非中文母語學習者中文寫作用詞錯誤偵測及更正之研究

Detection and Correction of Chinese Word Usage Errors for Learning Chinese as a Second Language

研究生：薛祐婷

指導教授：陳信希教授

106 年 7 月 20 日

Outline

- 1 Introduction
- 2 Related Work
- 3 The HSK Word Usage Error Dataset
- 4 Segment-level WUE Detection
- 5 Token-level WUE Detection
- 6 WUE Correction
- 7 Conclusion

1 Introduction

- Motivation
- Chinese Word Usage Error (WUE)
- Overview

1 Intro – Motivation

- More and more people around the world choose to learn Chinese as their second language.
- Grammatical error detection and correction (GEC) tools
  - Most studies are based on English learner data
  - But Chinese differs substantially from English
- Learner data is required!
  - Mistakes made by non-natives differ from those by natives
    - E.g. English verb tense error
      Native speakers: seldom
      Non-natives: one of the most common mistakes
  - Realistic evaluation on GEC systems targeting language learners

1 Intro – Motivation

- Ground-truth of correction must be manually annotated by trained annotators → available amount of data is limited
- At the time of this study, the largest available Chinese learner corpus was HSK dynamic composition corpus (by Beijing Language and Culture University).
- Word usage error (WUE) is the **most frequent** lexical-level error in the HSK corpus
  → WUE detection and correction tool is worth developing

1 Intro – Chinese WUE

- In Chinese sentences, a WUE is a grammatically or semantically incorrect token.

- HSK sub-types of WUE

Sub-type	Example
(1) Character disorder within a word	首先{CC先首} 眾所周知{CC眾所知周}
(2) Incorrect selection of a word	雖然現在還沒有實現{CC實踐}，......
(3) Non-existent word	殘留量{CC潛留量} 農產品{CC農作品}
(4) Word collocation error	最好的辦法是兩個都保持{CC走去}平衡。

1 Intro – Chinese WUE

- No sub-type annotation / division not clear

Morphological Error (W)
Non-existent word

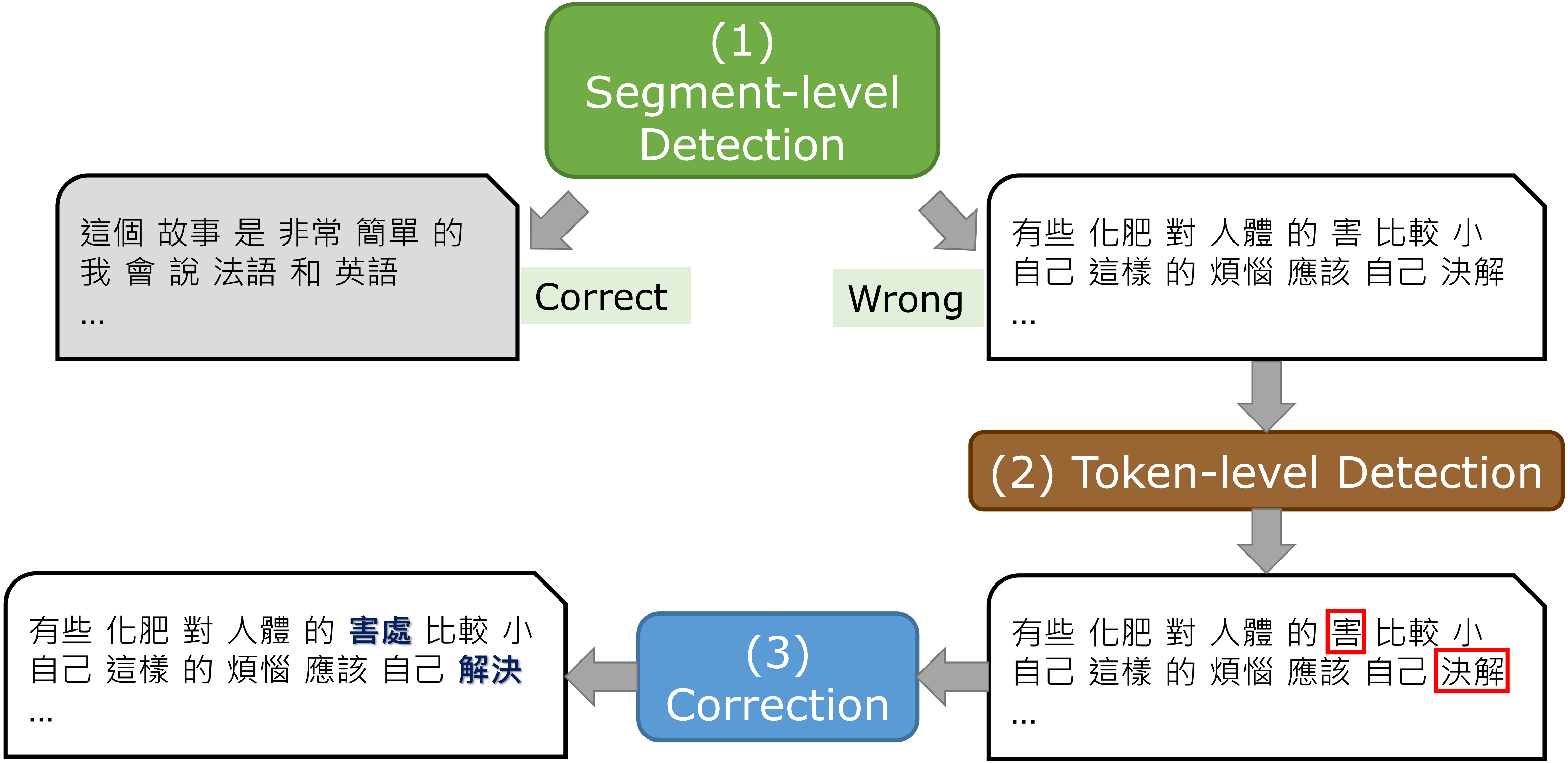
Usage Error (U)
Existent word

Sub-type
(1) Character disorder within a word
(2) Incorrect selection of a word
(3) Non-existent word
(4) Word collocation error

- Look up the erroneous token in a dictionary
  Not found → W-error

Sub-type	# instances
W	4,010
U	13,314

1 Intro – Overview



2 Related Work

- Grammatical Error Detection and Correction in English
- Grammatical Error Detection and Correction in Chinese
- Distributed Word Representations

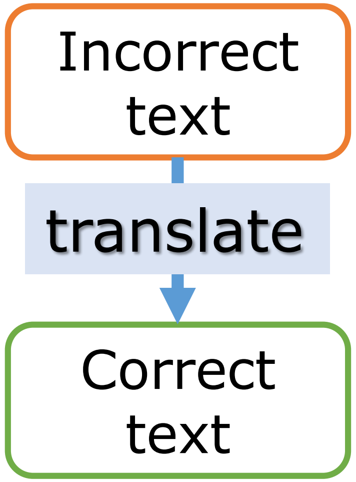
2 Related Work – GEC in English

- Leacock et al. (2014): handbook, comprehensive survey of GEC
  - Annotated learner data is important, but the **amount is limited**
    → difficult to build robust statistical model
  - <u>Solution</u>: Combine statistical models with rule-based approaches
  - <u>Solution</u>: Construct artificial error corpora
    - Distribution of artificial training data could differ from that of real test data
    - Ends up learning the way of synthesizing data, instead of language learners' pattern of making mistakes?
  - <u>Solution</u>: Make use of large "grammatical" text corpora
    - Difference in domain and style
      - Large corpora: newspaper or Wikipedia text, more formal
      - Language learners (especially beginners): write about themselves and daily lives
    - Low frequency = wrong usage?

2 Related Work – GEC in English

- Evaluation
  - Different typology of errors, different datasets → hard to compare
  - Shared tasks: evaluate GEC systems in a standardized manner
    - HOO 2011 (Dale and Kilgarriff, 2011), HOO 2012 (Dale et al., 2012)
    - CoNLL 2013 (Ng et al., 2013): article/determiner, preposition, noun number, verb form, subject-verb agreement
    - CoNLL 2014 (Ng et al., 2014): 28 error types
  - Approaches
    - Language models
    - Machine learning-based classifiers
    - Rule-based classifiers
    - Machine translation models

2 Related Work – GEC in English



- **Machine translation** approach to GEC
  - Advantage: no need to explicitly formulate types of the errors
  - Phrase-based statistical machine translation (SMT) framework
    - Dahlmeier and Ng (2011): **add phrase table entries** to handle semantic collocation errors due to similarity in writer's first language (L1) e.g. watch(看) / see(看)
    - Chollampatt et al. (2016b): add Neural Network Global Lexicon Model (NNGLM) & Neural Network Joint Model (NNJM) **features**
    - Chollampatt et al. (2016a): **adapt** a general NNJM with L1-specific text Kullback-Leibler divergence regularization term
- **Detection only**: Rei and Yannakoudakis (2016)
  - Correction can be subjective
  - Compare models: Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM)

2 Related Work – GEC in Chinese

- Shared Task for **Chinese Grammatical Error Diagnosis** (Yu et al., 2014; Lee et al., 2015, 2016)
  - Types:
    (1)redundant word (2)missing word (3)word disorder (4)word selection
  - Performance reported on whole dataset → unclear whether some systems are better at certain types
  - Only deal with detection but not correction
- Huang and Wang (2016): use LSTM for the above shared task
  - Randomly initialized word vector
  - Trained only on learner data, without incorporating information derived from external well-formed text
    → performance might be limited by the small amount of learner data

2 Related Work – GEC in Chinese

- HSK corpus-based research
  - Word Ordering Errors (WOEs)
    - Yu and Chen (2012): WOE detection with syntactic features, web corpus features, perturbation features
    - Chen et al. (2014): recommend correct word orderings with ranking SVM
  - Preposition Selection: Huang et al. (2016)
    - Gated recurrent unit (GRU)-based model
    - Select most suitable one from a closed set of 43 prepositions given context
    - Detect and correct preposition errors
- How to correct WUEs involving **open-set** types of words such as verbs and nouns?
  - Could be much more difficult since candidate set is huge
- **To the best of our knowledge, this is the first research dealing with general-type Chinese WUE correction.**

