




Word Segmentation With CNN

Ativit	Chaninchoduek	6220412019
Potchara	Vinitwattanakoon	6210412003
Nattapol	Hemtanon	6220412011
Peerawat	Khampuang	6110412006



Outline

- The problem and motivations
- Model Pipeline
- Data and preprocessing
- How our modeling techniques works
- Fine Tuning Parameters
- Experiments and results

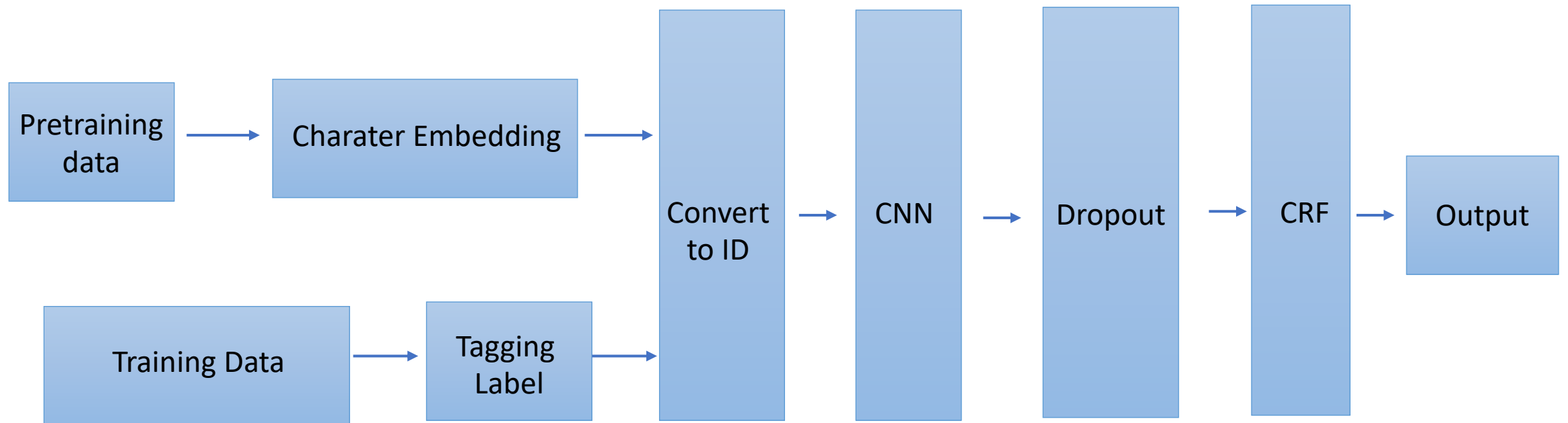
The problem and motivations

- CWS created before have 2 issue
 1. Rely on manually assigned n-gram feature -> Convolution Neural Network
 2. Doesn't use full word information -> Word Embedding

S	S	B	E	B	M	E	S
我	有	一	台	计	算	机	。
(I)	(have)	(a)		(computer)			(.)

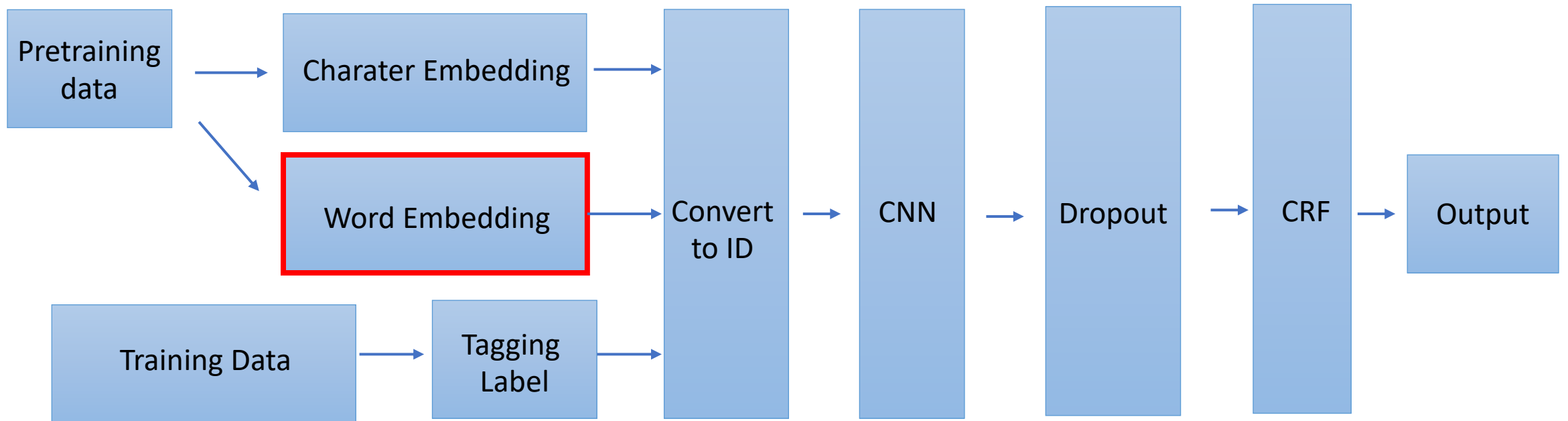
Concept(CWS)

Model : CONV-SEG



Concept(CWS)

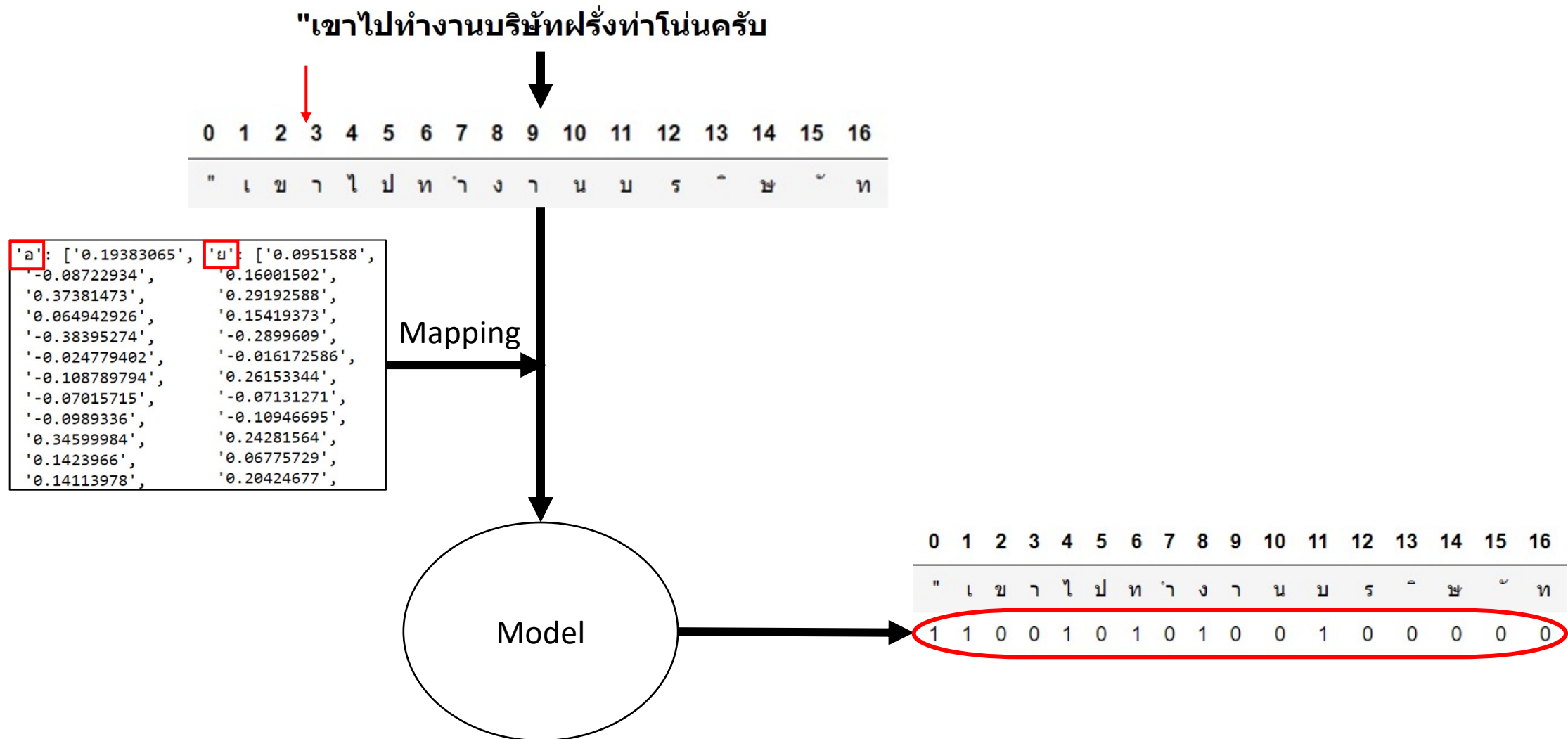
Model : WE-CONV-SEG (+ word embedding)



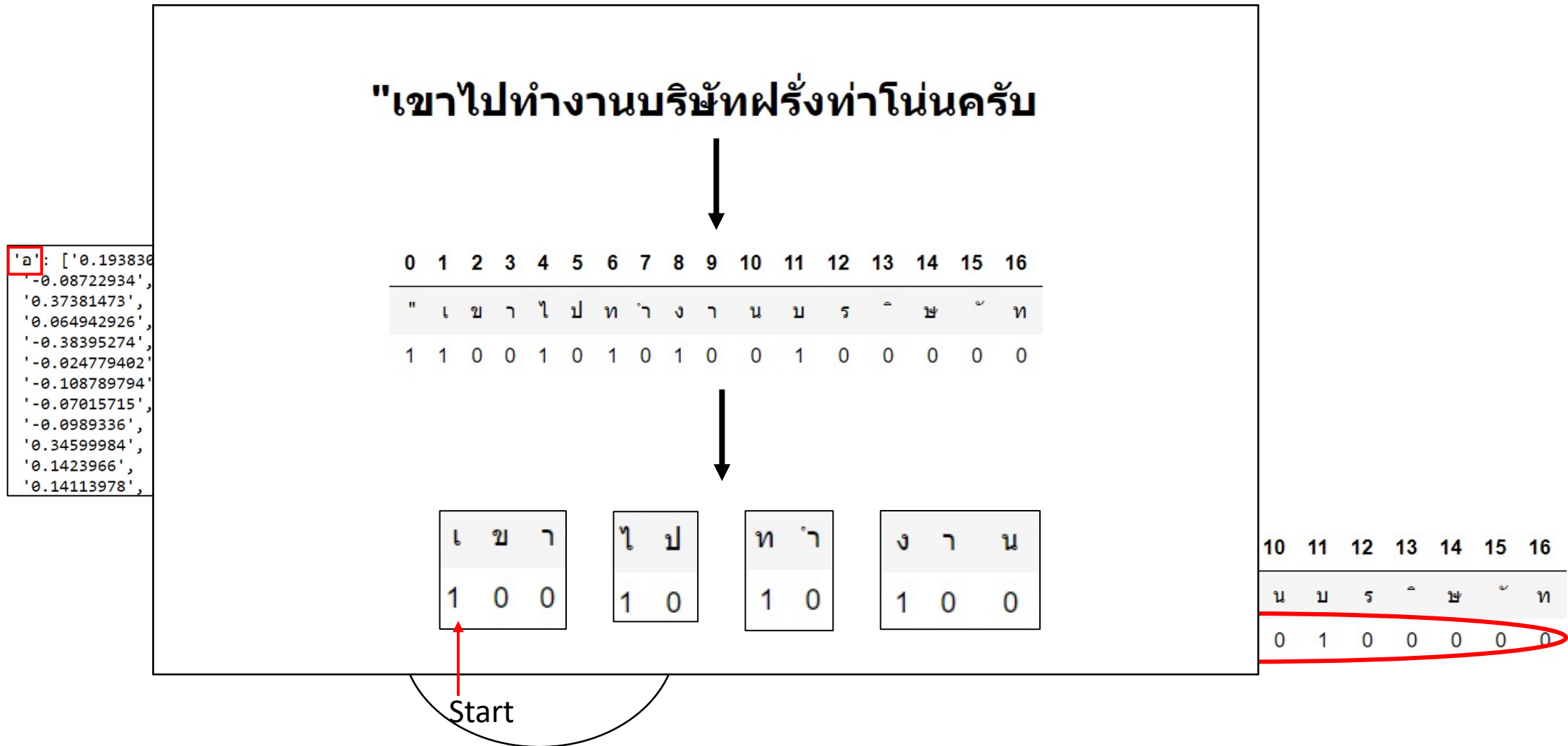
The problem and motivations

- Unlike English and other western languages, most east Asia languages, Thai languages are written without explicit word delimiters
- Thai language is ambiguous
- Don't want to use the N-gram Feature as other paper (too expensive to tag)

