problem

Training method



In practice, a lattice can contain more than 1000 nodes and 10000 links, depending on the length of the sentence.

Training method

- Nakagawa [COLING 2004] proposed a two-step training method based on the **word-base** Markov model and the **character-based** maximum entropy model.

Training method

- Nakagawa [COLING 2004] proposed a two-step training method based on the **word-base** Markov model and the **character-based** maximum entropy model.

Disadvantages

- Model parameters are separately estimated
- Hard to exploit useful features on words with the Markov model

Training method

- Nakagawa [COLING 2004] proposed a two-step training method based on the **word-base** Markov model and the **character-based** maximum entropy model.

Disadvantages

- Model parameters are separately estimated
 - Hard to exploit useful features on words with the Markov model
-
- We propose a simple, unified training method based on discriminative online learning.

Training method

- We apply an online learning algorithm called Margin Infused Relaxed Algorithm (**MIRA**) [Cammer, 2004].

Training method

- We apply an online learning algorithm called Margin Infused Relaxed Algorithm (**MIRA**) [Cammer, 2004].
- Online learning processes one training example at a time and updates model parameters to improve accuracy on that example. [▶ click](#)

Training method

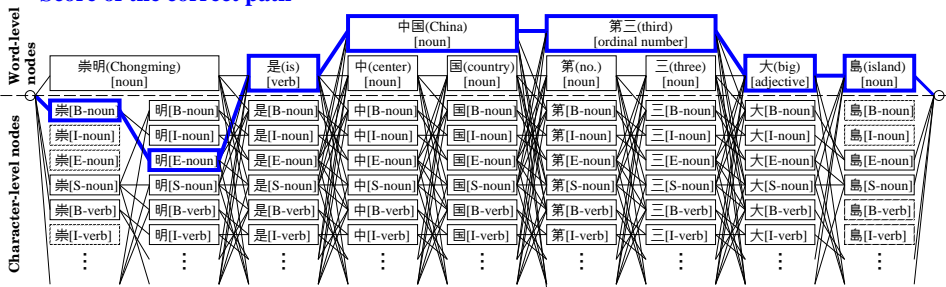
- We apply an online learning algorithm called Margin Infused Relaxed Algorithm (**MIRA**) [Cammer, 2004].
- Online learning processes one training example at a time and updates model parameters to improve accuracy on that example. [▶ click](#)

Advantages

- Quickly converge within a few iterations (e.g., 10)
- Enable us to incorporate **arbitrary features** to the model

Training method

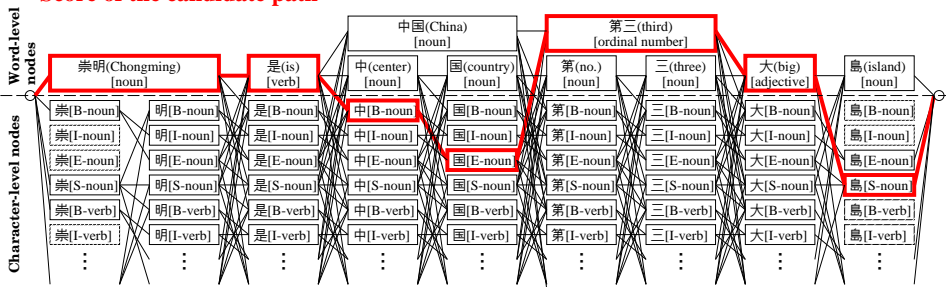
Score of the correct path



Calculate the score of the correct path $s(\mathbf{x}_t, \mathbf{y}_t; \mathbf{w})$