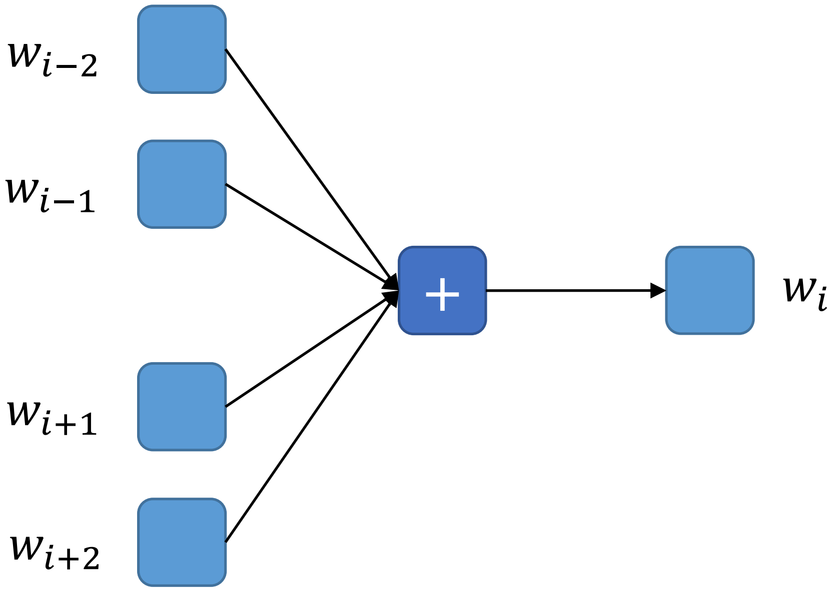
2 Related Work – Distributed Word Representations

- Distributed word representations (word embeddings) derived from neural network models have become popular in NLP
  - Assumption: similar words share similar context
  - Can be trained on large text corpora in an unsupervised manner
  - Real-valued vectors with low dimensionality (compared to vocabulary size)
  - Encode syntactic and semantic information implicitly beyond surface forms (Mikolov et al., 2013b)
- WUEs involve syntactic or semantic problems → vector representations could be promising
  - Three types of word embeddings are adopted throughout this research
  1. Word2vec CBOW/Skip-gram Word Embeddings
  2. CWINDOW/Structured Skip-gram Word Embeddings
  3. Character-enhanced Word Embedding (CWE)

2 Related Work – Word2vec CBOW & SG

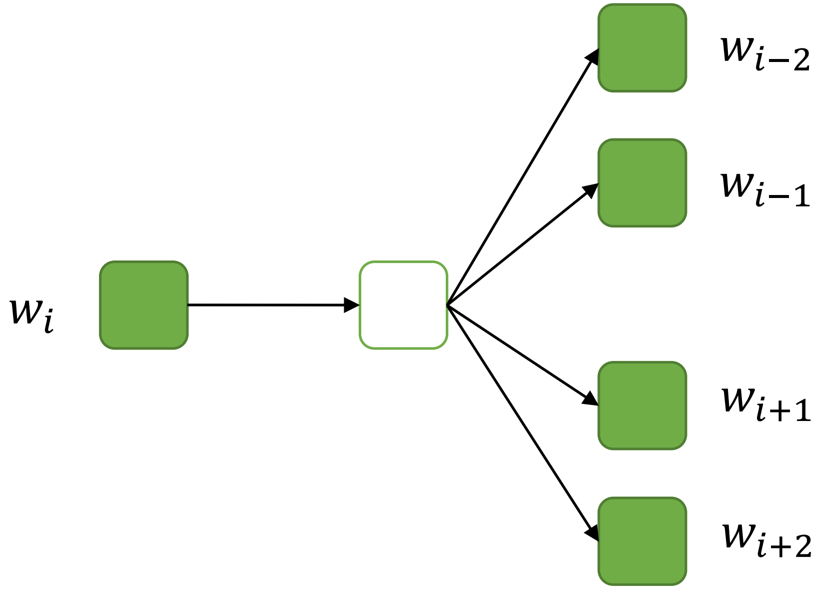
Continuous bag-of-words (CBOW)

Context predict target



Skip-gram (SG)

Target predict context

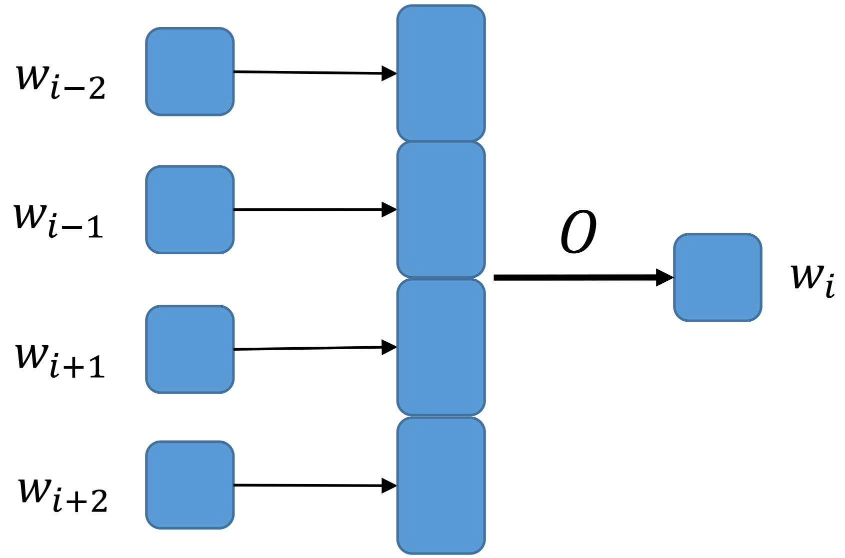


- Every context word treated equally → information of word order not preserved

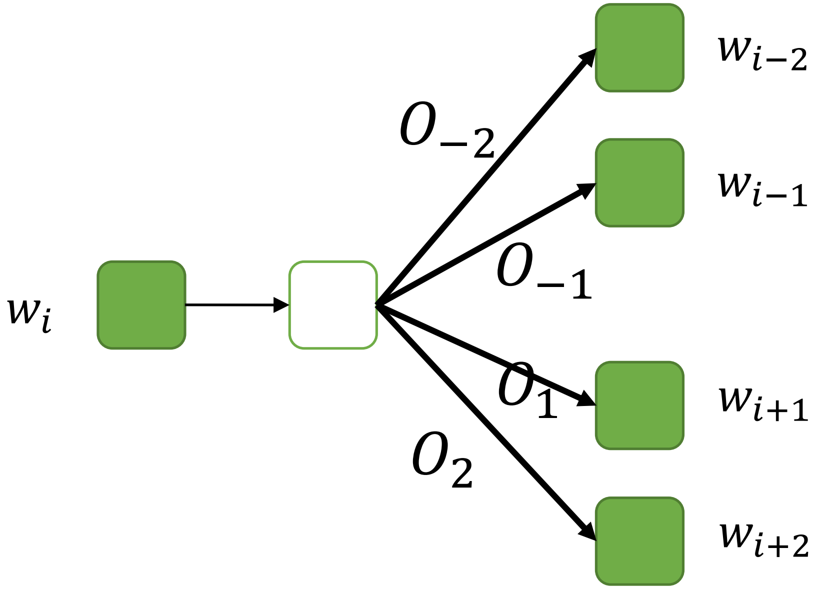
Mikolov et al. (2013a)

2 Related Work – CWIN & Struct-SG

Continuous window (CWIN)



Structured Skip-gram (Struct-SG)

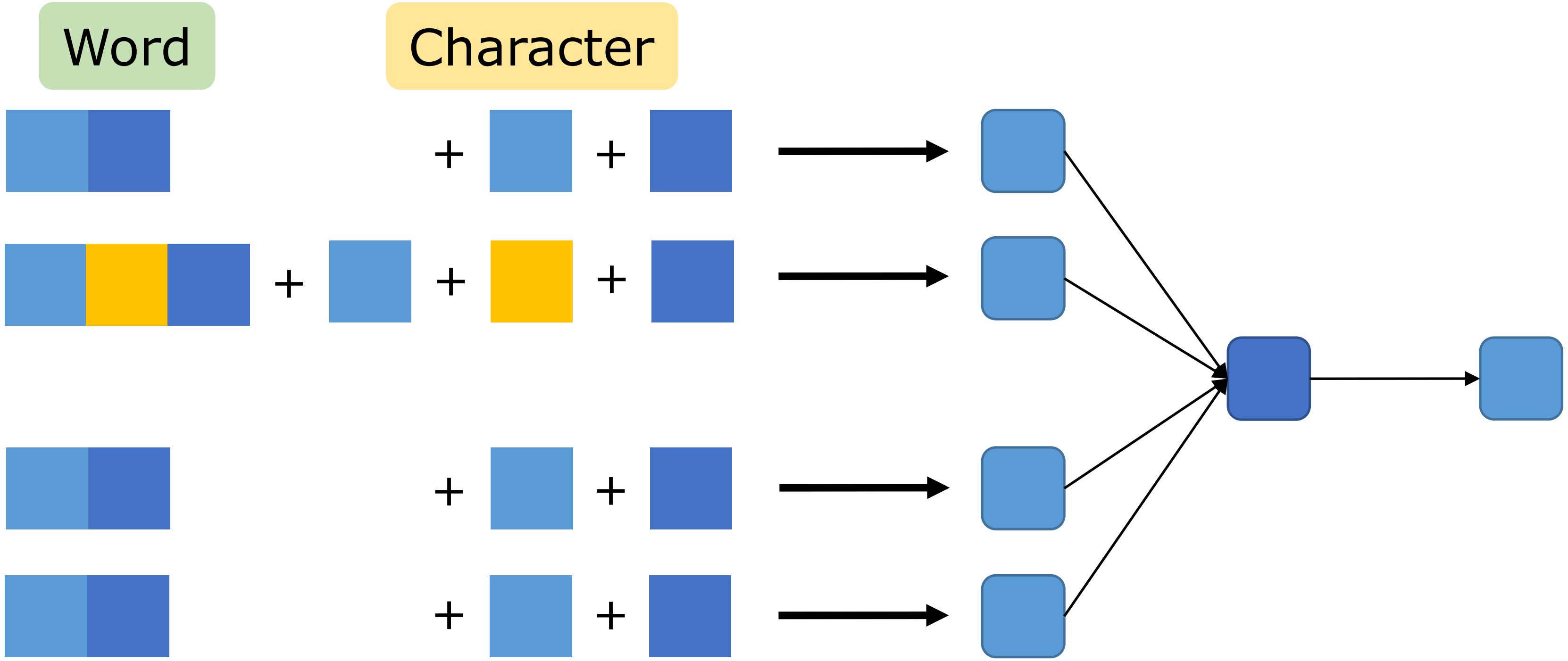


- Consider order of context words
- Projection matrices
- Useful for **syntactic** tasks