Model Pipeline



Model Pipeline



Data and preprocessing

- ใช้ **BEST Corpus** จาก **NECTEC**



File size : 68.3 MB



<http://www.bangkokhealth.com/healthne>
สงสัย|ติด|หวัด|นก| |อีก|คน|ยัง|นำ|ห่วง|
ตาม|ที่|<NE>นางประนอม ทองจันทร์</NE>| |กับ|
<NE>ด.ญ.กาญจนา กรองแก้ว</NE>| |ป่วย|สงส|
หลัง|เข้า|เยี่ยม|ดู|อาการ|ผู้ป่วย|แล้ว| |<NE>
ผล|การ|ดำเนิน|การ| |รวม|ทั้ง|สอบสวน|โรค|
ข่าว| |โดย| |<NE>น.พ.จรัส</NE>|กล่าว|ว่า|
ราย|ของ| |<NE>ด.ช.กิตติพงษ์</NE>|กับ| |
จะ|กลับ|บ้าน|ได้|ใน|ไม่|ช้า|นี้| |แต่|ใน|ราย|
3| |ราย| |ใน|ชั้น|นี้|ถือ|ว่า|เป็น|ผู้ป่วย|อยู่|
ป่วย|มี|อาการ|ปวด|บวม|ปวด|อีก|เสบ| |เนื่อง|
นก|แน่ชัด|หรือ|ไม่| |ต้อง|รอ|ผล|ตรวจ|จาก|ที่

Data and preprocessing

```
s = f.read()
s = re.sub(u'\uffff<AB>|</AB>|<NE>|</NE>', '', s)
s = re.sub(u'<POEM>', '<cut>?poem>', s)
s = re.sub(u'</POEM>', '<cut>', s)
```

http://www.bangkokhealth.com/healthne
สงสัย|ติด|หวัด|นก| |อีก|คน|ยัง|นำ|ห่วง|
ตาม|ที่|<NE>นางประนอม ทองจันทร์</NE>| |กับ
<NE>ด.ญ.กาญจนา กรองแก้ว</NE>| |ป่วย|สงส
หลัง|เข้า|เยี่ยม|ดู|อาการ|ผู้ป่วย|แล้ว| |<NE>
ผล|การ|ดำเนิน|การ| |รวม|ทั้ง|สอบสวน|โรค|
ข่าว| |โดย| |<NE>น.พ.จรัส</NE>|กล่าว|ว่า|
รายชื่อของ| |<NE>ด.ช.กิตติพงษ์</NE>| |กับ|
จะ|กลับ|บ้าน|ได้|ใน|ไม่|ช้า|นี้| |แต่|ใน|รายชื่อ
3| |รายชื่อ| |ใน|ชั้น|นี้|ถือว่า|เป็น|ผู้ป่วย|อยู่|
ป่วย|มี|อาการ|ปอด|บวม|ปอด|อักเสบ| |เนื่อง|
นก|แน่ชัด|หรือไม่| |ต้อง|รอ|ผล|ตรวจ|จาก|ห



พอ|เอ๋ย|ถึง|ชื่อ|นี้| |ก็|เรียก|รอย|ยิ้ม|ขึ้น|บน|ดวง|หน้า|ของ|ทุก|
หน้า
"|หญิง|สาว|ขยับ|จะ|ตอบ| |หาก|แล้ว|ก็|นั่ง|อยู่|ชั่วคราว
นอก|จาก|นั้น| |ต้อง|มี|พื้นที่|เพียงพอ|สำหรับ|รถ|บริการ|ที่|จะ|เข้า
อากาศยาน|ขณะ|ที่|เครื่อง|บิน|จอด|อีก|ด้วย
ผล|ไม้|บาง|ชนิด| |เมื่อ|แก่|สี|ผล|อาจ|เปลี่ยน|จาก|สี|เขียว|เป็น|
อิทธิพล|ของ|ปัจจัย|แวดล้อม|บาง|อย่าง| |เช่น
"|พี่|ก็|พูด|เกิน|ไป| |เขา|ลง|รูป|แม่|มากกว่า|มัง|ครับ|"
ทำ|ที่|ยาม|เล่า|ชาน|อย่าง|ออกรส|ของ|เจ้าของ|บ้าน

Data and preprocessing

```
random.shuffle(lines)
rate = int(len(lines)*0.8)
train_lines = lines[:rate]
test_lines = lines[rate:]
print(len(lines), len(train_lines), len(test_lines))
```

Train	119596
Test	29900
Total	149496

ต้องการ shuffle

- เพื่อแบ่ง train – test
- เพื่อ generalize และไม่ให้โมเดล learn pattern ตามประเภท article

News

WWW.KOMCHADLUEK.NET |
หา|<AB>จนท.</AB>|รถไฟ|จม|<NE>น้ำยม</NE>|ไม่|พบ|สาย|เหนือ
ค้นหา|คน|รถไฟ|เหยื่อ|ดิน|ถล่ม|หัว|รถไฟ|จักร|ใน| |<NE>จ.แพร่</NE>
เหตุ|จาก|น้ำ|เชี่ยว|และ|สูง|ขึ้น|อย่าง|ต่อเนื่อง|ขณะ|ที่|การ|เดิน|ร
รอ|ซ่อมแซม|ราง|จาก|โคลน|ถล่ม|เขต|<NE>เมืองเชียงใหม่</NE>|-
ที่|ภาค|ใต้|ทาง|เชื่อม|<NE>สุราษฎร์</NE>|-|<NE>พังงา</NE>|ถูก
ภาวะ|ฝน|ตก|หนัก|ที่|เกิด|ขึ้น|กับ|พื้นที่|ภาค|เหนือ|จน|ทำให้|มี|น้ำ
ว|กว้าง|สร้าง|ความ|เดือดร้อน|ให้|กับ|ผู้|ที่|อาศัย|อยู่|ใน|ที่|ลุ่ม|ใน
ส่ง|ผล|ให้|ดิน|จาก|ภูเขา|พังทลาย|ลง|มา|และ|ชน|เข้า|กับ|รถไฟ|ช
<NE>กรุงเทพฯ</NE>|เหตุ|เกิด|ใน|พื้นที่|บริเวณ|<NE>บ้านแก่งหลวง</NE>

Novel

๑๖ |
คง|จะ|เป็น|ครั้ง|แรก|ใน|ชีวิต| |ที่|ผม|โกรธ|ใคร|จน|เลือด|วิ่ง|ปรู
ผม|เกือบ|เหวี่ยง|ที่|ทับ|กระดาศ|ใส่|หัว|<NE>ไอ้เจ้าชโนเตอร์</NE>
เสียง| |ขณะ|ที่|มัน|หัน|หลัง|เดิน|ออก|ไป|นอก|ห้อง|ทำงาน|ของ|ผ
ผม|ชัก|ได้|กลืน|แปลกๆ| |อยู่|เหมือน|กัน| |ก่อน|จะ|ไป|นอก|ครั้ง
ครั้ง|ใหญ่|เกิด|ขึ้น|ใน|บริษัท|ข้าม|ชาติ|ที่|ผม|ทำงาน|อยู่| |แต่|ผม
จาก|ความ|เปลี่ยนแปลง|นั้น|ด้วย| |เพราะ|ผม|มั่นใจ|ตลอด|มา|ว่า|
|มี|ตำแหน่ง|เป็น|ผู้|บริหาร|ระดับ|อาวุโส|เป็น|คน|มี| |" |เส้น|

Data and preprocessing

```
for i in range(len(char_list)):
    for j in range(len(char_list[i])):
        if char_list[i][j] in ['}', '~', '^', '$']:
            char_list[i][j] = '<unk>'
```

```
model = Word2Vec(sentences=char_list, size=50, window=10, sg=1)
```

Number of unique char 172

```
[('ศ', 5),
 ('<unk>', 9),
 ('จ', 10),
 ('ข', 22),
 ('#', 22),
 (''', 28),
 (''', 28),
 ('Q', 30),
```

"คุณสวัสดีมาที่นี่ คุณแม่หลับแล้วหรือ?" หม่อมเลี้ยงถามถึงมารดาของท่านเสีย

Save json file

```
with open('chr2vec_w10_v50_inc_AB_NE.json', 'w', encoding="utf-8") as f:
    json.dump(ce, f)
```

Data and preprocessing

```
model = Word2Vec(sentences=char_list, size=50, window=10, sg=1)
```

'a': ['0.19383065',	'b': ['0.0951588',	'c': ['-0.029462913',
'-0.08722934',	'0.16001502',	'0.10380713',
'0.37381473',	'0.29192588',	'0.24058126',
'0.064942926',	'0.15419373',	'0.026909046',
'-0.38395274',	'-0.2899609',	'-0.17584643',
'-0.024779402',	'-0.016172586',	'0.08276293',
'-0.108789794',	'0.26153344',	'-0.07420771',
'-0.07015715',	'-0.07131271',	'-0.23014803',
'-0.0989336',	'-0.10946695',	'-0.14072508',
'0.34599984',	'0.24281564',	'0.11453628',
'0.1423966',	'0.06775729',	'0.13099736',
'0.14113978',	'0.20424677',	'0.30809203',

size = 50

Data and preprocessing

Step 1

```
def x_y_transformer(self, text, sep='|'):
    inputs_value = list()
    outputs_value = list()
    for word in text.split(sep):
        if len(word) == 0:
            continue
        outputs_value = outputs_value + [1] + [0]*(len(word) - 1)
        for char in word:
            if char == '\n':
                inputs_value.append(self.look_up_dict['<pad>'])
            elif char not in self.look_up_dict.keys():
                inputs_value.append(self.look_up_dict['<unk>'])
            else:
                inputs_value.append(self.look_up_dict[char])
```

"เขาไปทำงานบริษัทฝรั่งทำโน่นครับ"

| " | เขา | ไป | ทำ | งาน | บริษัท |

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Char	"	เ	ข	า	ไ	ป	ท	ำ	ง	า	น	บ	ร	ั	ษ	ั	ท
Encode	1	1	0	0	1	0	1	0	1	0	0	1	0	0	0	0	0

Data and preprocessing

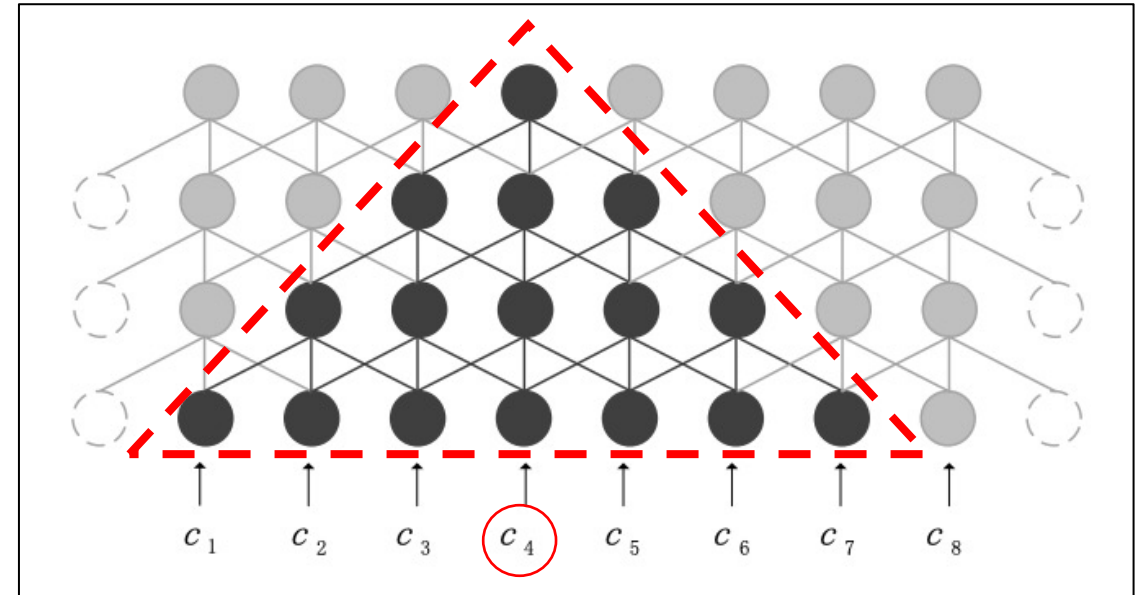
Step 2

ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```
char_size = len(outputs_value)
left_over = char_size%self.sentence_size
n_chunk = char_size//self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1+(left_over>0)):
    if i == 1:
        inputs_before = [self.look_up_dict['<pad>']]*self.overlap
        outputs_before = [0]*self.overlap
    else:
        inputs_before = inputs_value[(i-1)*self.sentence_size-self.overlap:(i-1)*self.sentence_size]
        inputs_before = inputs_before + [self.look_up_dict['<pad>']]*(self.overlap-len(inputs_before))
        outputs_before = outputs_value[(i-1)*self.sentence_size-self.overlap:(i-1)*self.sentence_size]
        outputs_before = outputs_before + [0]*(self.overlap-len(outputs_before))

    if i == n_chunk+1:
        inputs_after = [self.look_up_dict['<pad>']]*(self.sentence_size-left_over+self.overlap)
        outputs_after = [0]*(self.sentence_size-left_over+self.overlap)
    else:
        inputs_after = inputs_value[i*self.sentence_size:i*self.sentence_size+self.overlap]
        inputs_after = inputs_after + [self.look_up_dict['<pad>']]*(self.overlap-len(inputs_after))
        outputs_after = outputs_value[i*self.sentence_size:i*self.sentence_size+self.overlap]
        outputs_after = outputs_after + [0]*(self.overlap-len(outputs_after))