Training method

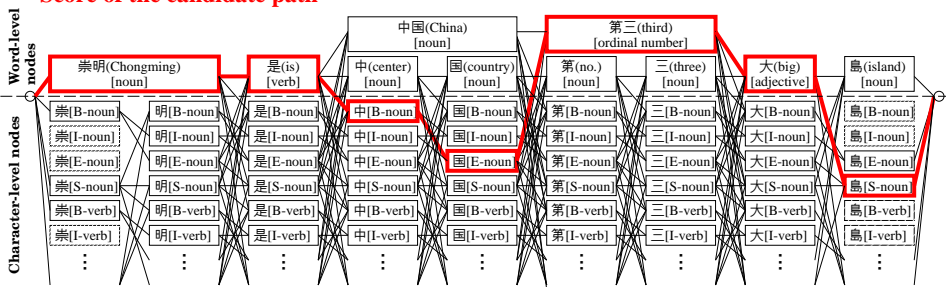
Score of the candidate path



Calculate the score of the best candidate path $s(x_t, \hat{y}; \mathbf{w})$

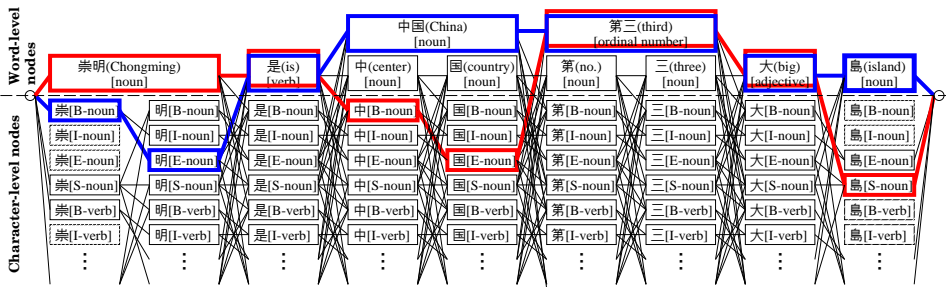
Training method

Score of the candidate path



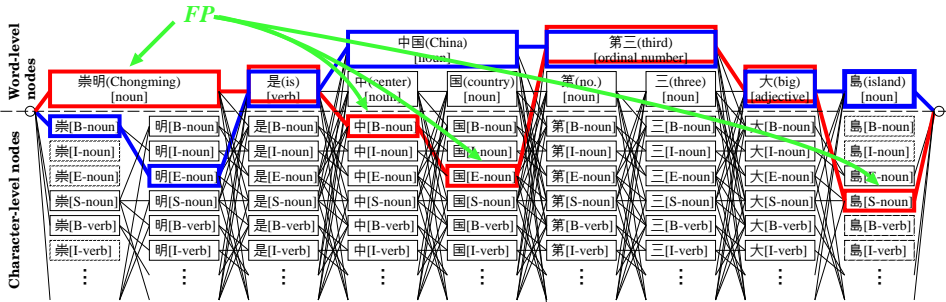
Can apply **Viterbi-style** algorithm and extend to k -best paths with **A*-style** algorithm

Training method



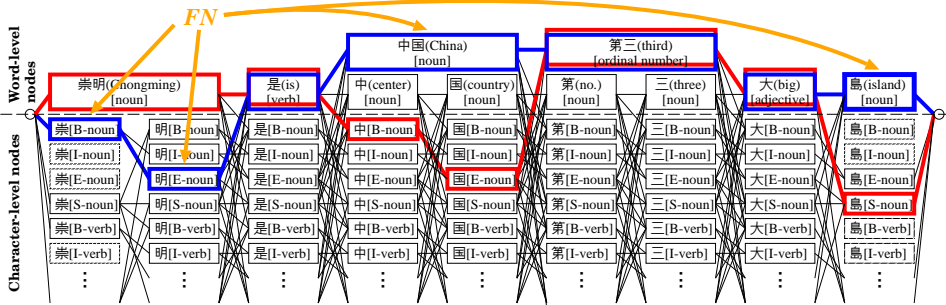
Calculate the loss function $L(\mathbf{y}_t, \hat{\mathbf{y}})$

Training method



False Positive (**FP**) = number of output nodes that are not in the correct path

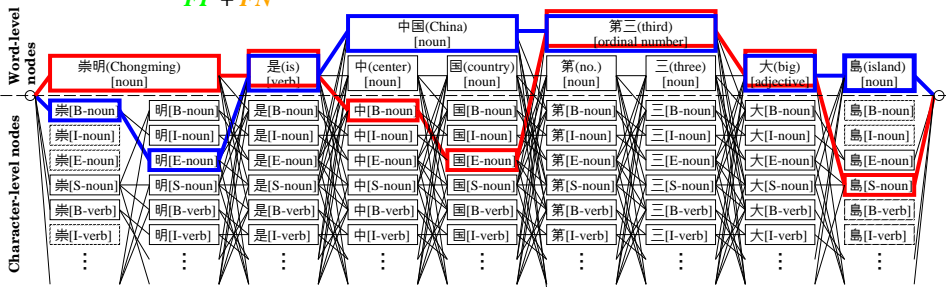
Training method



False Negative (*FN*) = number of nodes in the correct path that were not recognized by the system