Ling et al. (2015)

2 Related Work – CWE

- Character-enhanced Word Embedding (CWE)
  - Chinese **characters** usually take on their **own meanings**.
  - Word meaning can be inferred even without context!
    - E.g. 公車(bus) = 公(public) + 車(vehicle)



Chen et al. (2015)

3 HSK WUE Dataset

- Data Collection
- Linguistic Processing
- Split Sentence into Segments & Filtering

3 Dataset – Data Collection

- Split sentence by 。 ？ ！

Correct sentence	我曾經到台灣讀書交了很多外國朋友，我們是用漢語說話的。
Wrong sentence	可想而知，他們長大以後會遇到很多的麻煩，甚至不適應生活，造成**不甚**後果。
Correction of the wrong sentence	可想而知，他們長大以後會遇到很多的麻煩，甚至不適應生活，造成**不良**後果。

- A sentence containing *n* errors → *n* sentences with one error
  - A sentence may contain multiple errors, including errors of types other than WUE

○○E1○○○○○○E2○○○○E3○

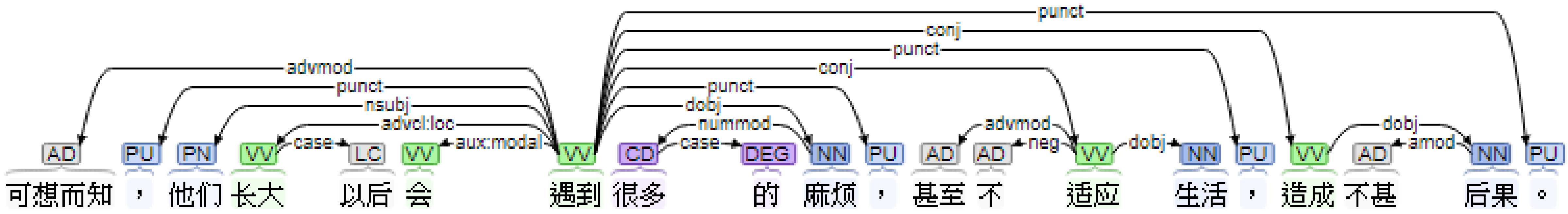
○○E1○○○○○○○○○○○○○

○○○○○○○○○E2○○○○○○

○○○○○○○○○○○○○○E3○

3 Dataset – Linguistic Processing

- Stanford CoreNLP
  - Word Segmentation
    - Sentence length = # tokens
  - POS Tagging
    - Tagging set: Chinese Penn Treebank
  - Dependency Parsing



- Will extract features based on these three levels of information

http://corenlp.run/

3 Dataset – Split Segments & Filtering

- Binary classification of correct & wrong sentence → 80% accuracy only with sentence length threshold!
  - A Chinese sentence is usually composed of several segments separated by ，
  - E.g. 3 segments: 如果我當推銷員的話，為了早點兒習慣，打算盡可能努力。
  - Longer sentence → more likely to make grammatical errors somewhere

	Average length
Correct sentence	7.8
Wrong sentence	25.6

- → Split into **segments** with punctuation marks (POS tag = PU)
- Filter segments:
  - Contain digits or English alphabets
  - Length < 5 (e.g. “您好！”, “不過，…”, “那時，…”)

	#
Correct segments	63,612
Wrong segments	17,324

(1)
Segment-level
Detection

這個 故事 是 非常 簡單 的
我 會 說 法語 和 英語
…

Correct

Wrong

有些 化肥 對 人體 的 害 比較 小
自己 這樣 的 煩惱 應該 自己 決解
…

4 Segment-level WUE Detection

- Features
- Machine Learning Classifiers
- Results & Discussion

4 Seg. Detection – Features

1. Google N-gram Features (**G**)
2. Dependency Count Features (**D**)
3. Dependency Bigram Features (**B**)
4. Single-character Features (**S**)
5. Word Embedding Features (**W**)

- All combined with segment length (*s_len*)

4 Seg. Detection – G Features

- Chinese version of Google Web 5-gram (Liu et al., 2010)
- MLE n-gram probability
  - E.g. tri-gram: $\mathrm{p}(w_i|w_{i-2}, w_{i-1}) = \frac{c(w_{i-2}, w_{i-1}, w_i)}{c(w_{i-2}, w_{i-1})}$
- $\mathbf{G} = (g_2, g_3, g_4, g_5)$, where

$$g_n = \sum_{i=n}^{L} \mathrm{p}(w_i|w_{i-n+1}, \dots, w_{i-1})$$

- Combine with *s_len* → let model handle the relationship between sum of probability & *s_len*
  - Might not be linear

4 Seg. Detection – D Features

- Errors in a sentence affect the result of segmentation and parsing.

Correct segment	**Wrong segment**
以下 **介紹** 一下 我 的 簡歷 和 經驗 。	以下 **紹 介** 一下 我 的 簡歷 和 經驗 。
nsubj(介紹-2, 以下-1)	nsubj(介-3, 以下-1)
root(ROOT-0, 介紹-2)	**advmod(介-3, 紹-2)**
advmod(介紹-2, 一下-3)	root(ROOT-0, 介-3)
assmod(經驗-8, 我-4)	advmod(介-3, 一下-4)
case(我-4, 的-5)	assmod(經驗-9, 我-5)
…	case(我-5, 的-6)
	…

4 Seg. Detection – D Features

- Example

聽說 貴 公司 在 國內 很 有名 ， 外國 顧客 也 很多 。

root(ROOT-0, **聽說-1**)
nn(**公司-3**, **貴-2**)
nsubj(**有名-7**, **公司-3**)
case(**國內-5**, **在-4**)
prep(**有名-7**, **國內-5**)
advmod(**有名-7**, **很-6**)
ccomp(**聽說-1**, **有名-7**)
nn(顧客-10, 外國-9)
nsubj(很多-12, 顧客-10)
advmod(很多-12, 也-11)
conj(**有名-7**, 很多-12)

Internal count		External count	
nn_int_cnt	1	nn_ext_cnt	1
nsubj_int_cnt	1	nsubj_ext_cnt	1
case_int_cnt	1	case_ext_cnt	1
prep_int_cnt	1	prep_ext_cnt	1
advmod_int_cnt	1	advmod_ext_cnt	1
ccomp_int_cnt	1	ccomp_ext_cnt	1
conj_int_cnt	0	conj_ext_cnt	1
all_dep_int_cnt	6	all_dep_ext_cnt	7

4 Seg. Detection – B Features

- Example: 親身 **體會** 了 一場 永遠 難忘 的 電單車 **意外**
- 6 words between 意外 and 體會 → out of the range of 5-gram

- **Dependency bigrams**
  - nsubj(體會-2, 親身-1) → 親身 體會
  - dobj(體會-2, 意外-9) → 體會 意外

	Bigram	Frequency
Wrong	體會 意外	0
Correct	經歷 意外	167

- Sum bigram probabilities for each dependency type
  - Collocating behavior might vary with dependency type
  - Internal sum: *dep*_int_sum_prob, all_ext_sum_prob
  - External sum: *dep*_int_sum_prob, all_ext_sum_prob

4 Seg. Detection – S Features

- A non-existent Chinese word (W-error) is usually separated into several **single-character words** after segmentation
  → important indicator of WUE
  1. ***seg_cnt***: # contiguous single-character blocks
  2. ***len2above_seg_cnt***: # contiguous single-character blocks with length > 2
  3. ***max_seg_len***: length of the maximum contiguous single-character block
  4. ***sum_seg_len***: sum of the lengths of all contiguous single-character blocks