    inputs.append(inputs_before
                  + inputs_value[(i-1)*self.sentence_size:i*self.sentence_size]
                  + inputs_after
                  )
    outputs.append(outputs_before
                  + outputs_value[(i-1)*self.sentence_size:i*self.sentence_size]
                  + outputs_after
                  )
return inputs, outputs
```



ผู้เป็นยายบนอุบอิบ

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Char	<pad>	<pad>	พ	.	"	เ	ื	น	ย	า	ย	บ	.	
Encode	0	0	1	0	0	1	0	0	0	1	0	0	1	0

sentence_size=10, overlap=2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Char	า	ย	บ	.	น	อ	.	บ	อ	.	บ	<pad>	<pad>	<pad>
Encode	0	0	1	0	0	1	0	0	0	0	0	0	0	0

Data and preprocessing

Step 2

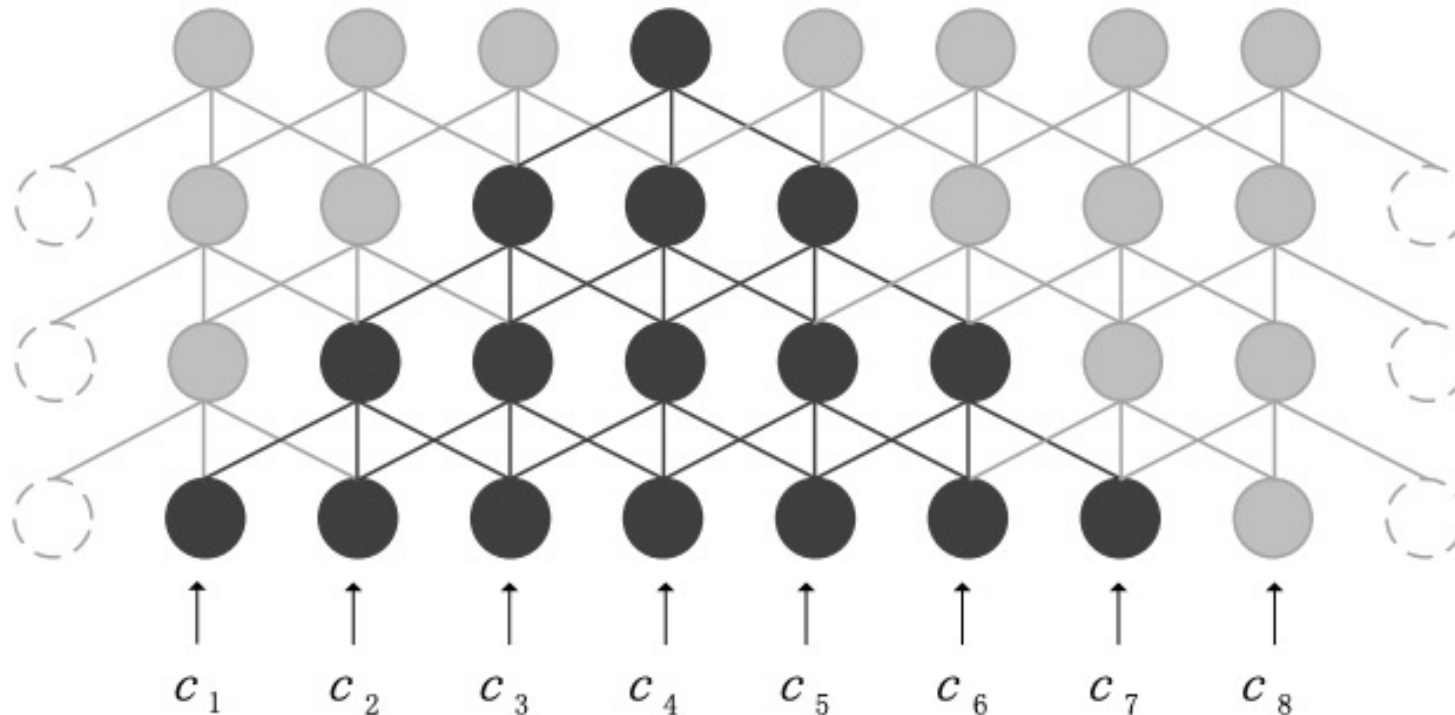
ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```
char_size = len(outputs_value)
left_over = char_size % self.sentence_size
n_chunk = char_size // self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1):
    if i == 1:
        inputs_before = [self.char_size]
        outputs_before = [0]
    else:
        inputs_before = inputs[-1]
        inputs_after = inputs[-1]
        outputs_before = outputs[-1]
        outputs_after = outputs[-1]

    if i == n_chunk+1:
        inputs_after = [self.char_size]
        outputs_after = [0]
    else:
        inputs_after = inputs[-1]
        inputs_after = inputs[-1]
        outputs_after = outputs[-1]
        outputs_after = outputs[-1]

    inputs.append(inputs_before + inputs_after)
    outputs.append(outputs_before + outputs_after)

return inputs, outputs
```



อุปธิ

sentence_size=10, overlap=2

Char	า	ย	บ	น	อ	บ	อ	ิ	บ	<pad>	<pad>	<pad>
Endcode	0	0	1	0	0	1	0	0	0	0	0	0

Data and preprocessing

Step 2

ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```

char_size = len(outputs_value)
left_over = char_size%self.sentence_size
n_chunk = char_size//self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1):
    if i == 1:
        inputs_before = [self.char_size-1]
        outputs_before = [0]
    else:
        inputs_before = inputs[-1]
        inputs_after = inputs[-1]
        outputs_before = outputs[-1]
        outputs_after = outputs[-1]

    if i == n_chunk+1:
        inputs_after = [self.char_size-1]
        outputs_after = [0]
    else:
        inputs_after = inputs[-1]
        inputs_after = inputs[-1]
        outputs_after = outputs[-1]
        outputs_after = outputs[-1]

    inputs.append(inputs_before + inputs_after)
    outputs.append(outputs_before + outputs_after)

return inputs, outputs
        
```

c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8

อุปอิบ

Char	า	ย	บ	น	อ	บ	อ	ิ	บ	<pad>	<pad>	<pad>
Endcode	0	0	1	0	0	1	0	0	0	0	0	0

17

Data and preprocessing

Step 2

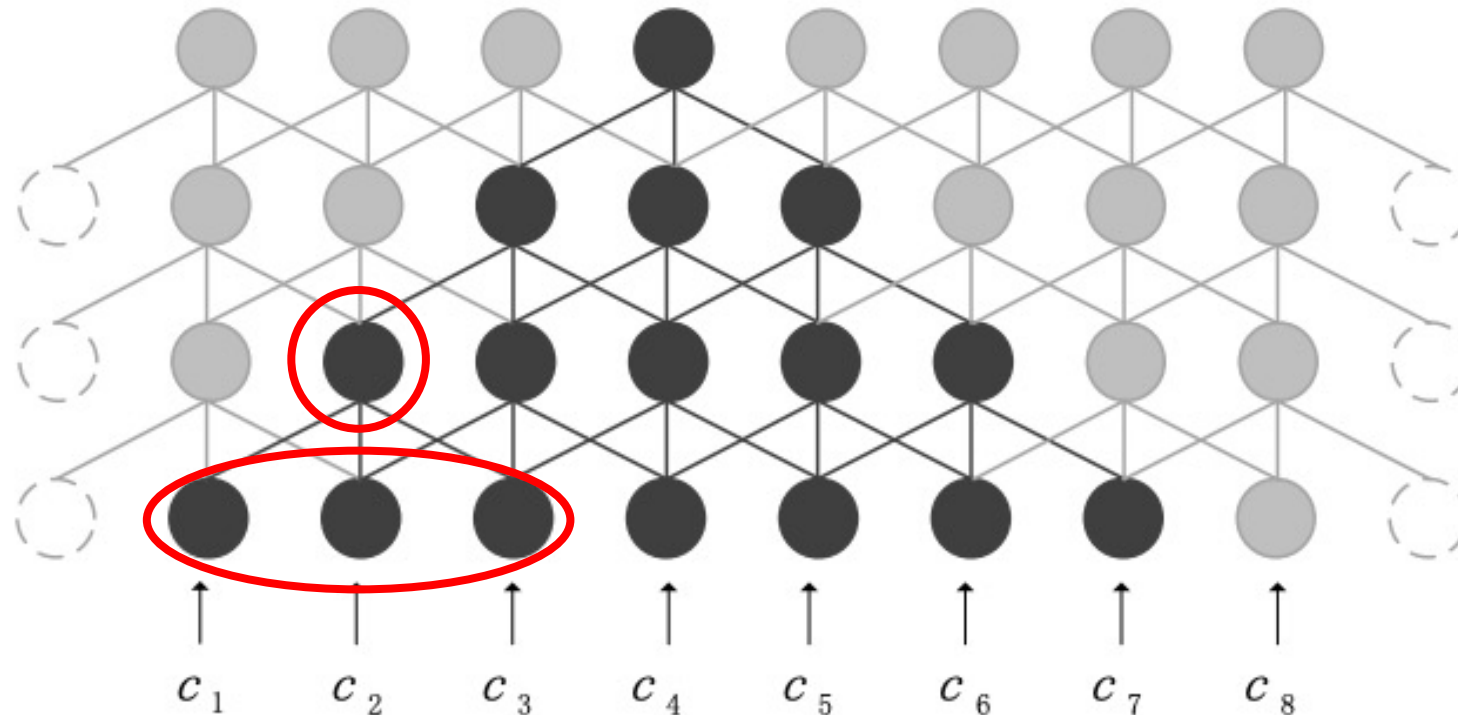
ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```
char_size = len(outputs_value)
left_over = char_size % self.sentence_size
n_chunk = char_size // self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1):
    if i == 1:
        inputs_before = [self.char_size]
        outputs_before = [0]
    else:
        inputs_before = inputs[-1]
        inputs_after = inputs[-1]
        outputs_before = outputs[-1]
        outputs_after = outputs[-1]

    if i == n_chunk+1:
        inputs_after = [self.char_size]
        outputs_after = [0]
    else:
        inputs_after = inputs[-1]
        inputs_after = inputs[-1]
        outputs_after = outputs[-1]
        outputs_after = outputs[-1]

    inputs.append(inputs_before + inputs_after)
    outputs.append(outputs_before + outputs_after)

return inputs, outputs
```



| อับธิบ

sentence_size=10, overlap=2

Char	า	ย	บ	น	อ	บ	อ	ิ	บ	<pad>	<pad>	<pad>
Endcode	0	0	1	0	0	1	0	0	0	0	0	0

Data and preprocessing

Step 2

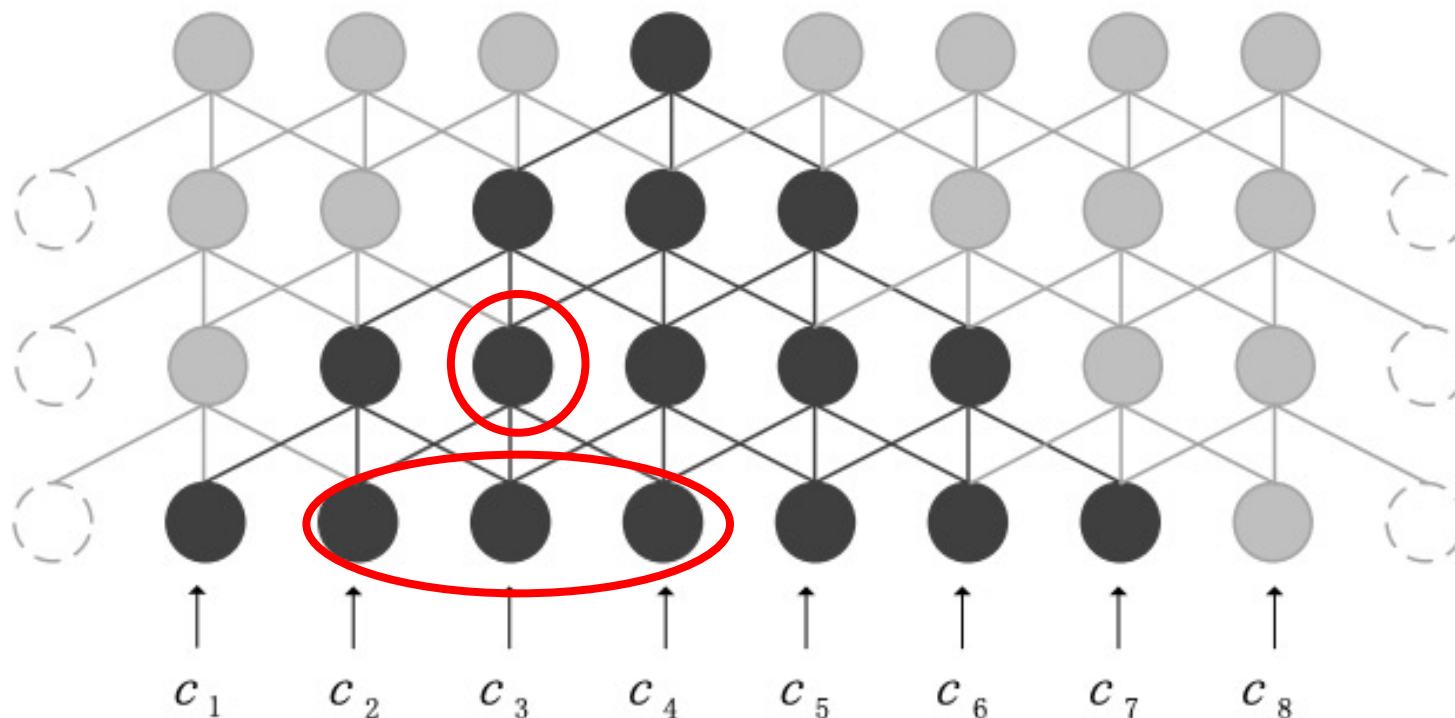
ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```
char_size = len(outputs_value)
left_over = char_size % self.sentence_size
n_chunk = char_size // self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1):
    if i == 1:
        inputs_before = [self.char_size]
        outputs_before = [0]
    else:
        inputs_before = inputs[-1]
        inputs_after = inputs[-1]
        outputs_before = outputs[-1]
        outputs_after = outputs[-1]