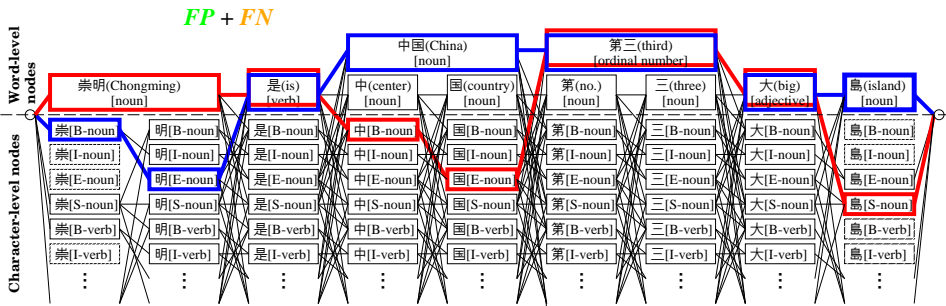
Training method

$FP + FN$



Our loss function $L(y_t, \hat{y}) = FP + FN$

Training method



Our loss function can reflect how bad the predicted path \hat{y} is compared to the correct path y_t .

MIRA

Concept

MIRA updates the weight vector $\mathbf{w}^{(i)}$ by keeping the norm of the change as small as possible [McDonald, 2006].

$$\begin{aligned} \mathbf{w}^{(i+1)} &= \operatorname{argmin}_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}^{(i)}\| & (1) \\ \text{s.t. } \forall \hat{\mathbf{y}} \in \text{best}_k(\mathbf{x}_t; \mathbf{w}^{(i)}) : \\ s(\mathbf{x}_t, \mathbf{y}_t; \mathbf{w}) - s(\mathbf{x}_t, \hat{\mathbf{y}}; \mathbf{w}) &\geq L(\mathbf{y}_t, \hat{\mathbf{y}}) \end{aligned}$$

- Eq (1) can be solved using Hildreth's algorithm.
- Check our paper (pp. 513–521) for our feature set.

Experiments

Benchmark corpus

- Chinese Penn Treebank (CTB)
- Used the same training, development, and test sets as reported in the literature.

Evaluation

- Used standard recall (R), precision (P), and F_1 .

Parameter tuning

Training corpus

no. of kwn words
= 472345

rare words
occur ≤ 1
time

no. of artificial unk words
= 21594

evaluate

Dev set

Baseline

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9208	0.9127	0.9167

Parameter tuning

Training corpus

no. of kwn words
= 459303

rare words
occur ≤ 2
times

no. of artificial unk words
= 34636

evaluate

Dev set

Baseline

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9208	0.9127	0.9167
2	0.9280	0.9202	0.9241

Parameter tuning

Training corpus

no. of kwn words
= 449679

rare words
occur ≤ 3
times

no. of artificial unk words
= 44260

evaluate

Dev set

Baseline

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9208	0.9127	0.9167
2	0.9280	0.9202	0.9241
3	0.9312	0.9257	0.9285

Parameter tuning

Training corpus

no. of kwn words
= 441595

rare words
occur ≤ 4
times

no. of artificial unk words
= 52344

evaluate

Dev set

Baseline

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9208	0.9127	0.9167
2	0.9280	0.9202	0.9241
3	0.9312	0.9257	0.9285
4	0.9302	0.9257	0.9280

Parameter tuning

Training corpus

no. of kwn words
= 434725

rare words
occur ≤ 5
times

no. of artificial unk words
= 59214

evaluate

Dev set

Baseline

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9208	0.9127	0.9167
2	0.9280	0.9202	0.9241
3	0.9312	0.9257	0.9285
4	0.9302	0.9257	0.9280
5	0.9301	0.9257	0.9279