- Example:
  而且 我 認為 貴 公司 是 我國 最 大 的

Feature	**Value**
seg_cnt	4
len2above_seg_cnt	1
max_seg_len	3
sum_seg_len	6

4 Seg. Detection – W Features

- Train CBOW/SG word embeddings on the Chinese part of the ClueWeb09 dataset

Embedding size	400
Window size	5
# negative samples	10
Iterations	20

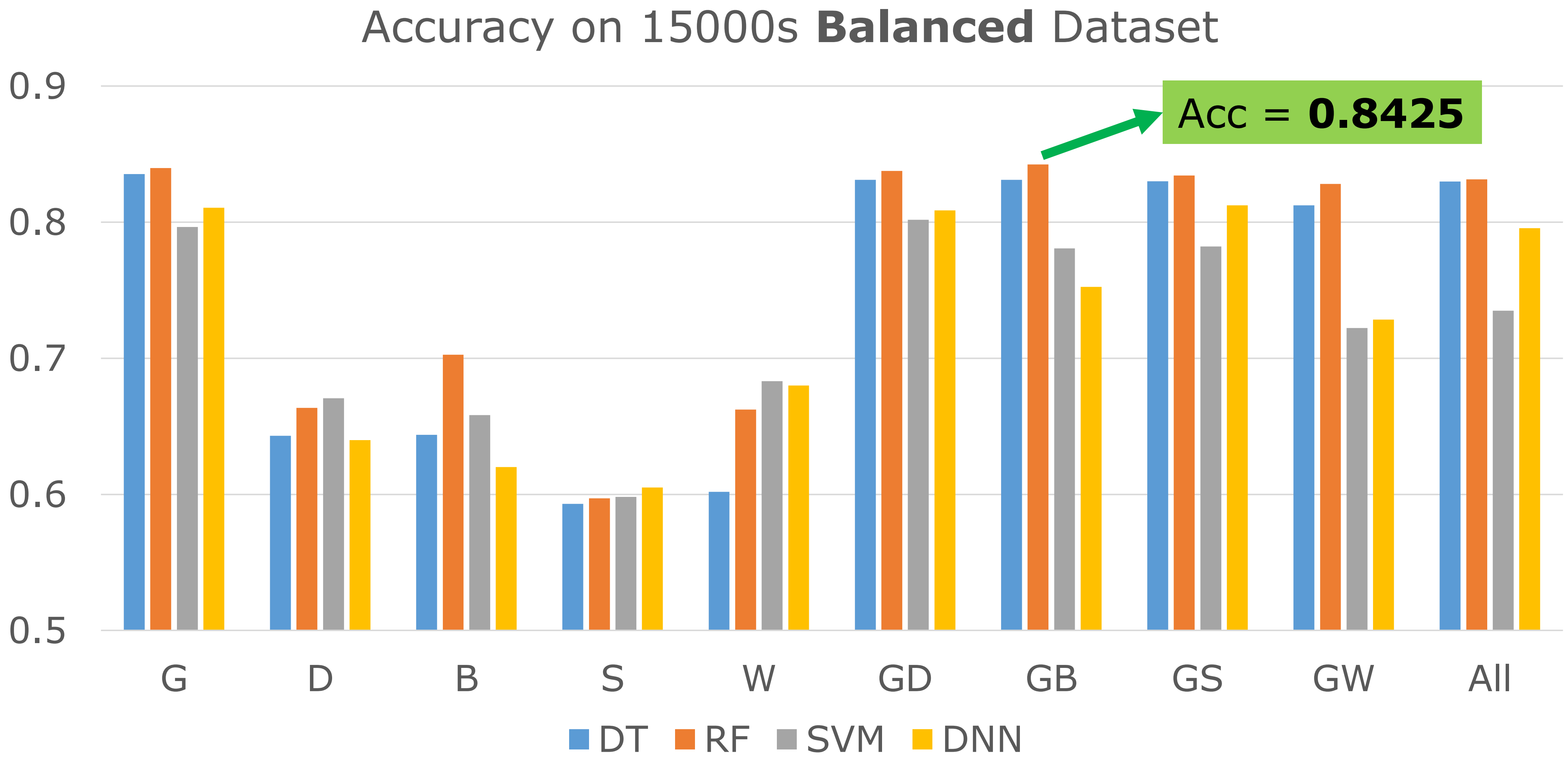
- Concatenate CBOW and SG embeddings into a feature vector W (dim=800)

4 Seg. Detection – Classifiers

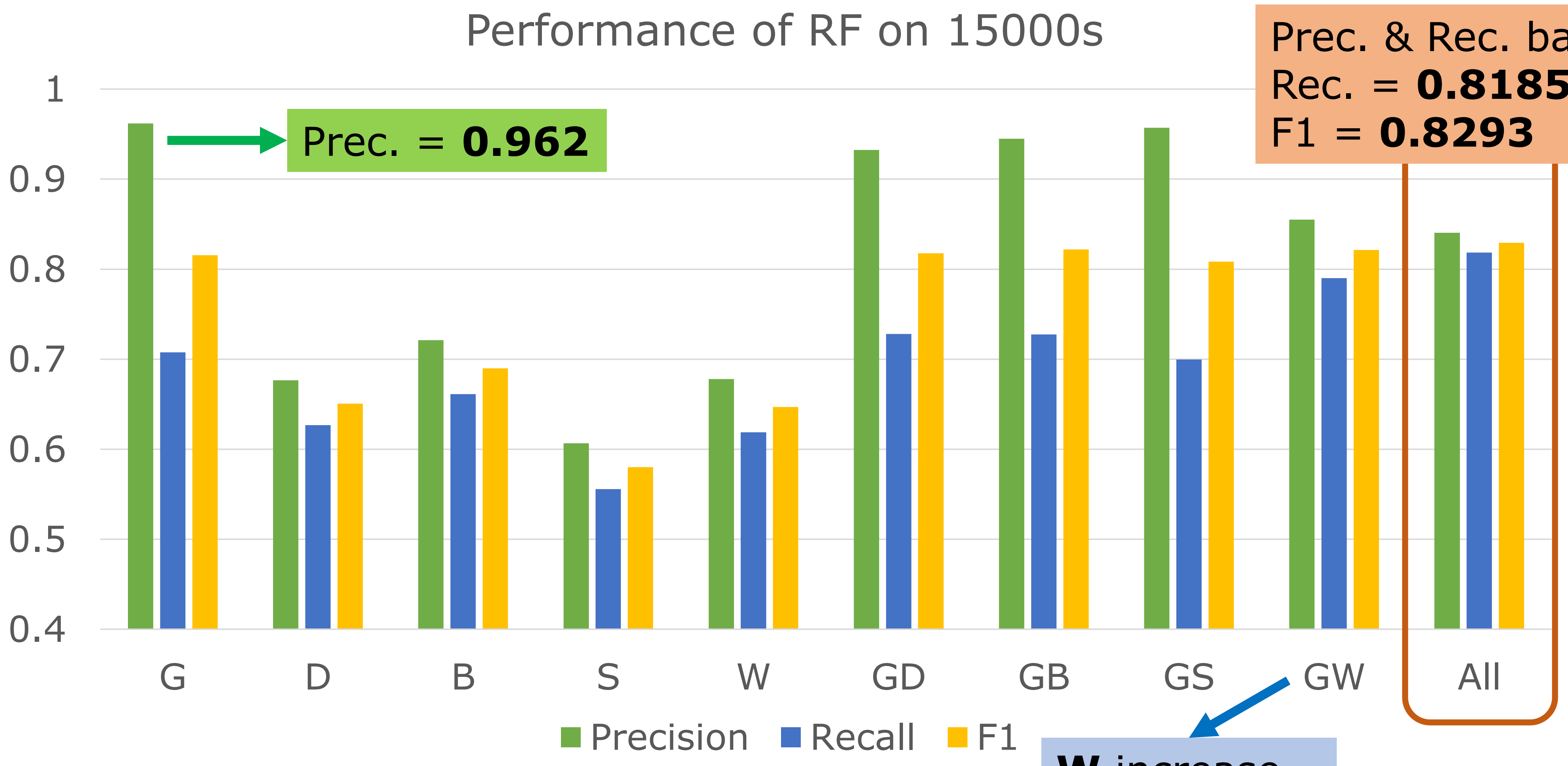
- Decision Tree (DT)
- Random Forest (RF)
- Support Vector Machine with RBF kernel (SVM)
- Feed-forward Neural Network (Deep Neural Network, DNN)