    if i == n_chunk+1:
        inputs_after = [self.char_size]
        outputs_after = [0]
    else:
        inputs_after = inputs[-1]
        inputs_after = inputs[-1]
        outputs_after = outputs[-1]
        outputs_after = outputs[-1]

    inputs.append(inputs_before + inputs_after)
    outputs.append(outputs_before + outputs_after)

return inputs, outputs
```



| อับธิบ

sentence_size=10, overlap=2

Char	า	ย	บ	น	อ	บ	อ	ิ	บ	<pad>	<pad>	<pad>
Endcode	0	0	1	0	0	1	0	0	0	0	0	0

Data and preprocessing

Step 2

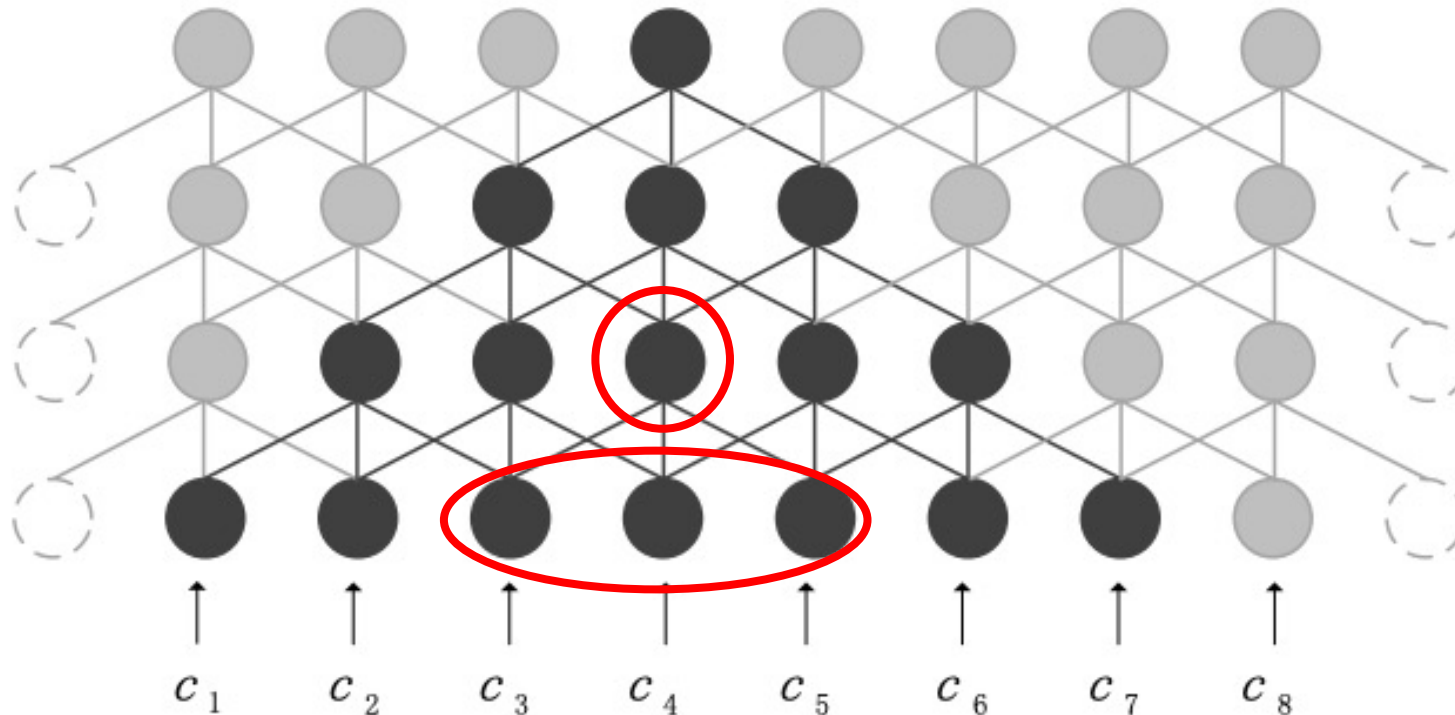
ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```
char_size = len(outputs_value)
left_over = char_size % self.sentence_size
n_chunk = char_size // self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1):
    if i == 1:
        inputs_before = [self.char_size]
        outputs_before = [0]
    else:
        inputs_before = inputs[-1]
        inputs_after = inputs[-1]
        outputs_before = outputs[-1]
        outputs_after = outputs[-1]

    if i == n_chunk+1:
        inputs_after = [self.char_size]
        outputs_after = [0]
    else:
        inputs_after = inputs[-1]
        inputs_after = inputs[-1]
        outputs_after = outputs[-1]
        outputs_after = outputs[-1]

    inputs.append(inputs_before + inputs_v
                + inputs_a
                )
    outputs.append(outputs_b
                + outputs_s
                + outputs_a
                )

return inputs, outputs
```



อุปธิบ

sentence_size=10, overlap=2

Char	า	ย	บ	น	อ	บ	อ	ิ	บ	<pad>	<pad>	<pad>
Endcode	0	0	1	0	0	1	0	0	0	0	0	0

Data and preprocessing

Step 2

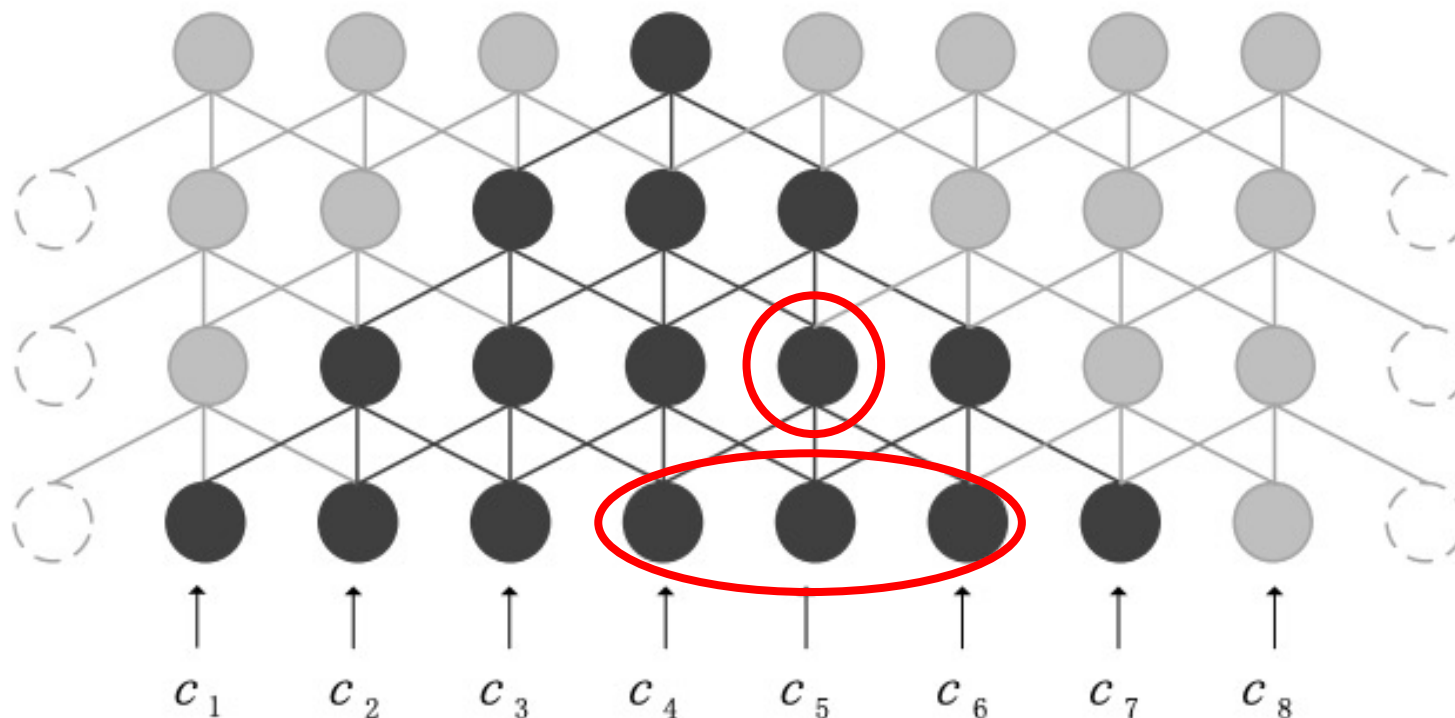
ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```
char_size = len(outputs_value)
left_over = char_size % self.sentence_size
n_chunk = char_size // self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1):
    if i == 1:
        inputs_before = [self.char_size - 1]
        outputs_before = [0]
    else:
        inputs_before = inputs[-1]
        inputs_after = inputs[-1]
        outputs_before = outputs[-1]
        outputs_after = outputs[-1]

    if i == n_chunk+1:
        inputs_after = [self.char_size - 1]
        outputs_after = [0]
    else:
        inputs_after = inputs[-1]
        inputs_after = inputs[-1]
        outputs_after = outputs[-1]
        outputs_after = outputs[-1]

    inputs.append(inputs_before + inputs_after)
    outputs.append(outputs_before + outputs_after)

return inputs, outputs
```



| อธิบาย

sentence_size=10, overlap=2

Char	า	ย	บ	น	อ	บ	อ	ิ	บ	<pad>	<pad>	<pad>
Endcode	0	0	1	0	0	1	0	0	0	0	0	0

Data and preprocessing

Step 2

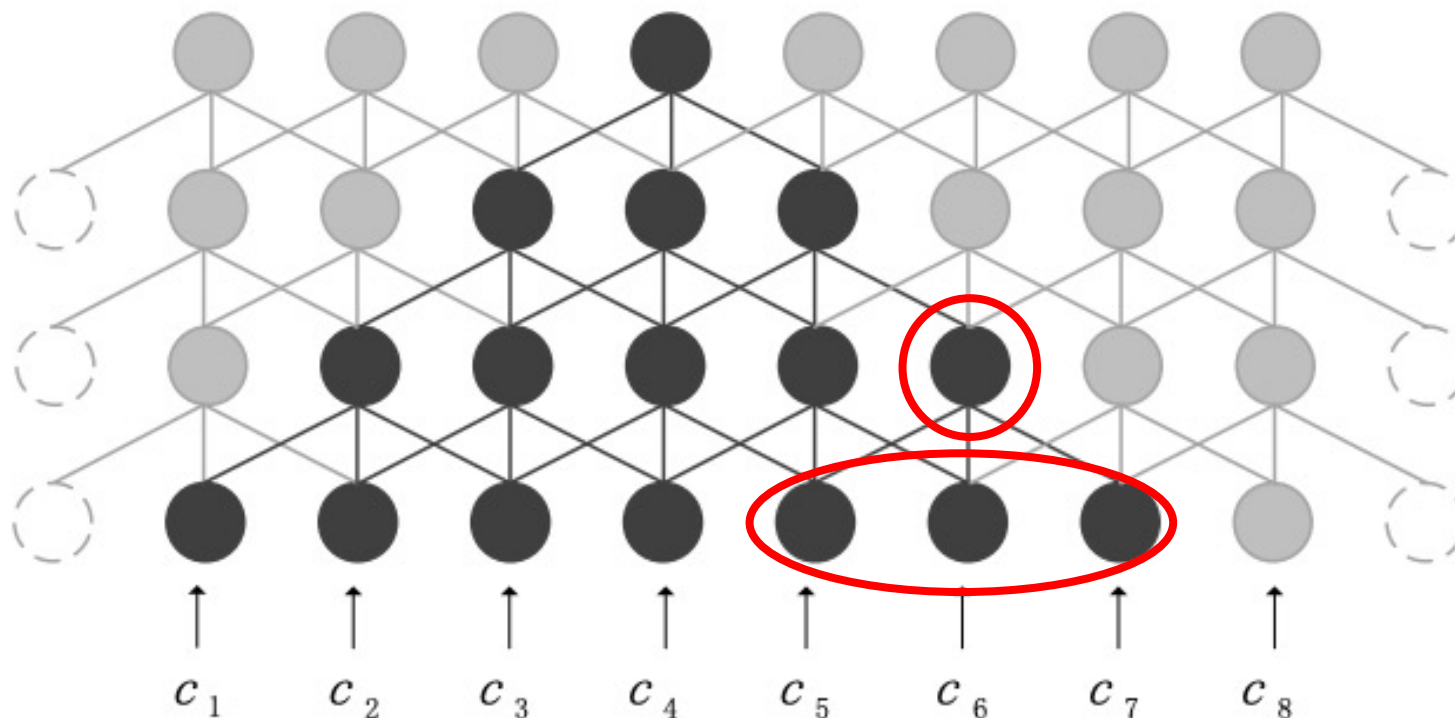
ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```
char_size = len(outputs_value)
left_over = char_size % self.sentence_size
n_chunk = char_size // self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1):
    if i == 1:
        inputs_before = [self.char_size]
        outputs_before = [0]
    else:
        inputs_before = inputs[-1]
        inputs_after = inputs[-1]
        outputs_before = outputs[-1]
        outputs_after = outputs[-1]

    if i == n_chunk+1:
        inputs_after = [self.char_size]
        outputs_after = [0]
    else:
        inputs_after = inputs[-1]
        inputs_after = inputs[-1]
        outputs_after = outputs[-1]
        outputs_after = outputs[-1]

    inputs.append(inputs_before + inputs_after)
    outputs.append(outputs_before + outputs_after)

return inputs, outputs
```