Parameter tuning

Training corpus

no. of kwn words
= 449679

rare words
occur ≤ 3
times

no. of artificial unk words
= 44260

evaluate

Dev set

Baseline

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9208	0.9127	0.9167
2	0.9280	0.9202	0.9241
3	0.9312	0.9257	0.9285
4	0.9302	0.9257	0.9280
5	0.9301	0.9257	0.9279

Parameter tuning

Training corpus

Perform 10-fold cross validation
on the training corpus using
a baseline ($r=1$) model to extract

unidentified
unknown
words
(14875)

Error-driven

Results of Seg&Tag on dev set

r	R	P	F_1
-----	-----	-----	-------

Parameter tuning

Training corpus

no. of kwn words
= 459412

rare words
occur ≤ 1
time (21594)

unidentified
unknown
words
(14875)

no. of artificial unk words
= 34527

evaluate

Dev set

Error-driven

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9282	0.9197	0.9239

Parameter tuning

Training corpus

no. of kwn words
= 450196

rare words
occur ≤ 2
times (34636)

unidentified
unknown
words
(14875)

no. of artificial unk words
= 43743

evaluate

Dev set

Error-driven

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9282	0.9197	0.9239
2	0.9283	0.9218	0.9251

Parameter tuning

Training corpus

no. of kwn words
= 442423

rare words
occur ≤ 3
times (44260)

unidentified
unknown
words
(14875)

no. of artificial unk words
= 51516

evaluate

Dev set

Error-driven

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9282	0.9197	0.9239
2	0.9283	0.9218	0.9251
3	0.9317	0.9260	0.9288

Parameter tuning

Training corpus

no. of kwn words
= 435475

rare words
occur ≤ 4
times (52344)

unidentified
unknown
words
(14875)

no. of artificial unk words
= 58464

evaluate

Dev set

Error-driven

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9282	0.9197	0.9239
2	0.9283	0.9218	0.9251
3	0.9317	0.9260	0.9288
4	0.9312	0.9262	0.9287

Parameter tuning

Training corpus

no. of kwn words
= 429475

rare words
occur ≤ 5
times (59214)

unidentified
unknown
words
(14875)

no. of artificial unk words
= 64464

evaluate

Dev set

Error-driven

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9282	0.9197	0.9239
2	0.9283	0.9218	0.9251
3	0.9317	0.9260	0.9288
4	0.9312	0.9262	0.9287
5	0.9299	0.9261	0.9280

Parameter tuning

Training corpus

no. of kwn words
= 442423

rare words
occur ≤ 3
times (44260)

unidentified
unknown
words
(14875)

no. of artificial unk words
= 51516

evaluate

Dev set

Error-driven

Results of Seg&Tag on dev set

r	R	P	F_1
1	0.9282	0.9197	0.9239
2	0.9283	0.9218	0.9251
3	0.9317	0.9260	0.9288
4	0.9312	0.9262	0.9287
5	0.9299	0.9261	0.9280

Comparison with the best existing methods on CTB 5.0

Previous work

- **Jiang08a**: The perceptron algorithm with global features by Jiang *et al.* [ACL 2008]
- **Jiang08b**: The perceptron algorithm with lattice reranking by Jiang *et al.* [COLING 2008]
- **N&U07**: The hybrid method trained with the word-base Markov model and the character-based maximum entropy model by Nakagawa and Uchimoto [ACL 2007]

Results on CTB 5.0

Table: Comparison of F_1 results with previous studies on CTB 5.0

Method	Seg	Seg&Tag
N&U07	0.9783	0.9332
Jiang08a	0.9785	0.9341
Jiang08b	0.9774	0.9337

Results on CTB 5.0

Table: Comparison of F_1 results with previous studies on CTB 5.0

Method	Seg	Seg&Tag
N&U07	0.9783	0.9332
Jiang08a	0.9785	0.9341
Jiang08b	0.9774	0.9337
Ours (baseline)	0.9779	0.9360

Results on CTB 5.0

Table: Comparison of F_1 results with previous studies on CTB 5.0

Method	Seg	Seg&Tag
N&U07	0.9783	0.9332
Jiang08a	0.9785	0.9341
Jiang08b	0.9774	0.9337
Ours (baseline)	0.9779	0.9360
Ours (error-driven)	0.9787	0.9367