- Scale feature values to zero mean and unit variance for SVM & DNN

4 Seg. Detection – Results & Discussion



4 Seg. Detection – Results & Discussion

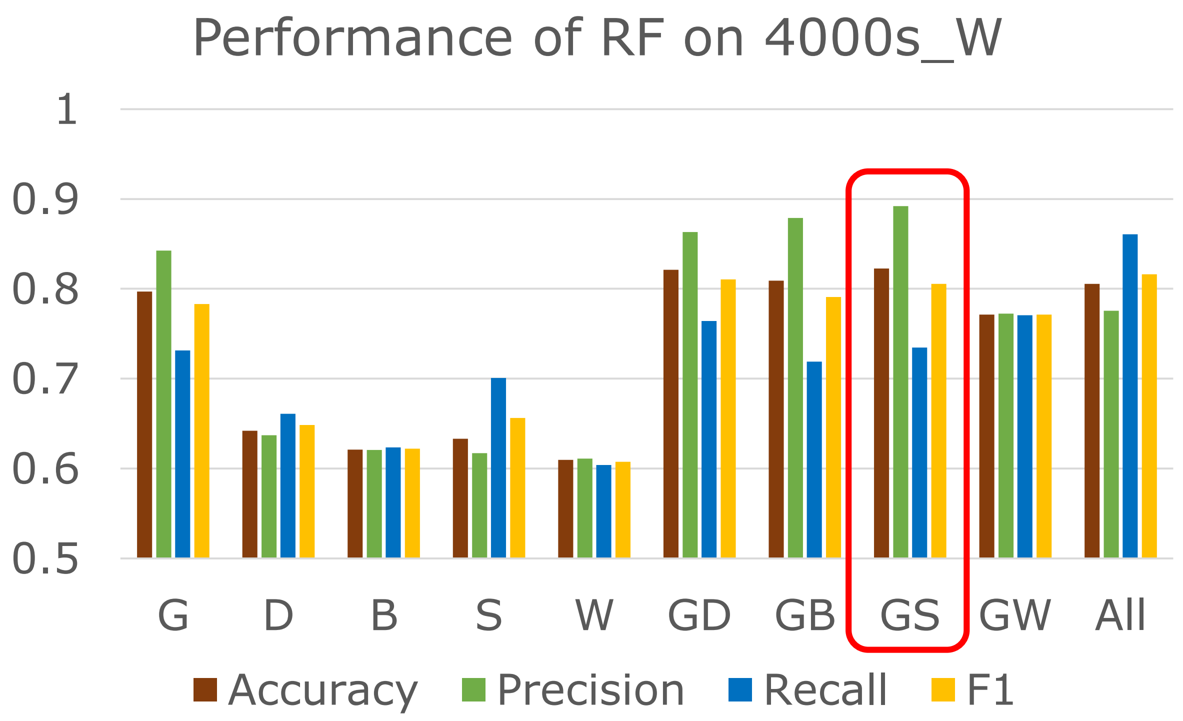


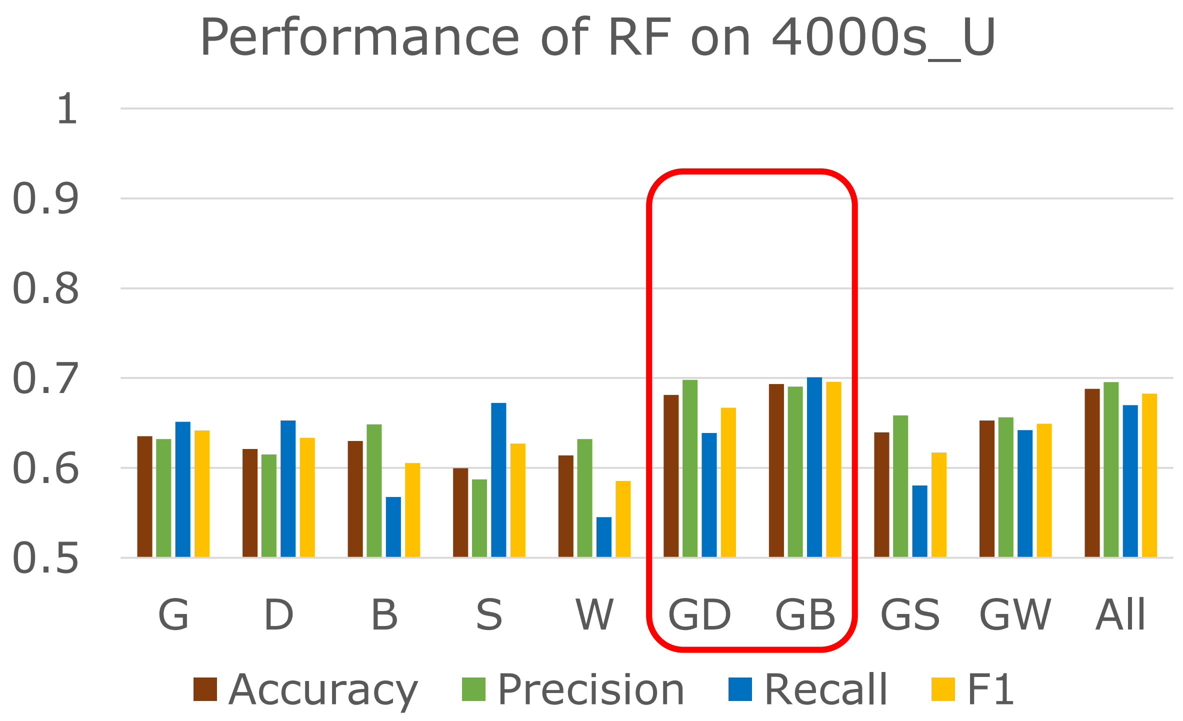
Prec. & Rec. balanced
Rec. = **0.8185**
F1 = **0.8293**

W increase recall by 9%

4 Seg. Detection – Results & Discussion

- Sub-type evaluation
  - Sample 4,000 segments from each WUE subtype and combine them with 4,000 correct segments respectively → 4000s_W and 4000s_U





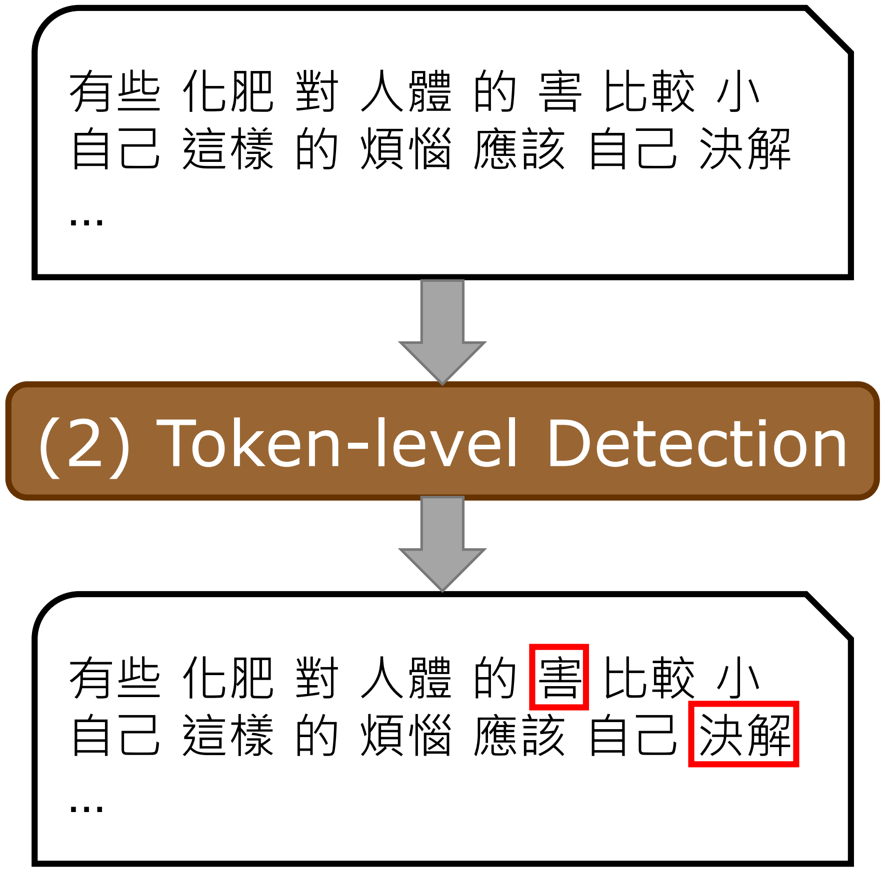
4 Seg. Detection – Results & Discussion

- S not very effective on its own, but G+S is powerful for W-errors
  - Existence of single-character words
    → not sufficient to conclude that there is something wrong
  - **Correct** segment: 有 人 對 她 說
    - Many single-character words due to its grammatical structure
  - **Wrong** segment: 他們 應該 <u>共 敬</u> 父母 //correction: <u>尊敬</u>
    - Bigram probability of "共 敬" < 0.0001

- For U-errors, D and B, which are derived from the result of dependency parsing, are more useful
  - Help handle collocation errors better, especially those involving long-distance dependency

4 Seg. Detection – Conclusion

- Best result:
  accuracy = 0.8425
  precision = 0.9450
  recall = 0.7274
  F1 = 0.8220
- RF is the best classifier for the proposed features
- With suitable model and combination of features, **precision** can be up to **96.2%**.
  - If a segment is classified as wrong by our high-precision model, it is very likely that there is indeed some WUE.



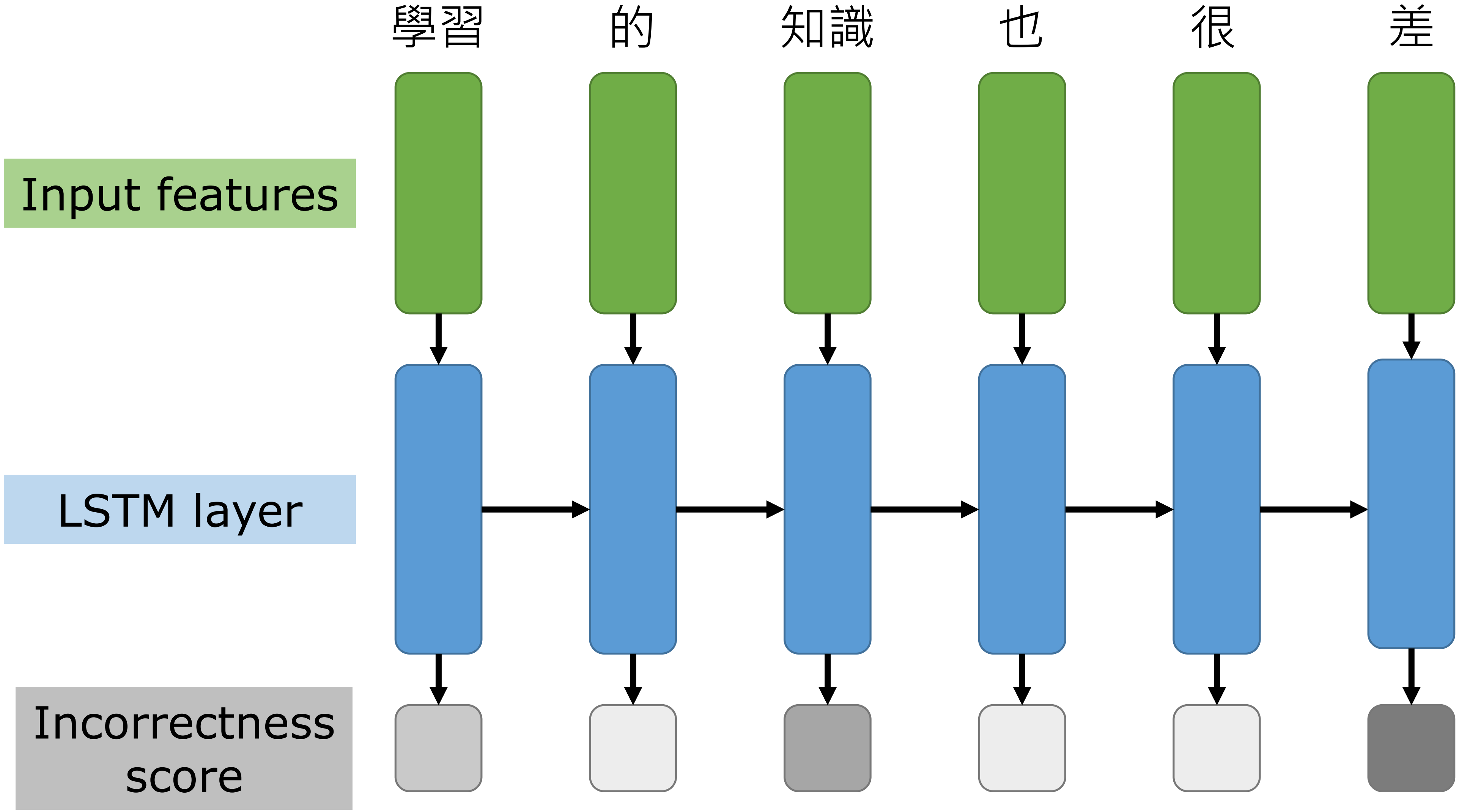
5 Token-level WUE Detection

- Dataset
- Bidirectional LSTM model
- Features
- Evaluation
- Results and Analysis