| อับธิบ

sentence_size=10, overlap=2

Char	า	ย	บ	ั	น	อ	,	บ	อ	ั	บ	<pad>	<pad>	<pad>
Endcode	0	0	1	0	0	1	0	0	0	0	0	0	0	0

Data and preprocessing

Step 2

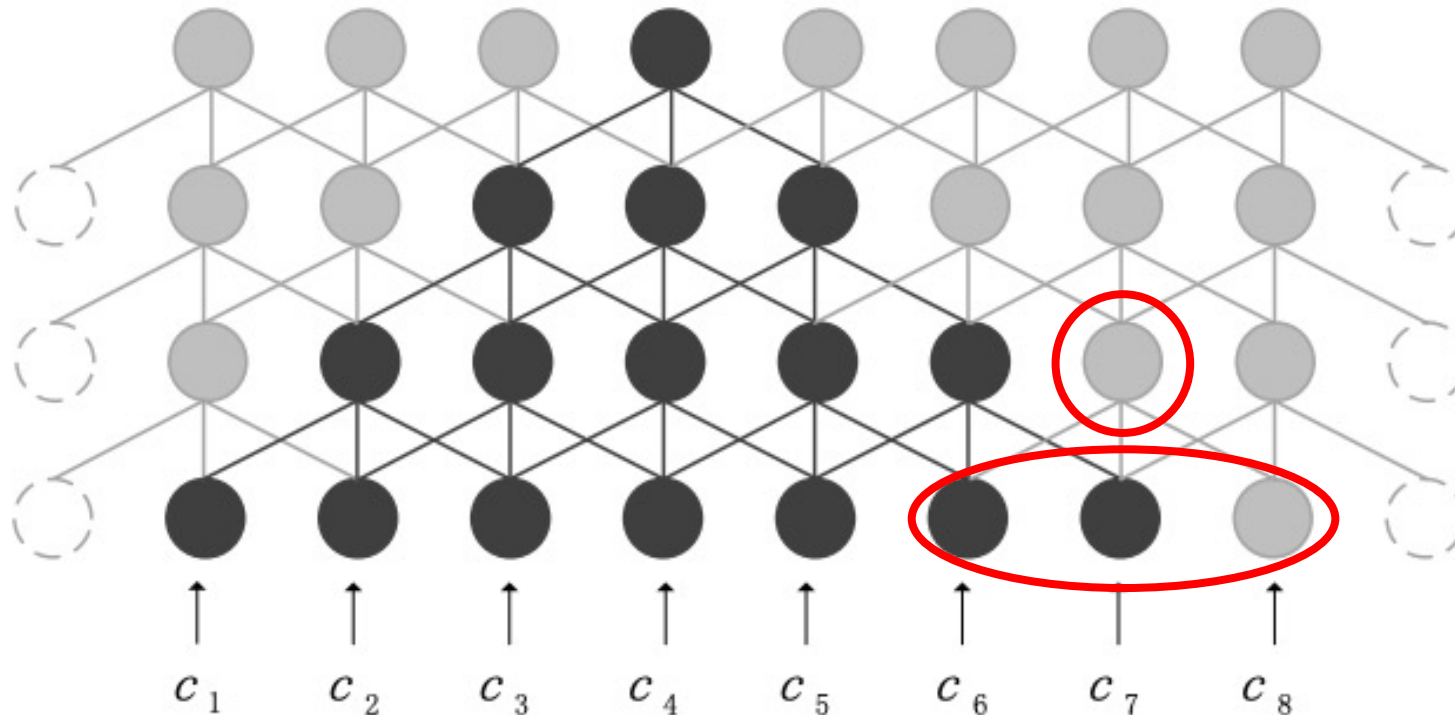
ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```
char_size = len(outputs_value)
left_over = char_size % self.sentence_size
n_chunk = char_size // self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1):
    if i == 1:
        inputs_before = [self.char_size]
        outputs_before = [0]
    else:
        inputs_before = inputs[-1]
        inputs_after = inputs[-1]
        outputs_before = outputs[-1]
        outputs_after = outputs[-1]

    if i == n_chunk+1:
        inputs_after = [self.char_size]
        outputs_after = [0]
    else:
        inputs_after = inputs[-1]
        inputs_after = inputs[-1]
        outputs_after = outputs[-1]
        outputs_after = outputs[-1]

    inputs.append(inputs_before + inputs_after)
    outputs.append(outputs_before + outputs_after)

return inputs, outputs
```



| อับธิบ

sentence_size=10, overlap=2

Char	า	ย	บ	น	อ	บ	อ	ิ	บ	<pad>	<pad>	<pad>
Endcode	0	0	1	0	0	1	0	0	0	0	0	0

Data and preprocessing

Step 2

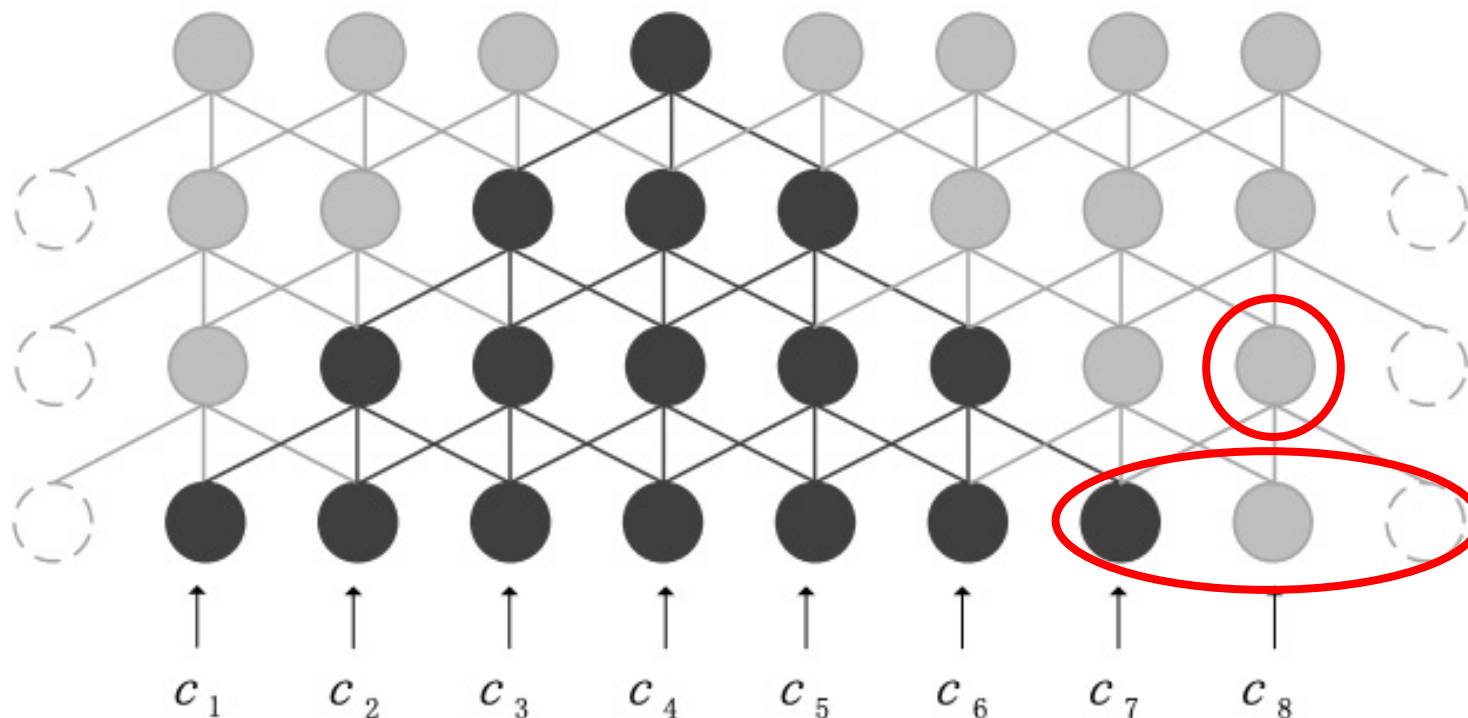
ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```
char_size = len(outputs_value)
left_over = char_size % self.sentence_size
n_chunk = char_size // self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1):
    if i == 1:
        inputs_before = [self.char_size]
        outputs_before = [0]
    else:
        inputs_before = inputs[-1]
        inputs_after = inputs[-1]
        outputs_before = outputs[-1]
        outputs_after = outputs[-1]

    if i == n_chunk+1:
        inputs_after = [self.char_size]
        outputs_after = [0]
    else:
        inputs_after = inputs[-1]
        inputs_after = inputs[-1]
        outputs_after = outputs[-1]
        outputs_after = outputs[-1]

    inputs.append(inputs_before + inputs_after)
    outputs.append(outputs_before + outputs_after)

return inputs, outputs
```



อุปธิบ

sentence_size=10, overlap=2

Char	า	ย	บ	น	อ	บ	อ	ิ	บ	<pad>	<pad>	<pad>
Endcode	0	0	1	0	0	1	0	0	0	0	0	0

Data and preprocessing

Step 2

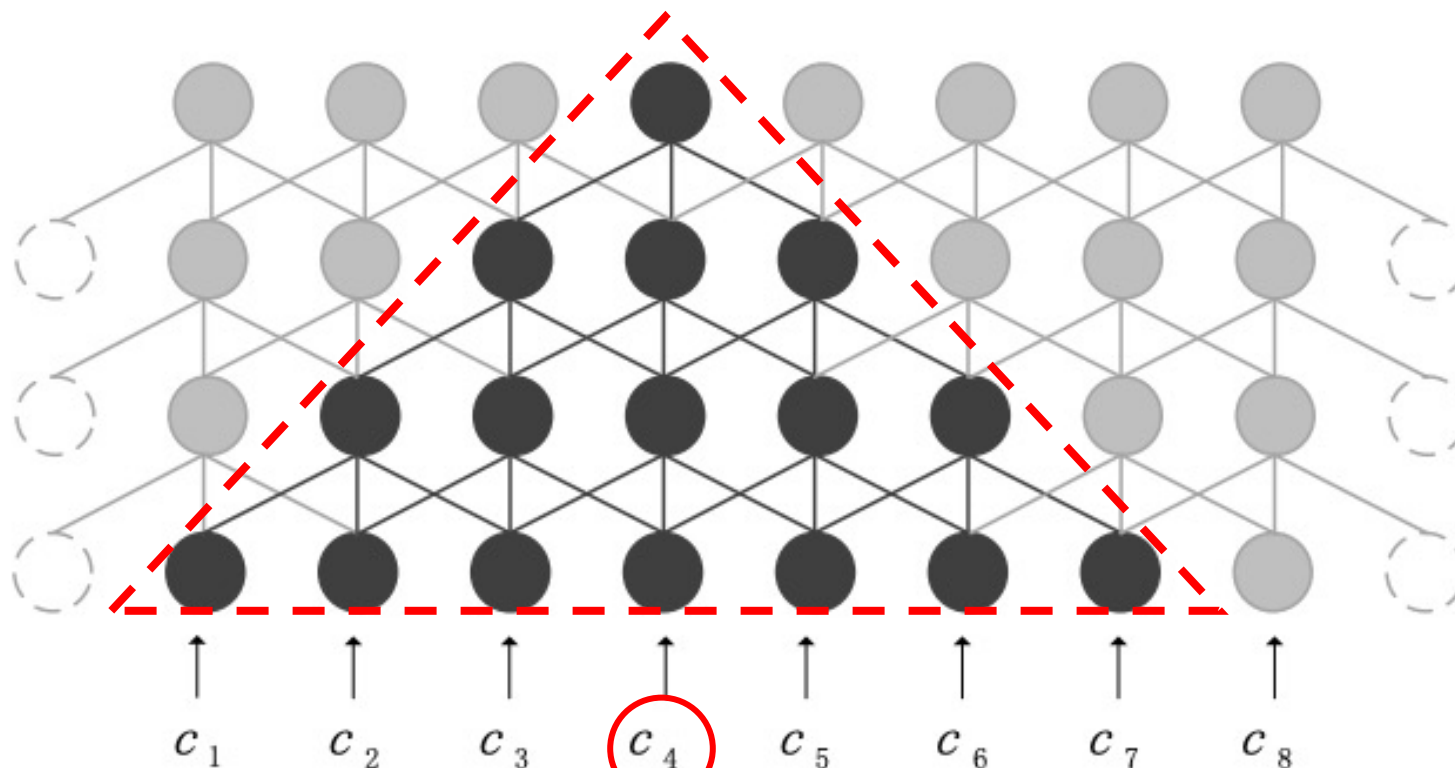
ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```
char_size = len(outputs_value)
left_over = char_size % self.sentence_size
n_chunk = char_size // self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1):
    if i == 1:
        inputs_before = [self.char_size]
        outputs_before = [0]
    else:
        inputs_before = inputs[-1]
        inputs_after = inputs[-1]
        outputs_before = outputs[-1]
        outputs_after = outputs[-1]