Additional comparison on CTB 3.0

Previous work

- **N&L04**: The character-based maximum entropy model by Ng and Low [EMNLP 2004]
- **Z&08**: The perceptron algorithm by Zhang and Clark [ACL 2008]
- **N&U07**: The hybrid method trained with the word-base Markov model and the character-based maximum entropy model by Nakagawa and Uchimoto [ACL 2007]

Results on CTB 3.0

Trial	Seg			Seg&Tag		
	N&U07	Z&C08	Ours (base.)	N&U07	Z&C08	Ours (base.)
1	0.9701	0.9721	0.9732	0.9262	0.9346	0.9358
2	0.9738	0.9762	0.9752	0.9318	0.9385	0.9380
3	0.9571	0.9594	0.9578	0.9023	0.9086	0.9067
4	0.9629	0.9592	0.9655	0.9132	0.9160	0.9223
5	0.9597	0.9606	0.9617	0.9132	0.9172	0.9187
6	0.9473	0.9456	0.9460	0.8823	0.8883	0.8885
7	0.9528	0.9500	0.9562	0.9003	0.9051	0.9076
8	0.9519	0.9512	0.9528	0.9002	0.9030	0.9062
9	0.9566	0.9479	0.9575	0.8996	0.9033	0.9052
10	0.9631	0.9645	0.9659	0.9154	0.9196	0.9225
Avg.	0.9595	0.9590	0.9611	0.9085	0.9134	0.9152

Table: Comparison of F_1 results for N&U07, Z&C08, and our baseline model on CTB 3.0

Note that the experimental setting is based on **10-fold cross validation** according to previous work.

Results on CTB 3.0

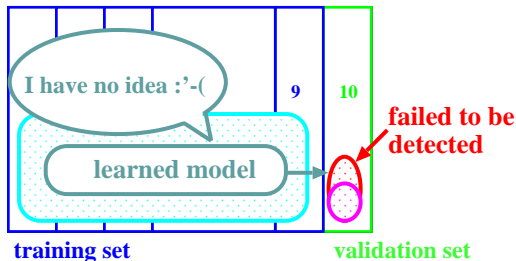
Table: Comparison of averaged F_1 results (by 10-fold cross validation) with previous studies on CTB 3.0

Method	Seg	Seg&Tag
N&L04	0.9520	-
N&U07	0.9595	0.9085
Z&C08	0.9590	0.9134
Ours (baseline)	0.9611	0.9152

Conclusion

Our approach has two important advantages:

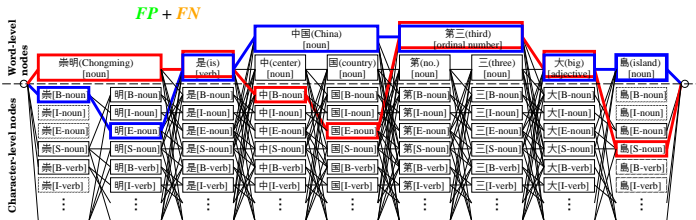
- 1 a simple method for **learning characteristics of unknown words from errors** in the training corpus



Conclusion

Our approach has two important advantages:

- ① a simple method for **learning characteristics of unknown words from errors** in the training corpus
- ② an efficient training method for the word-character hybrid model based on **MIRA**



Conclusion

Our approach has two important advantages:

- ① a simple method for **learning characteristics of unknown words from errors** in the training corpus
- ② an efficient training method for the word-character hybrid model based on **MIRA**

Our approach is language-independent, and we plan to apply it to other Asian languages such as Thai and Japanese in future work.

Thank you.

Questions?

Generic online learning algorithm

Input: Training set $\mathcal{S} = \{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1}^T$

Output: Feature weight vector \mathbf{w}

- 1: $\mathbf{w}^{(0)} = \mathbf{0}; \mathbf{v} = \mathbf{0}; i = 0$
- 2: **for** $iter = 1$ to N **do**
- 3: **for** $t = 1$ to T **do**
- 4: $\mathbf{w}^{(i+1)} = \text{update } \mathbf{w}^{(i)}$ according to sample $(\mathbf{x}_t, \mathbf{y}_t)$
- 5: $\mathbf{v} = \mathbf{v} + \mathbf{w}^{(i+1)}$
- 6: $i = i + 1$
- 7: $\mathbf{w} = \mathbf{v} / (N \times T)$