5 Token Detection – Dataset

- "Wrong" part of the 15000s dataset used in previous stage
- Each sentence segment has **exactly one** token-level position that is erroneous
- Filter out any segment whose corrected version differs from it by more than one token due to segmentation issue
  - Some W-error instances are filtered out since the erroneous token is segmented into several words
  - Focus on errors that can be corrected by **replacing one single token**
- Total: 10,510 sentence segments
  - 10% validation
  - 10% testing
  - 80% training

5 Token Detection – LSTM

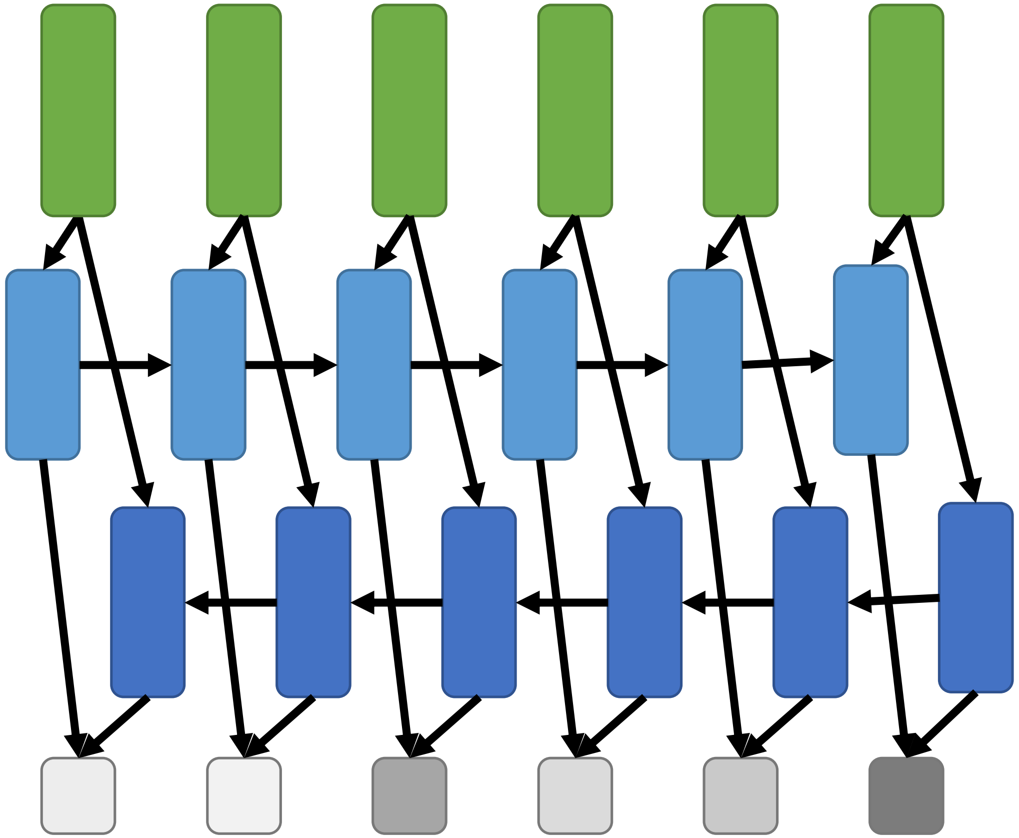


5 Token Detection – Bidirectional LSTM

- Bidirectional LSTM

Forward LSTM

Backward LSTM



- Example: 店 是 爸爸 (*留在,留給) <u>我們</u> 的
  - Need the **future** information to detect the error

5 Token Detection – Features

<table>
<tr><td>Word</td><td>當時</td><td>我們</td><td>都</td><td>相信</td><td>*農作品</td><td>沒有</td><td>農藥</td></tr>
<tr><td colspan="8">• Embedding size = 400, trainable
1. Random
2. CBOW / SG
3. CWIN / Struct-SG: consider the order of context words</td></tr>
<tr><td>POS</td><td>NT</td><td>PN</td><td>AD</td><td>VV</td><td>NN</td><td>VE</td><td>NN</td></tr>
<tr><td colspan="8">• Embedding size = 20, trainable (# unique POS = 30)
• Random</td></tr>
<tr><td>OOV</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>2gram</td><td>-1</td><td>P(我們|當時)
0.0109</td><td>0.0116</td><td>0.0004</td><td>0.0000</td><td>0.0000</td><td>0.000017</td></tr>
<tr><td>3gram</td><td>-1</td><td>-1</td><td>P(都|當時,我們)
0.0621</td><td>0.0022</td><td>0.0000</td><td>0.0000</td><td>0.0000</td></tr>
</table>

5 Token Detection – Evaluation

- Accuracy, MRR
- Hit@2
  - One most common type of WUEs is **collocation error**
  - Wrong segment: 學習 的 **知識** 也 很 **差** //Problem: word pair (知識, 差)
  - Correction 1: 學習 的 知識 也 很 **不足**
  - Correction 2: 學習 的 **態度** 也 很 差
  - Both correction acceptable
    - Which is better? highly depend on the context, or even the intended meaning
  - Proposing two closely-related potentially erroneous tokens can be useful
- Hit@20%
  - Take segment length (*s_len*) into account
  - Hit@r%: regard an instance as correct if the answer is ranked within the top $\max(1, \lfloor s\_len * r\% \rfloor)$ candidate(s)

5 Token Detection – Results & Analysis

Model	**Features**	**Accuracy**	**MRR**	**Hit@2**	**Hit@20%**
Rand. baseline	-	0.1239	0.3312	0.2478	0.1611
LSTM	Rand. Emb.	0.4186	0.6010	0.7222	0.6565
	CBOW	0.4072	0.5923	0.7155	0.6432
	SG	0.4072	0.5910	0.7146	0.6365
	CWIN	0.4853	0.6537	0.7774	0.7031
	Struct-SG	0.4710	0.6412	0.7650	0.6889
Bi-LSTM	CWIN	0.4795	0.6547	0.7840	0.7174
	+ POS	**0.5138**	**0.6789**	0.8097	0.7479
	+ N-gram	0.4948	0.6719	**0.8173**	**0.7507**

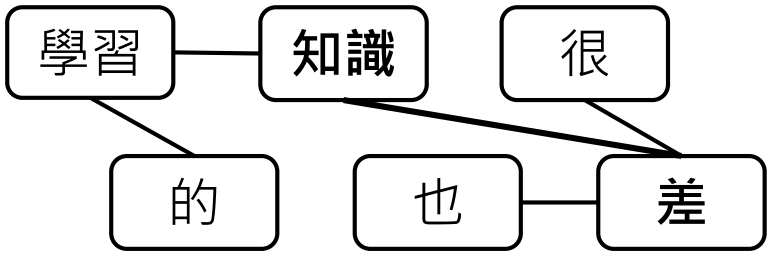
5 Token Detection – Results & Analysis

- LSTM vs. Bi-LSTM
  - Hit@20% rates on different length of segments
  - CWIN + POS + n-gram

Length (#tests)	# proposed	LSTM	Bi-LSTM
< 10 (645)	1	0.7426	**0.7659**
10 ~ 14 (137)	2	0.6908	**0.7319**
15+ (89)	3+	**0.7416**	0.7079

5 Token Detection – Results & Analysis

- Justification for **hit@2**: WUE usually involves a pair of words
- Are top two candidates proposed really closely related?
- Examine **dependency distance**
  - Undirected graph, node = word, edge = dependency relation
  - $dis(c_1, c_2)$: shortest path distance between first candidate c_1 and second candidate c_2 *// Average segment length = 9.24*
  - a: ground-truth error position



Bi-LSTM(CWIN + POS + n-gram)	
# correct ($c_1 = a$)	520 (49.48%)
# tests where $c_2 = a$	339 (32.25%)
Average $dis(c_1, c_2)$ when $c_2 = a$	2.07
# tests where $c_2 = a$ and $dis(c_1, c_2) = 1$	129 (12.27%)

5 Token Detection – Results & Analysis

- Effectiveness of **POS features**
  - POS tagger trained on well-formed text, but learner data is noisy
  - POS tag **changed** after correction: 26.7%

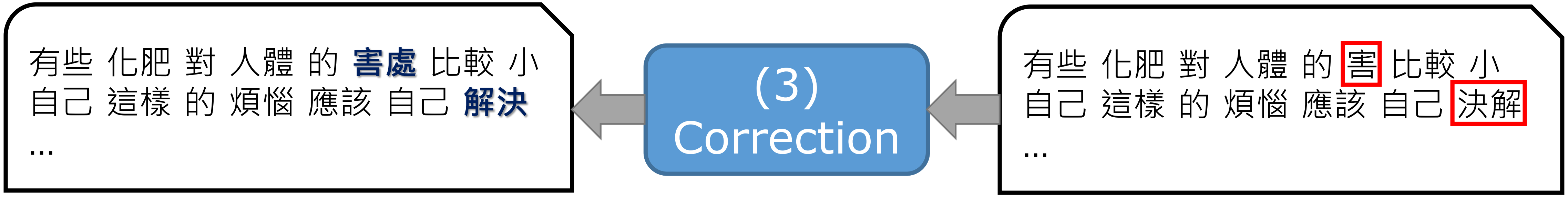
POS (# tests)	CWIN	CWIN+POS
VV (325)	0.8123	**0.8185**
NN (282)	0.6879	**0.7447**
AD (134)	0.6194	**0.7015**

Invalid in Chinese

	應該	有	別人	的	*盡力
POS	VV	VE	NN	DEC	AD
w/o POS	0.048	**0.226**	0.030	0.016	0.042
w/ POS	0.010	0.066	0.031	0.071	**0.077**

5 Token Detection – Conclusion

- Feature
  - **External information**: pre-trained word embedding, POS, n-gram
  - CWIN/Struct-SG are better word features for WUE detection.
  - POS information can be useful for detecting ungrammatical construction.
- Model
  - Bi-LSTM is more preferred than LSTM
- The best model can rank ground-truth error position within top two in 80.97% cases
  - Top two candidates usually closely related, according to dependency distance



6 WUE Correction

- Criteria for Correction
- Correction Generation Model
- Features
- Language Model Re-ranking
- Automatic Evaluation
- Human Evaluation

6 Correction – Criteria

- Given a token in a segment that is known to be erroneous, we aim to generate a suitable correction for it.