    if i == n_chunk+1:
        inputs_after = [self.char_size]
        outputs_after = [0]
    else:
        inputs_after = inputs[-1]
        inputs_after = inputs[-1]
        outputs_after = outputs[-1]
        outputs_after = outputs[-1]

    inputs.append(inputs_before + inputs_after)
    outputs.append(outputs_before + outputs_after)

return inputs, outputs
```



อุปธิ

sentence_size=10, overlap=2

Char	า	ย	บ	ั	น	อ	บ	อ	ั	บ	<pad>	<pad>	<pad>
Endcode	0	0	1	0	0	1	0	0	0	0	0	0	0

Data and preprocessing

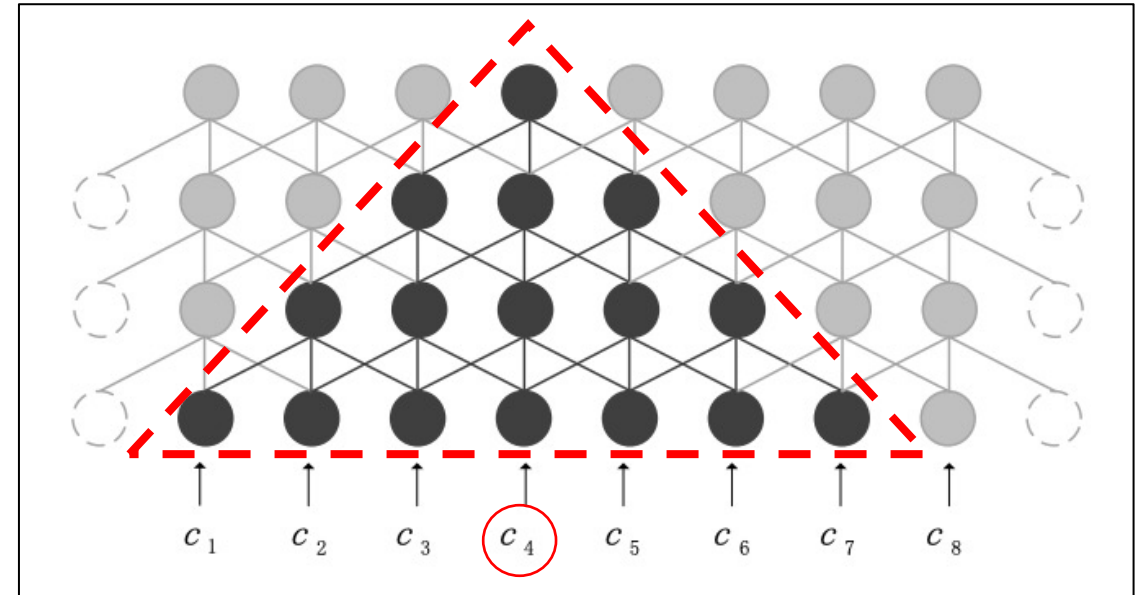
Step 2

ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

```
char_size = len(outputs_value)
left_over = char_size%self.sentence_size
n_chunk = char_size//self.sentence_size
inputs = []
outputs = []
for i in range(1, n_chunk+1+(left_over>0)):
    if i == 1:
        inputs_before = [self.look_up_dict['<pad>']]*self.overlap
        outputs_before = [0]*self.overlap
    else:
        inputs_before = inputs_value[(i-1)*self.sentence_size-self.overlap:(i-1)*self.sentence_size]
        inputs_before = inputs_before + [self.look_up_dict['<pad>']]*(self.overlap-len(inputs_before))
        outputs_before = outputs_value[(i-1)*self.sentence_size-self.overlap:(i-1)*self.sentence_size]
        outputs_before = outputs_before + [0]*(self.overlap-len(outputs_before))

    if i == n_chunk+1:
        inputs_after = [self.look_up_dict['<pad>']]*(self.sentence_size-left_over+self.overlap)
        outputs_after = [0]*(self.sentence_size-left_over+self.overlap)
    else:
        inputs_after = inputs_value[i*self.sentence_size:i*self.sentence_size+self.overlap]
        inputs_after = inputs_after + [self.look_up_dict['<pad>']]*(self.overlap-len(inputs_after))
        outputs_after = outputs_value[i*self.sentence_size:i*self.sentence_size+self.overlap]
        outputs_after = outputs_after + [0]*(self.overlap-len(outputs_after))

    inputs.append(inputs_before
                  + inputs_value[(i-1)*self.sentence_size:i*self.sentence_size]
                  + inputs_after
                  )
    outputs.append(outputs_before
                  + outputs_value[(i-1)*self.sentence_size:i*self.sentence_size]
                  + outputs_after
                  )
return inputs, outputs
```



ผู้เป็นยาย|บ่น|อุบอิบ

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Char	<pad>	<pad>	พ	.	"	เ	ป	"	น	ย	า	ย	บ	
Encode	0	0	1	0	0	1	0	0	0	1	0	0	1	0

sentence_size=10, overlap=2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Char	า	ย	บ	.	น	อ	.	บ	อ	"	บ	<pad>	<pad>	<pad>
Encode	0	0	1	0	0	1	0	0	0	0	0	0	0	0

Data and preprocessing

Step 2

ตัดตามขนาดของ sentence size
และเพิ่ม overlap ตามจำนวน layer

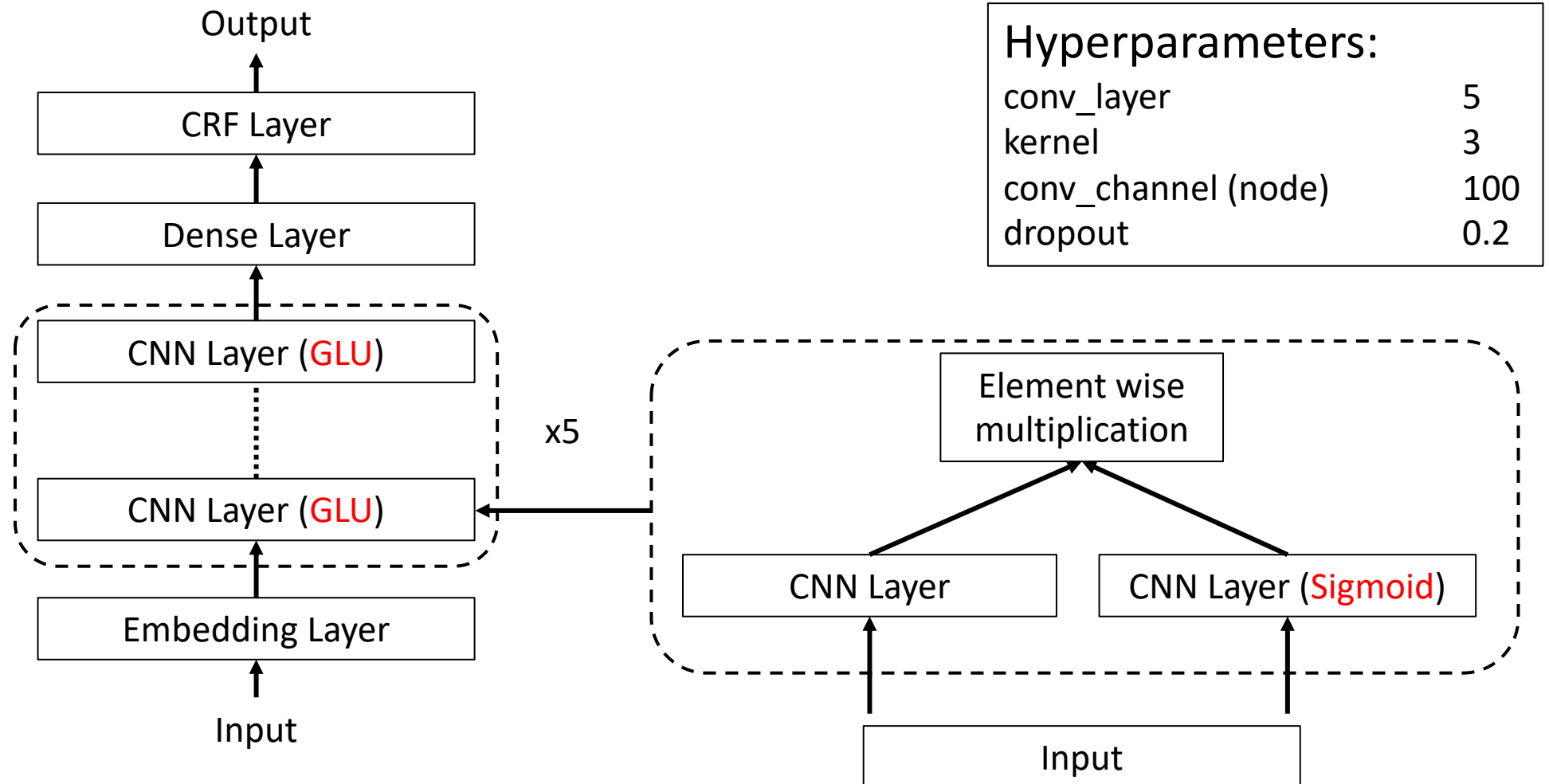
ผู้|เป็น|ยาย|บ่น|อุบอิบ

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Char	<pad>	<pad>	ผ	ู	๋	เ	ป	ั	น	ย	า	ย	บ	.
Endcode	0	0	1	0	0	1	0	0	0	1	0	0	1	0

sentence_size=10, overlap=2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Char	า	ย	บ	.	น	อ	,	บ	อ	ั	บ	<pad>	<pad>	<pad>
Endcode	0	0	1	0	0	1	0	0	0	0	0	0	0	0

How our modeling techniques works



How our modeling techniques works

```
def create_model_crf(self, conv_layer=5, conv_chanel=50, kernel=3, dropout_rate=0.2, glu = False):  
    x = layers.Input(shape=(self.ndim))  
    y = layers.Embedding(self.chr_size,  
                        self.chr_vec_size,  
                        weights=[self.embedding_matrix],  
                        trainable=False)(x)  
    y = layers.Dropout(dropout_rate)(y)  
  
    '''create convolutional layers'''  
    if glu:  
        for i in range(conv_layer):  
            y1 = layers.Conv1D(conv_chanel, kernel, padding='same')(y)  
            y2 = layers.Conv1D(conv_chanel, kernel, activation='sigmoid', padding='same')(y)  
            y = layers.Multiply()([y1,y2])  
            y = layers.Dropout(dropout_rate)(y)  
        else:  
            for i in range(conv_layer):  
                y = layers.Conv1D(conv_chanel, kernel, activation='relu', padding='same')(y)  
                y = layers.Dropout(dropout_rate)(y)  
  
    y = layers.Dense(16, activation=None)(y)  
    y = CRF(dtype='float32')(y)  
    self.model = ModelWithCRFLoss(models.Model(x, y))  
    self.model.compile(optimizer='adam')  
    self.crf = True
```

Embedding

Convolution

CRF

Hyperparameters:

conv_layer	5
kernel	3
conv_channel (node)	100
dropout	0.2

Why our model only use Character Embedding

- Having Word Embedding allows the alg. to better capture word segment from sequence of characters because alg. was provided with cheat sheet in training data (that contains possible words)
- In original paper they applied this model to Chinese language which normally has **1-4 characters per word**.
- While Thai language normally has **2-10+ characters per word** which in turn cost a lot more memory space than Chinese language counterparts

Why our model only use Character Embedding (Example)

Let assume we have 50 dimensions for each word in word vector on both language and sentence length of 100 characters.

Memory cost per sentence is

[sentence length] x [dimensions] x [number of possible word] x [4 byte]

Chinese case (4 characters per word): number of possible word is 10

$[100] \times [50] \times [10] \times [4 \text{ byte}] = 200,000 \text{ byte}$

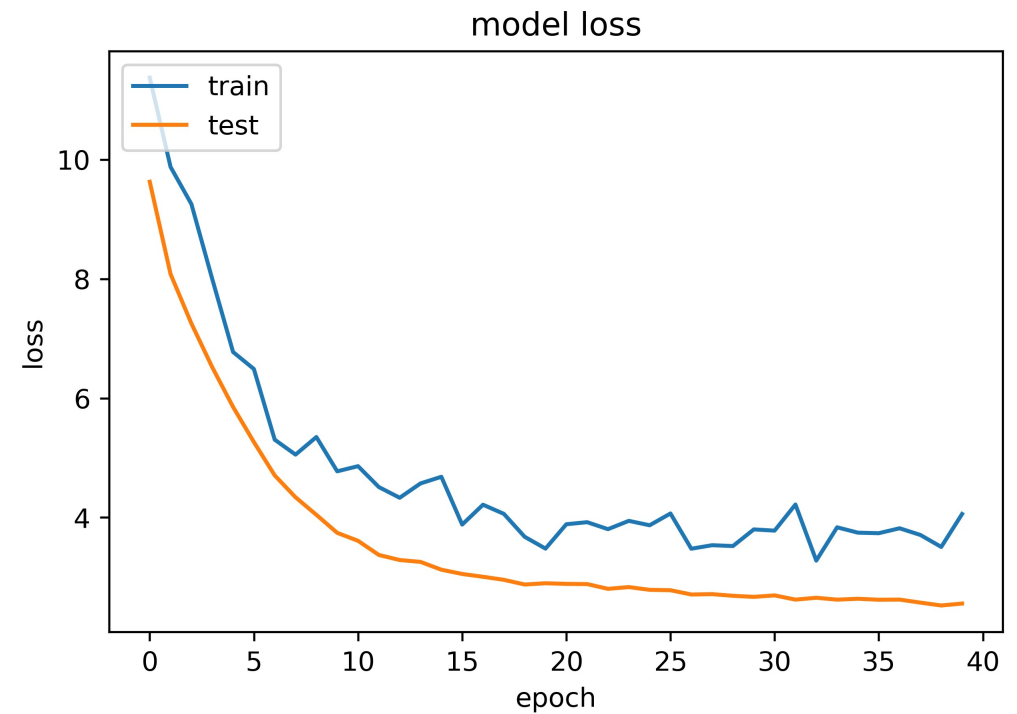
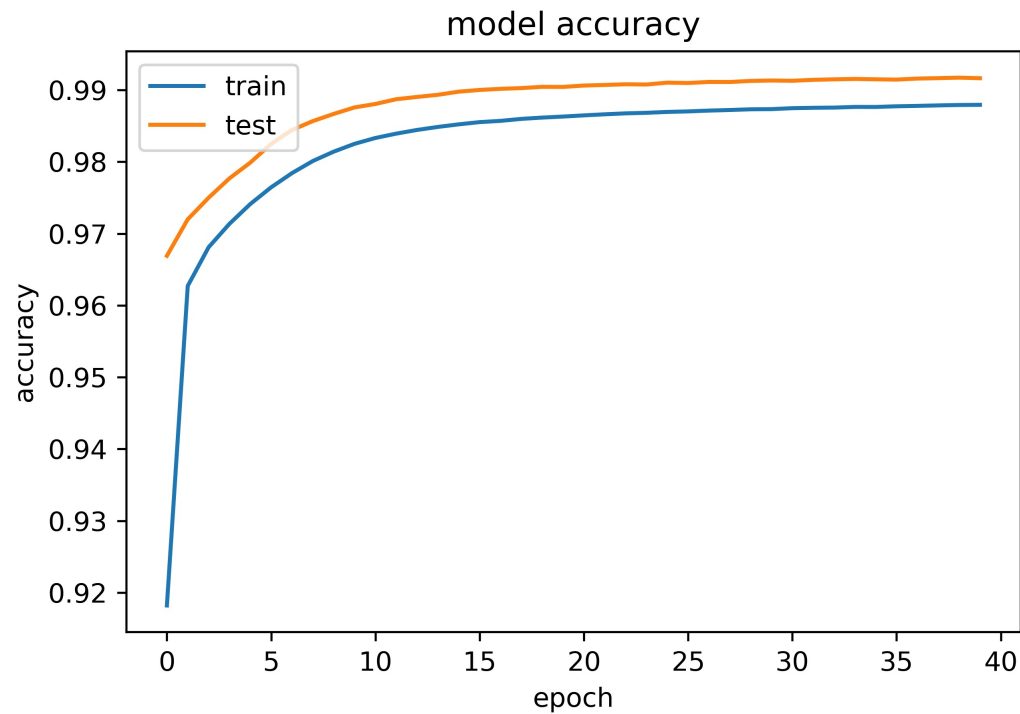
Thai case (10 characters per word): number of possible word is 55

$[100] \times [50] \times [55] \times [4 \text{ byte}] = 1,100,000 \text{ byte}$

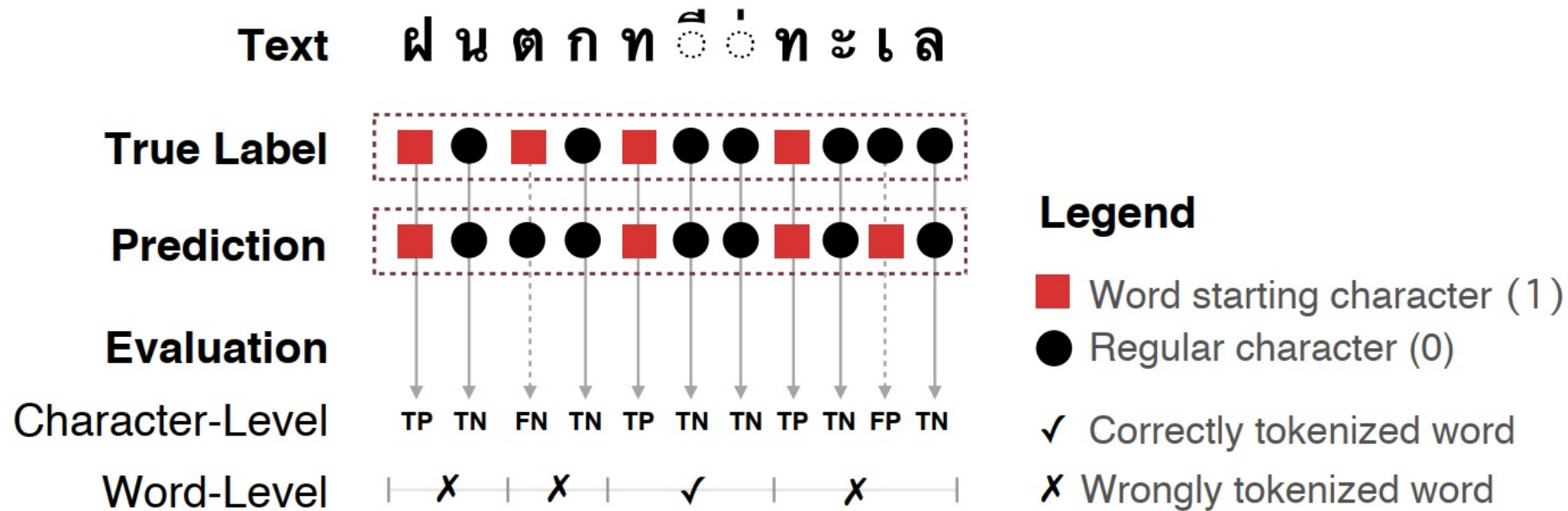
And now multiply this number with batch size!

Fine Tuning Parameters

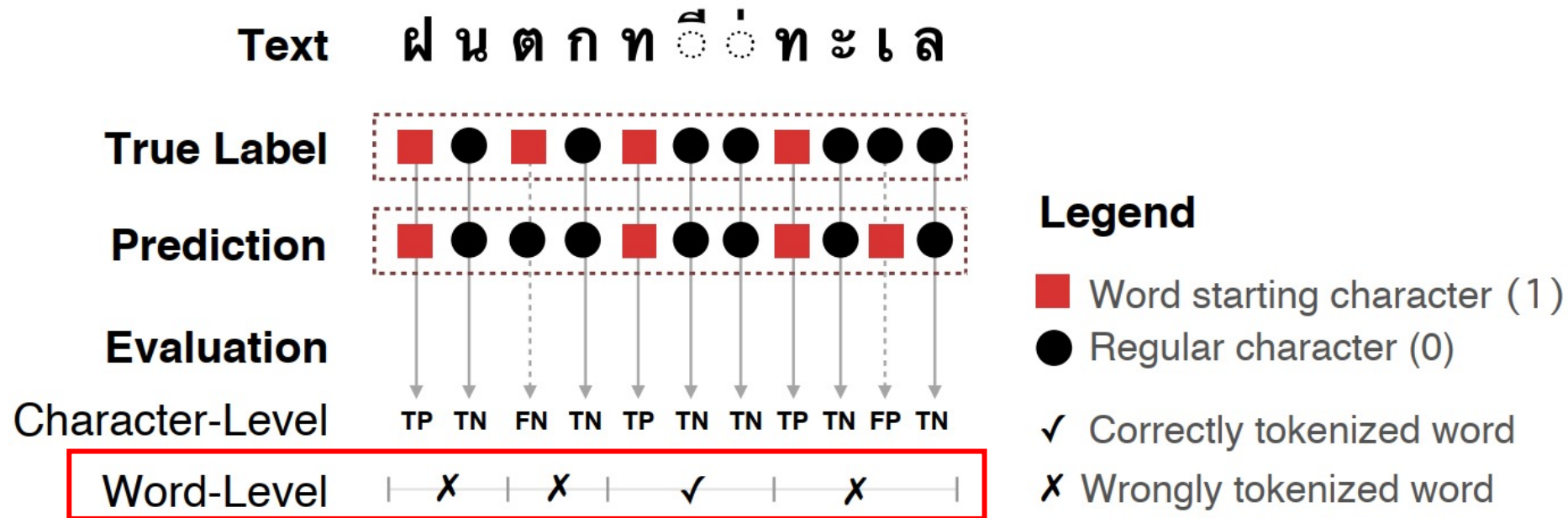
- Compare accuracy & loss changes in each epoch



Fine Tuning Parameters



Fine Tuning Parameters



Experiments and results

- Character level:

Precision: 0.9695

Recall: 0.9895

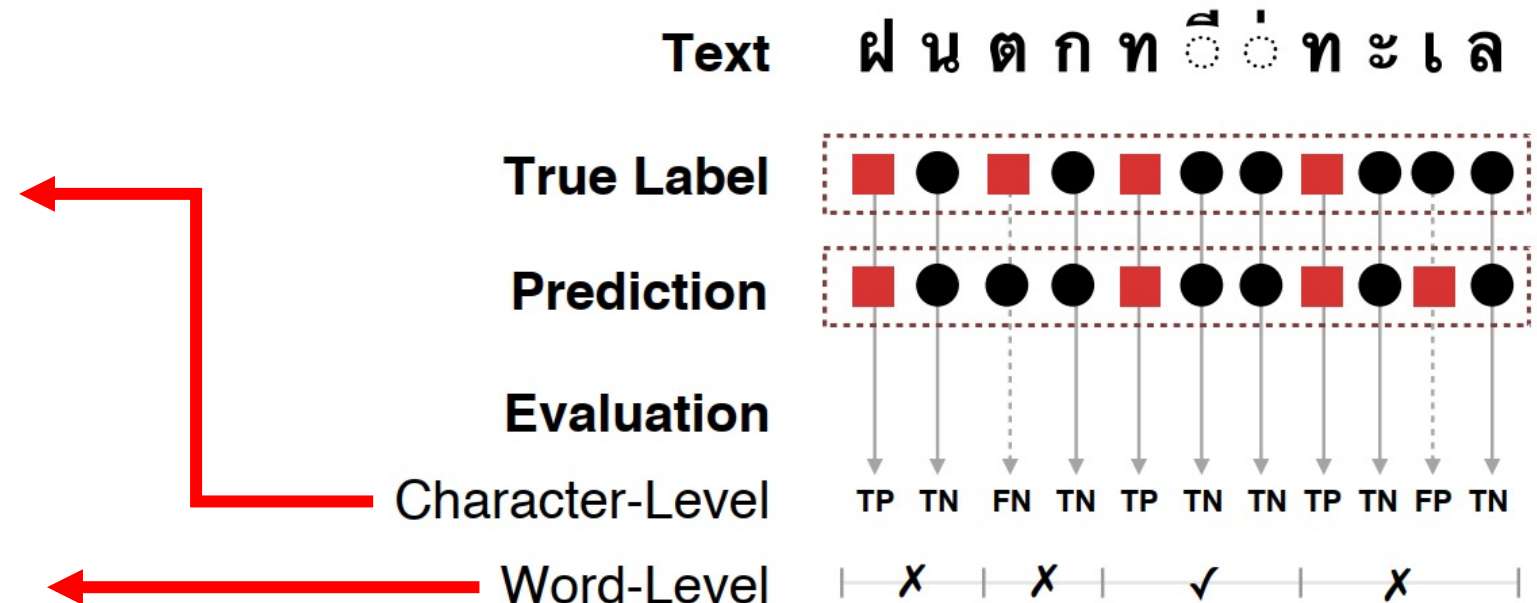
F-score: 0.9794

- Word level:

Precision: 0.9411

Recall: 0.9605

F-score: 0.9507



Attacut

'ธนาคารกรุงไทย|สร้าง|ปรากฏการณ์|ครั้ง|ใหม่| |สู่|ศึก|ดิจิทัล|ใน|วงการ|ธนาคารไทย|ที่|แข่งขัน|กัน|อย่าง|ดุเดือด| |ด้วย|การ|เปิด|ตัว|บริษัท|ลูก| |อินฟินิธัส| นาย|
|กรุงไทย จำกัด| |(Infinitas| |by| Krunghthai)| |ทำ|การ|วิจัย|และ|พัฒนา|ผลิตภัณฑ์|ทาง|การ|เงิน|ดิจิทัล|รูปแบบ|ใหม่'

'Zoom| |รายงาน|ผล|ประกอบ|การ|ประจำ|ไตรมาส|ที่| |3| |ตาม|ปี|การ|เงิน|บริษัท| |2021| |สิ้นสุด|วัน|ที่| |31| |ตุลาคม| |2020| |รายได้|รวม|เพิ่ม|ขึ้น| |3
67|%| |เทียบ|กับ|ช่วง|เดียว|กัน|ใน|ปี|ก่อน|เป็น| |777|.7| |ล้าน|ดอลลาร์| |และ|มี|กำไร|สุทธิ|แบบ| |GAPP| |198|.4| |ล้าน|ดอลลาร์'

WordSeg

'|ธนาคารกรุงไทย|สร้าง|ปรากฏการณ์|ครั้ง|ใหม่| |สู่|ศึก|ดิจิทัล|ใน|วงการ|ธนาคารไทย|ที่|แข่งขัน|กัน|อย่าง|ดุเดือด| |ด้วย|การ|เปิด|ตัว|บริษัท|ลูก| |อินฟินิธัส|
| |นาย| |กรุงไทย จำกัด| |(Infinitas| |by| |Krunghthai)| |ทำ|การ|วิจัย|และ|พัฒนา|ผลิตภัณฑ์|ทาง|การ|เงิน|ดิจิทัล|รูปแบบ|ใหม่|'

'|Zoom| |รายงาน|ผล|ประกอบ|การ|ประจำ|ไตรมาส|ที่| |3| |ตาม|ปี|การ|เงิน|บริษัท| |2021| |สิ้นสุด|วัน|ที่| |31| |ตุลาคม| |2020| |รายได้|รวม|เพิ่ม|ขึ้น|
| |367|%| |เทียบ|กับ|ช่วง|เดียว|กัน|ใน|ปี|ก่อน|เป็น| |777|.7| |ล้าน|ดอลลาร์| |และ|มี|กำไร|สุทธิ|แบบ| |GAPP| |198|.4| |ล้าน|ดอลลาร์|'

Deepcut

'ธนาคารกรุงไทย|สร้าง|ปรากฏการณ์|ครั้ง|ใหม่| |สู่|ศึก|ดิจิทัล|ใน|วงการ|ธนาคารไทย|ที่|แข่งขัน|กัน|อย่าง|ดุเดือด| |ด้วย|การ|เปิด|ตัว|บริษัท|ลูก| |อินฟินิธัส|
|นาย กรุงไทย จำกัด| |(Infinitas| |by| Krunghthai)| |ทำ|การ|วิจัย|และ|พัฒนา|ผลิตภัณฑ์|ทาง|การ|เงิน|ดิจิทัล|รูปแบบ|ใหม่'

'Zoom| |รายงาน|ผล|ประกอบ|การ|ประจำ|ไตรมาส|ที่| |3| |ตาม|ปี|การ|เงิน|บริษัท| |2021| |สิ้นสุด|วัน|ที่| |31| |ตุลาคม| |2020| |รายได้|รวม|เพิ่ม|ขึ้น|
|367|%| |เทียบ|กับ|ช่วง|เดียว|กัน|ใน|ปี|ก่อน|เป็น| |777|.7| |ล้าน|ดอลลาร์| |และ|มี|กำไร|สุทธิ|แบบ| |GAPP| |198|.4| |ล้าน|ดอลลาร์'

WordSeg

'กนก|คน|ตลก|ชวน|ดวง|กมล|คน|**ผอมรชัมภมรตมดอม**|ตอก|ขจรสอง|คน|ช้อบ|จอด|รถ|ตรง|ตรอก|ยอม|ทน|อด|นอน|อดกรนรอย|ลภมรตมดก|หอม|บน|ขอนแก่น|ตรง|คลองมอญ|ลม|บน|หวน|สอบ|จวนปอย|ผม|ปรก|คอ|สอง|สมรสมพร|คน|จร|พบ|สอง|**อรชร|สมพร**|ปอง|สองสมรยอม|**ลง|คลอง|ลอยคอ**|มอง|สอง|อรชร|มอง|อก|มอญ|คอ|มอง|ผม|มอง|จวน|สอง|คน|จวนสมพร|บอภ|ชวน|สอง|คน|ถอน|สมอ|ลงชลลอง|วอน|สอง|หน|สอง|อร|ชร|ถอย|หลบ|สมพร|วอน|จวน|พลพรรค|สด|สว้ย|หมด|สนกร|กนก|ชวน|ดวงกมล|ชง|นม|ผง|รชัมภมรบนดอนฝน|ตก|ตลอดจวน|ถนบปอน|จอม|ปลวก|ตรง|ตรอก|จอด|รถ|ถลอกปอก|ลง|**สอง|สมรรอนก**|ปรอทจก|มดจก|ปลวกจก|หนอน|ลง|คอ|สมพร|คง|ลอย|คอ|ลอย|วน|บอภ|สอพลอ|คน|สว้ย|ผสม|บท|สวด|ของ|ขอม|คน|หนอ|คน|สม|พร|สวด|วนจวน|อรชร|สอง|คน|จวน|จวนยงยง|คอ|ตก|ยอม|นอน|ลง|บน|บท|สมพร|ยก|ชอง|ผง|ทอง|ปลอม|ผสม|ลง|นม|ชง|ของ|สอง|สมรสมพร|ถอน|ผม|นวล|ลลอ|สอง|คน|ป็น|ผสม|ตอน|หลอม|รวมนม|ชง|สมพร|สวด|บท|ขอมถอยววน|หก|หน|ขอรรค|ตอน|วอน|ผอง|ช่น|จง|อวย|พร|สอง|ดวง|สมรรอด|ปลอด|นรก|คน|คน|จร|หมอน|สกปรก|ฝน|ตก|จวน|จอม|ปลวก|ยวบ|ลง|มด|ปลวก|หนอน|ออกชอกซอน|ลง|ผสม|นม|ชง|จวน|บท|สวด|หมด|ผ|สมพร|คน|สกปรก|คง|หลง|ยก|นม|ชง|ชด|ลง|คอ|รอ|ครอบครอง|สอง|คน|สว้ย|ปลวก|มด|หนอน|ลวนชอกซอน|จวน|สมพร|ปวด|คอง|อลงหนอน|ครวญ|นอน|หงอ|ช่ม|บน|กอง|หนอนกอง|ปลวก|รอ|หมอ|ตรวจ|ลม|ฝน|สงบ|ลง|ผอง|ปวงชน|พล|พรรค|ครบ|คน|ของ|**สอง|อร|ชร|ยก|พลสมทบชกกองหวด**|ตบ|สมพร|จวน|ถดถอย|ตกตม|จมลง|คลอง'