- Criteria of a suitable correction

1. **Correctness**: result must be a syntactically and semantically correct Chinese sentence segment.
2. **Similarity**: meaning must be as close to the writer's intended meaning as possible.

6 Correction – Criteria

- Example 1

		Correctness	Similarity
Wrong segment	生活方式已經**猛烈**地改變了		
Correction 1	生活方式已經**強烈**地改變了	X	O
Correction 2	生活方式已經**緩慢**地改變了	O	X
V Correction 3	生活方式已經**劇烈**地改變了	O	O

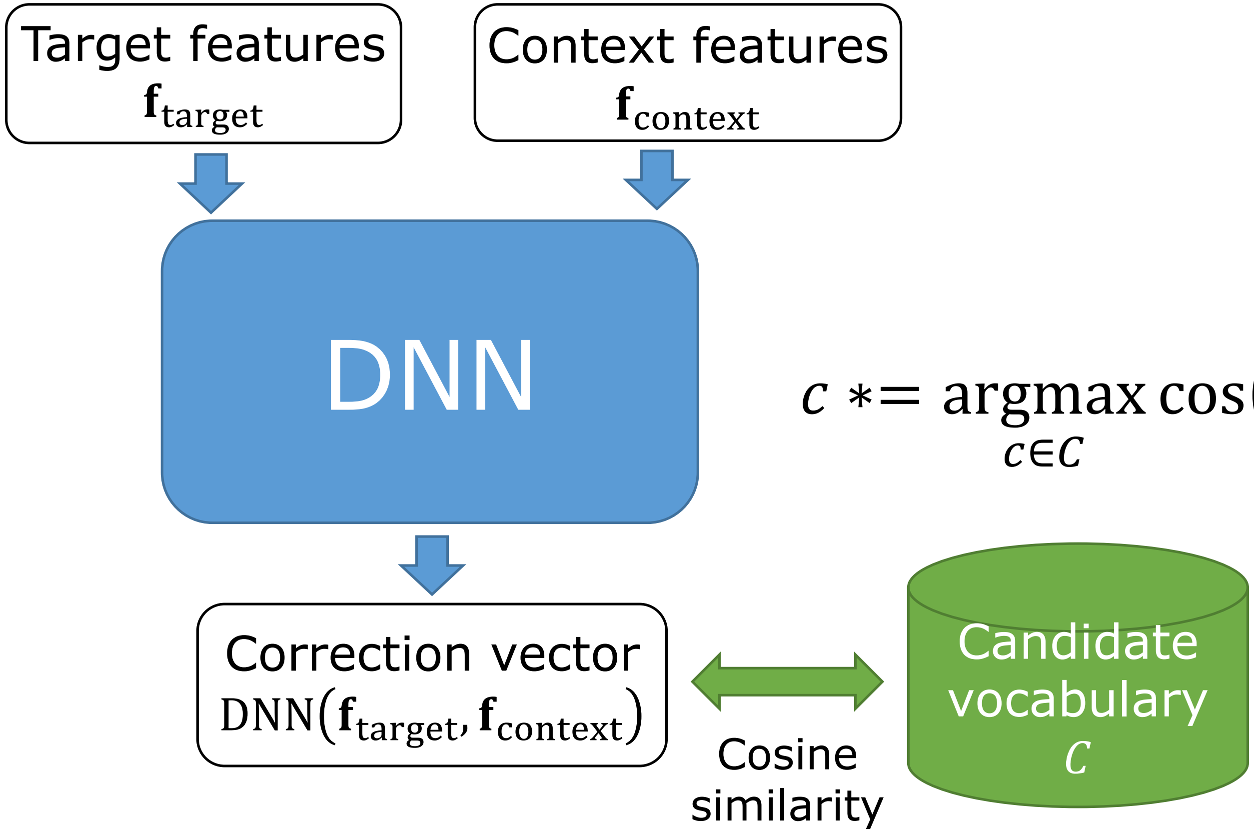
- Example 2

		Correctness	Similarity
Wrong segment	發生這種情況的**情緒**很多		
Ground-truth correction	發生這種情況的**因素**很多	O	?

- **Correctness > similarity**: incorrect sentence can confuse language learners!

6 Correction – Model

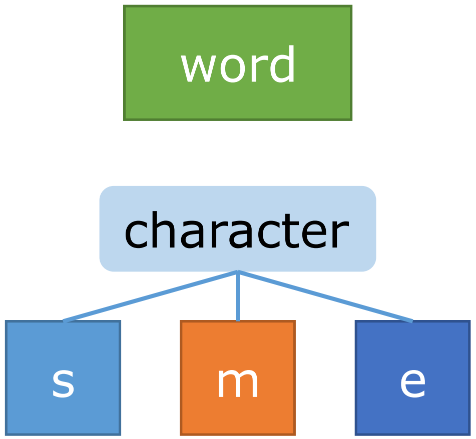
- **Target**: erroneous token that needs correction
  Context: other words in the segment
- Both need to be considered to meet the two criteria



$$c* = \underset{c \in C}{\text{argmax}} \cos\left(\text{DNN}\left(\mathbf{f}_{\text{target}}, \mathbf{f}_{\text{context}}\right), \text{vec}(c) \right)$$

$|C| = 48{,}394$

6 Correction – CWE Features



- $\mathbf{CWE_w}$: Target CWE+P Word Embedding

農産品 = 農産品 + 農 + 産 + 品

*農作品 = 農 + 作 + 品

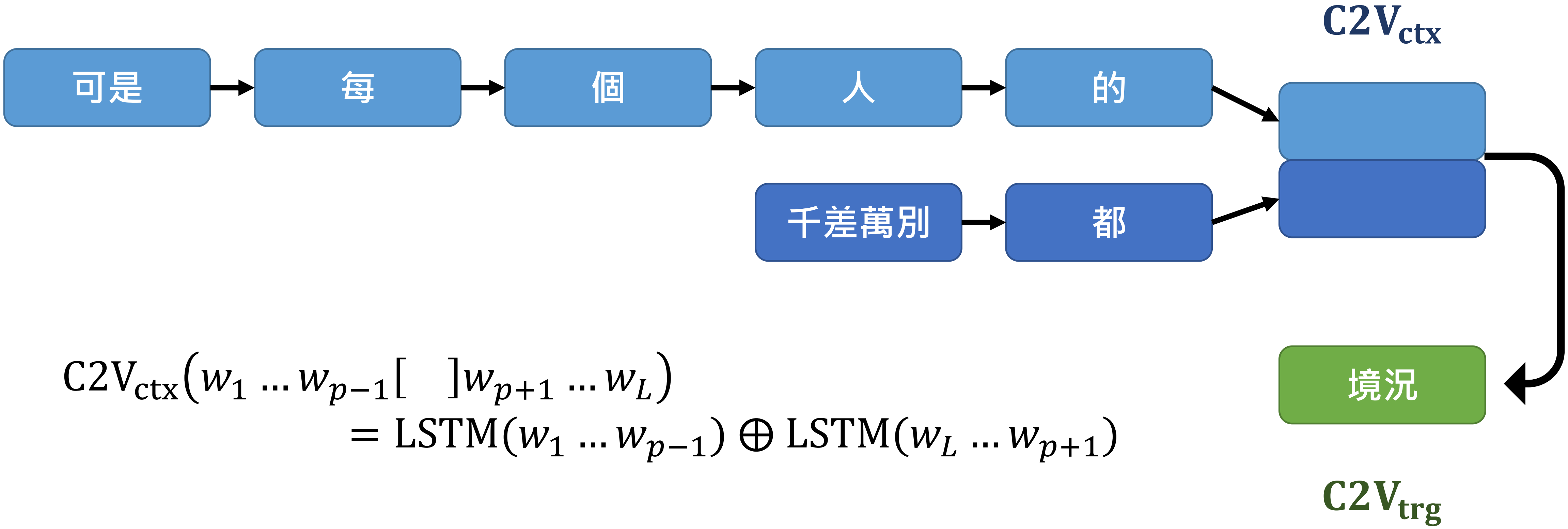
解決 = 解決 + 解 + 決

- $\mathbf{CWE_c}$: Target CWE **Position-insensitive** Character Embedding

*決解 = 決 + 決 + 決 + 解 + 解 + 解

6 Correction – Context2vec Features

- Context: 可是 每 個 人 的 [ ] 都 千差萬別
- Context2vec representation



$$\mathrm{C2V_{ctx}}\left(w_1 \ldots w_{p-1}[\quad]w_{p+1} \ldots w_L\right) = \mathrm{LSTM}(w_1 \ldots w_{p-1}) \oplus \mathrm{LSTM}(w_L \ldots w_{p+1})$$