' กนก | คน | ดลกชวน | ดวง | กมล | คน | **ผอม | รอยชมภมรตม | ดอม | ดอก |** ขจร | สอง | คน | ขอบ | จอด | รก | ตรง | ตรงก | ยอม | ทนอด | นอน | อด | กรน | รอยลภมรตม | ดอก | หอม | บน | ข
อน | ตรง | คลองมอญ | ลม | บน | หวน | สอม | จน | ปอย | ผม | ปรักค | สอง | สมรสมพร | คน | จร | พม | สอง | **อรชรสมพร |** ปอง | สอง | สมร | ยอม | **ลิง | คลองลอย | คอ |** มอง | สอง | อรชร | ม
อง | อก | มอง | คอ | มอง | ผม | มอง | จน | สอง | คน | จงนสมพร | บอก | ชวน | สอง | คน | ถอน | สมอ | ลง | ชลลองวอน | สอง | หน | สอง | อรชรถอย | หลบ | สมพร | วอน | จน | พล | พรรค | สด
| สวาย | หมด | สนกรก | นกชวน | ดวง | กมลชงนม | ผง | รอยชมภมร | บนดอน | ฝน | ตก | ตลอดจน | ถน | ปอน | จอม | ปลวก | ตรง | ตรงก | จอด | รก | ถลอก | ปอก | ลง | **สอง | สมร | มอง | นก |**
ปรอท | จ | ก | มดจก | ปลวก | จก | หนอน | ลง | คอสมพร | คง | ลอย | คอ | ลอยวน | บอก | สอ | พลอ | คน | สวาย | ผสม | บท | สวด | ของ | ขอม | คน | หนอ | คน | สมพร | สวดวน | จน | อรชร | สอ
ง | คน | จงน | จงวาย | งวาย | จงค | ตก | ยอม | นอน | ลง | บน | บท | สมพรยก | ของ | ผง | ทอง | ปลอม | ผสม | ลง | นมชง | ของ | สอง | สมรสมพร | ถอน | ผม | นวล | ลอ | สอง | คน | ปน | ผสม
| ดอน | หลอม | รวม | นม | ชงสมพร | สวด | บท | ขอม | ถอย | ว | กวน | หก | หน | ขอ | พรรค | ดอนวอน | ผอง | ชน | จงวาย | พร | สอง | ดวง | สมร | รอด | ปลอด | นรก | คน | คน | จร | หมอน | สก
ปรก | ฝน | ตก | จน | จอม | ปลวก | ยวน | ลง | มด | ปลวก | หนอน | ออก | ซอกซอน | ลง | ผสม | นม | ชง | จน | บท | สวด | หมด | ผล | สมพร | คน | สกปรก | คง | หลง | ยก | นม | ชง | ชด | ลง | ค
อ | รอ | ครอบครอง | สอง | คน | สวาย | ปลวก | มด | หนอนอลวน | ซอกซอน | จน | สมพร | ปวด | คอง | อ | ลง | หอน | นอน | ครวญ | นอน | หงอชม | บน | กอง | หนอนกอง | ปลวกรอ | หมอ | ตร
ว | ลม | ฝน | สงบ | ลง | ผอง | ปวง | ชน | พล | พรรค | ครบ | คน | ของ | **สอง | อรชรยกพลสมทม | ชก | ถอง | หวด | ดม |** สมพร | จน | ถดถอย | ตกตม | จม | ลง | คลอง | '

'นก|คน|ตลก|ชวน|ดวง|กมล|คน|**ผอม|รชมภมรตมดอม**|ตอก|ขจร|สอง|คน|ชอบ|จอด|รถ|ตรง|ตรงอก|ยอม|ทนอดนอนอดกรนรอย|लग्न|รตม|ดอก|หอม|บน|ชวน|ตรง|คลองมอญลม|บน|หวาน|สอ|จน|ป๋อย|ผม|ปรก|คอ|สอง|สมรสมพรคน|จรพบสอง|**อรชรสมพร**|ป่อง|สอง|สมรยอม|**ลงคลองลอย|คอ**|น่อง|สอง|อรชรมอง|อก|ม|อง|คอ|มอง|ผม|มอง|จน|สอง|คน|จกนสมพร|บอก|ชวน|สอง|คน|ถอนสมอลง|ชลล่อง|วอน|สอง|หน|สอง|อรชรถ|อย|หลบ|สมพรวอน|จน|พล|พรรค|สด|สว|ย|หมด|สนกรกนกชวน|ดวง|กม|ล|ชงนม|ผง|รอ|ชมภมรบน|ดอน|ฝน|ตก|ตลอดจน|ถนนปอน|จอม|ปลวก|ตรง|ตรงอก|จอด|รถ|ถลอกปอก|ลง|**สอง|สมรมอง|นก**|ปรอท|จก|มด|จก|ปลวก|จก|หนอนลง|คอสมพร|คงลอย|คอ|ลอยวน|บอก|สอ|พลอ|คน|สว|ย|ผสม|บท|สวด|ของ|ขอม|คน|หนอ|คน|สมพรสวด|วน|จน|อรชร|สอง|คน|จนจน|ววยววยงคอ|ตก|ยอม|นอน|ลง|บน|บก|สมพรยก|ซ่ง|ผง|ทอง|ปลอมผสม|ลงนมชง|ของ|สอง|สมรสมพรถอน|ผม|นวล|ล่อสอง|คน|ป็น|ผสม|ดอน|หลอม|รวม|นม|ชง|สมพร|สวด|บท|ขอ|ม|ถอยวากน|หก|หน|ขวรรด|ดอน|วอน|ผอง|ชน|จง|อวยพร|สอง|ดวง|สมรรอด|ปลอด|นรก|คน|คน|จรหมอน|สกปรก|ฝน|ตก|จน|จอม|ปลวกยวบลง|มด|ปลวก|หนอน|ออก|ซอกซอน|ลง|ผสม|นม|ชง|จน|บท|สวด|หมด|ผล|สมพร|คน|สกปรก|คง|หลงยก|นม|ชง|ซด|ลง|คอ|รอ|ครอบครอง|สอง|คน|สว|ย|ปลวก|มด|หนอนอลวน|**ซอกซอน|จน|สมพรปัด|คองอลง**|หนอนนอน|ครวญ|นอน|หงอชม|บน|กอง|หนอนกอง|ปลวก|รอ|หมอ|ตรวจ|ลม|ฝน|ส่งบ|ลง|ผอง|ปวง|ชน|พล|พรรค|ครบ|คน|ของ|**สองอรชรยกพลสมทบชก**|ถอง|หวด|ตบ|สมพร|จน|ถดถอย|ตกตมจม|ลง|คลอง'

We can segment English word as well!

WordSeg

```
'|Although| |GLU| |turns| |out| |to| |be| |intrinsically| |simple|,| |the| |description| |of| |GLU| |from| |the| |original|  
|paper| |has| |been| |confusing| |to| |some| |of| |the| |readers|.| |When| |I| |worked| |on| |the| |CycleGAN| |based| |voic  
e| |conversion|,| |I| |did| |not| |implement| |correctly| |for| |the| |first| |time|.| |After| |a| |few| |years| |when| |I|  
|looked| |back| |at| |the| |paper|,| |I| |almost| |misunderstood| |it| |again|.| |The| |official| |PyTorch| |GLU| |function|  
|was| |also| |very| |confusing| |to| |the| |users|.|'
```

Thank you!