6 Correction – Context2vec Features

- Context2vec sentence completion

$$c* = \underset{c \in C}{\operatorname{argmax}} \cos\left(\mathrm{C2V_{ctx}}(w_1 \ldots w_{p-1}[\quad]w_{p+1} \ldots w_L), \mathrm{C2V_{trg}}(c) \right)$$

- WUE correction ≠ sentence completion

		Correctness	**Similarity**
Wrong segment	可是 每 個 人 的 **對應** 都 千差萬別		
C2V sentence completion	可是 每 個 人 的 **[境況]** 都 千差萬別	O	X
Ground-truth correction	可是 每 個 人 的 **反應** 都 千差萬別	O	O

6 Correction – POS Features

- **Systematic transitions** of POS tags before & after correction

Original POS	Correction POS	# instances (%)
(unchanged)		722 (68.70%)
VV	NN	27 (2.57%)
NN	VV	21 (2.00%)
P	VV	17 (1.62%)
DEC //的	DEV //地	15 (1.43%)
VV	P	13 (1.24%)

- One-hot encoding of POS → learn different transformation function for different source POS (POS of the erroneous token)

6 Correction – LM Re-ranking

- Correctness criterion not taking priority over similarity criterion
- Can generate segments seriously **violating correctness criterion**

		Correctness	Similarity
Wrong segment	到 山頂 **之間** 路 走 得 不 容易		
Model prediction	到 山頂 **期間** 路 走 得 不 容易	X	O
Ground-truth correction	到 山頂 **的** 路 走 得 不 容易	O	?

- Should be eliminated by a language model (LM)
  - LM probability reflects the level of correctness

6 Correction – LM Re-ranking

- LMs (trained on the Chinese ClueWeb corpus)
  - Traditional N-gram Language Model (N-gram LM)
    - *n* = 5
    - Modified Kneser-Ney smoothing (Heafield et al., 2013)
  - Recurrent Neural Network Language Model (RNNLM)
- Re-ranking: combine ranks with **weighted harmonic mean**

$$r_{\mathrm{com}} = \frac{1}{\frac{\alpha}{r_{\mathrm{LM}}} + \frac{1-\alpha}{r_{\mathrm{DNN}}}}$$

  - α: tuned with validation set
  - r_{com} can be interpreted as rank, **smaller better**

6 Correction – Automatic Evaluation

Target features	Context features	Acc.	MRR	Hit@5	Hit@10	Hit@50
Baselines (No training on the WUE dataset)						
-	**N-gram LM**	0.1659	0.2438	0.3268	0.4029	0.5951
-	**RNNLM**	0.1468	0.2208	0.2847	0.3611	0.5793
-	$C2V_{ctx}$	0.0714	0.1170	0.1575	0.2114	0.3611
Correction Generation Model – Context2vec Features						
$C2V_{trg}$	-	0.2507	0.3030	0.3561	0.3932	0.5024
-	$C2V_{ctx}$	0.1249	0.1746	0.2273	0.2741	0.4010
$C2V_{trg}$	$C2V_{ctx}$	0.3249	0.3891	0.4566	0.4976	0.6185

Ignore target

Target is important!

6 Correction – Automatic Evaluation

Target features	Context features	Acc.	MRR	Hit@5	Hit@10	Hit@50
Correction Generation Model – Context2vec Features						
$C2V_{trg}$	-	0.2507	0.3030	0.3561	0.3932	0.5024
$C2V_{trg}$	$C2V_{ctx}$	0.3249	0.3891	0.4566	0.4976	0.6185
Correction Generation Model – CWE + Other Features						
CWE_{w}		0.2898	0.3545	0.4195	0.4693	0.5971
+ CWE_{c}		0.2946	0.3570	0.4234	0.4722	0.6078
+ $C2V_{trg}$	+ $C2V_{ctx}$	0.3512	0.4250	0.5024	0.5571	0.6800
+ POS		**0.3717**	**0.4378**	**0.5063**	**0.5688**	**0.6956**

Handle OOV target

6 Correction – Automatic Evaluation

Target features	Context features	Acc.	MRR	Hit@5	Hit@10	Hit@50
Correction Generation Model – Context2vec Features						
$\mathbf{C2V_{trg}}$	-	0.2507	0.3030	0.3561	0.3932	0.5024
$\mathbf{C2V_{trg}}$	$\mathbf{C2V_{ctx}}$	0.3249	0.3891	0.4566	0.4976	0.6185
Correction Generation Model – CWE + Other Features						
$\mathbf{CWE_w}$		0.2898	0.3545	0.4195	0.4693	0.5971
+ $\mathbf{CWE_c}$		0.2946	0.3570	0.4234	0.4722	0.6078
+ $\mathbf{C2V_{trg}}$	+ $\mathbf{C2V_{ctx}}$	0.3512	0.4250	0.5024	0.5571	0.6800
+ POS		**0.3717**	**0.4378**	**0.5063**	**0.5688**	**0.6956**

6 Correction – Automatic Evaluation

- DNN + LM Re-ranking

Model	Acc.	MRR	Hit@5	Hit@10	Hit@50	Hit@100
Best DNN	0.3717	0.4378	0.5063	0.5688	0.6956	0.7415
+ N-gram LM ($\alpha = 0.355$)	**0.3727**	**0.4605**	**0.5561**	**0.6439**	**0.8039**	**0.8488**
+ RNNLM ($\alpha = 0.255$)	**0.3727**	0.4527	0.5278	0.6205	0.7808	0.8302

- Example in which LM helps
  - 我從上小學起成績就(*一起,一直)都不理想
  - LM rank: 7 / DNN rank: 1284
  - Ans rank: 19

6 Correction – Human Evaluation

- Correction can be subjective, **alternatives** may exist!

		Correctness	Similarity
Wrong segment	不過 我們 要以 堅定 的 **定心** 與 病 對抗		
Model rank 1	不過 我們 要以 堅定 的 **自信** 與 病 對抗	O	?
Model rank 2	不過 我們 要以 <u>堅定 的 **信念**</u> 與 病 對抗	O	?
Model rank 3	不過 我們 要以 堅定 的 **理智** 與 病 對抗	?	?
Model rank 4	不過 我們 要以 堅定 的 **自信心** 與 病 對抗	O	?
Model rank 5	不過 我們 要以 堅定 的 **毅力** 與 病 對抗	O	?
Ground-truth correction	不過 我們 要以 堅定 的 **決心** 與 病 對抗	O	O

6 Correction – Human Evaluation

- Using single-answer ground-truth can **underestimate** system performance
- Human annotation
  - Ground-truth correction c_0
  - Rank r candidate c_r where $r \leq 5$ and $r < r_{ans}$
    - r_{ans}: rank of c_0 predicted by model
- Annotation instance: a pair of segments (S1), (S0)
  - (S1): candidate correction (ground-truth or system generated)
  - (S0): wrong segment
- Annotation questions (binary)
  - ***is_c***: Is (S1) syntactically and semantically correct?
  - ***is_g***: Is (S1) a correction of (S0)?

6 Correction – Human Evaluation

- Update ranks according to annotation result
  - r: original rank / $\bar{r}$: updated rank

```
r̄ = r
for r′ = 1 to 5
  if is_g(c_r′) and is_c(c_r′)
    r̄ = r′
    break
```

- Use $\bar{r}$ to re-calculate the evaluation metrics

Evaluation	**Acc.**	**MRR**	**Hit@5**	**Hit@10**	**Hit@50**	**Hit@100**
Ground-truth	0.3727	0.4605	0.5561	0.6439	0.8039	0.8488
+ Annotation	0.6829	0.7784	0.9122	0.9171	0.9502	0.9600

6 Correction – Error Analysis

- Performance on most frequent target POS tags

POS (# instances)	Accuracy	MRR	Mean rank
VV (316)	0.67	0.77	26.12
NN (277)	0.64	0.73	73.97
AD (130)	0.65	0.75	96.16
P (62)	0.81	0.88	3.10
VA (45)	0.60	0.76	1.98
DEV (23) //地	1.00	1.00	1.00
PN (21)	0.71	0.80	2.33

6 Correction – Conclusion

- Both **context** and **target** information need to be considered to determine a suitable WUE correction

- **LM re-ranking** further emphasizes **correctness**

- Human evaluation is conducted since there might be alternative corrections.
- In more than 90% of the cases, at least one of the top 5 candidates is an acceptable correction.

7 Conclusion and Future Work

- Conclusion
- Future Work

7 Conclusion and Future Work

- Information used in each stage

<table>
<tr><th>Info.</th><th>Segment Detection</th><th>Token Detection</th><th colspan="2">Correction</th></tr>
<tr><td>Character</td><td>• Single-character</td><td></td><td rowspan="2">• CWE word & char. embedding</td><td></td></tr>
<tr><td>Word</td><td>• N-gram prob.
• CBOW/SG</td><td>• CWIN/ Struct-SG
• N-gram prob.</td><td>• Context2vec
• N-gram LM</td></tr>
<tr><td>POS</td><td></td><td>• POS embedding</td><td colspan="2">• POS one-hot encoding</td></tr>
<tr><td>Dependency</td><td>• Dep. count
• Dep. bigram</td><td>* Evaluation</td><td colspan="2"></td></tr>
</table>

7 Conclusion and Future Work

- Future work
  - Wider context: sentence, paragraph, …
    - Conjunction
      e.g. (*終於, 所以)我只好放棄自己的希望
      e.g. (*還是, 並且)努力要理解媽媽時代的思想和看法
    - Discourse dependent
      e.g. 如果我是(*我, 她)的話 // Why not 你?
    - Meaning changed
      e.g. (*理解, 解決)各種的問題
  - Similar pronunciation
    - E.g. 最深刻的(*影響, 印象)是島上的小學運動會
    - E.g. 就會(*揮服, 恢復)到以前的穩定的經濟情況了

Q&A