Lecture 12. Subword Models

Css224n Natural language Processing with Deep Learning

UOS STAT NLP Study
Hyehyeon Moon

2021.01.22

01

INDEX

모델에 사용하는 언어학의 최소단위

1. Phonetics(음성학) : 우리가 내는 음성을 물리적/생리적 현상으로 규명

2. Phonology(음운학) : phoneme(음소)-어떤 언어에서 의미구별 기능을 갖는 음성상의 최소 단위
>Categorical perception(범주지각) : /ba/와 /pa/가 들어간 두 단어를 구별할 때, 이 발음에 집중해서 단어를 구별

3. Morphology(형태론) : 최소의 의미를 갖는 단위인 형태소
>neural network(Luong, Socher, &manning2013)
>Wickelphones(generating past tense forms and trigram, Rumelhart&McClelland 1986)
>Microsoft's DSSM(semantic model and character n-grams, Huang, He, Gao, Deng, Acero, &Hect 2013)

Phonetics(first level of linguistics) ->모델에 사용못함 -> Phonology -> Morphology -> 모델에 사용했는데 꽤 괜찮게 나옴

왜 models below the word level이 필요한가?

- 언어의 특징에 따른 다양한 단어의 정의

  - No word segmentation 美国关岛国际机场及其办公室均接获
  - Words (mainly) segmented: *This is a sentence with words*
    - Clitics?
      - Separated **Je vous ai apporté** des bonbons
      - Joined فقلناها = ها +نا +قال +ف = so+said+we+it
    - Compounds?
      - Separated life insurance company employee
      - Joined Lebensversicherungsgesellschaftsangestellter

Clitics : 접어 (I'm에서의 m)

Compounds : 혼합

- Need large, open vocabulary

  - Rich morphology: nejneobhospodařovávatelnějšímu
    ("to the worst farmable one")
  - Transliteration: Christopher ↦ Kryštof
  - Informal spelling:

Models below the word level을 만들 때 주의할 점

- 대부분의 NLP work은 작성된 form의 언어를 학습함
- 하지만, 언어는 한 가지가 아닌 여러가지임!

- Digraph : 이중글자(영어의 ph나 sh 같이 두 글자가 하나의 음으로 나타남)
- Fossilized : 고착화된
- Syllabic : 음절의
- Moraic : 단음절 하나의 단위
- Ideographic : 표의문자(시각에 의해 사상 전달)

• Phonemic (maybe digraphs)	jiyawu ngabulu	Wambaya
• Fossilized phonemic	thorough failure	English
• Syllabic/moraic	ᑐᖏᓕᐊᖏᓚᐅᖅ	Inuktitut
• Ideographic (syllabic)	去年太空船二号坠毁	Chinese
• Combination of the above	インド洋の島	Japanese

Character-Level Models에 접근하는 관점 2가지

1. Word embeddings can be composed from character embeddings

   이미 word 바탕의 잘 짜여진 모델+character level로 보완

   - Generates embeddings for unknown words
   - Similar spellings share similar embeddings
   - Solves OOV problem

2. Connected language can be processed as characters

   아예 character level에서 시작해서 언어에 접근

아무 의미 없는 단위에서 시작했어도 놀랍게도 성공적임!

Purely character-level models(2번째 관점)

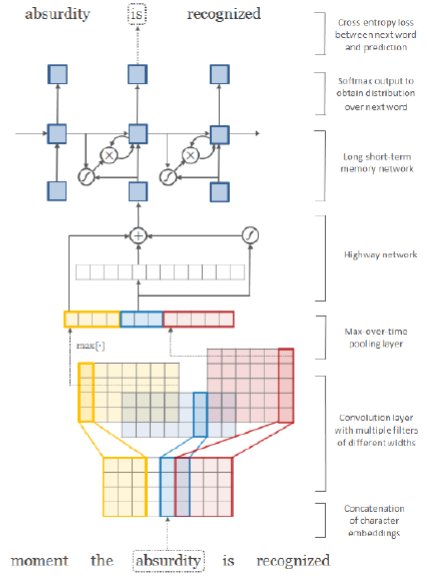
- Initially, unsatisfactory performance
  - (Vilar et al., 2007; Neubig et al., 2013)
- Decoder only
  - (Junyoung Chung, Kyunghyun Cho, Yoshua Bengio. arXiv 2016).
- Then promising results
  - (Wang Ling, Isabel Trancoso, Chris Dyer, Alan Black, arXiv 2015)
  - (Thang Luong, Christopher Manning, ACL 2016)
  - (Marta R. Costa-Jussà, José A. R. Fonollosa, ACL 2016)

초기에는 성과가 없었지만,
점차 나아지면서 꽤 높은 성공을 이루었다.

예시

Character-Aware Neural Language Models
(Kim, Jernite, Sontag, and Rush 2015)

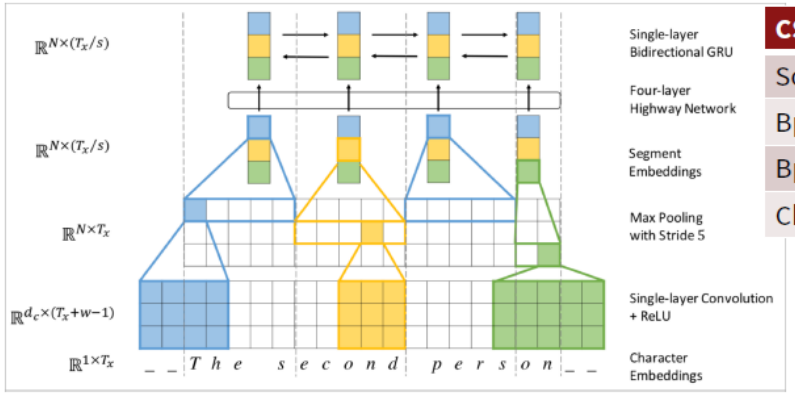
- Character-based word embedding
- Utilizes convolution, highway network, and LSTM



38

예시

Jason Lee, Kyunghyun Cho, Thomas Hoffmann. 2017.
Encoder as below; decoder is a char-level GRU



CS-En	WMT 15	Test
Source	Target	BLEU
Bpe	Bpe	20.3
Bpe	Char	22.4
Char	Char	22.5

Purely character-level models의 성공

source	Her *11-year-old* daughter , *Shani Bart* , said it felt a little bit *weird*
human	Její **jedenáctiletá** dcera **Shani Bartová** prozradila , že je to trochu **zvláštní**
char	Její **jedenáctiletá** dcera , **Shani Bartová** , říkala , že cítí trochu ***divně***
word	Její <unk> dcera <unk> <unk> řekla , že je to trochu divné
	Její **11-year-old** dcera **Shani ,** řekla , že je to trochu ***divné***

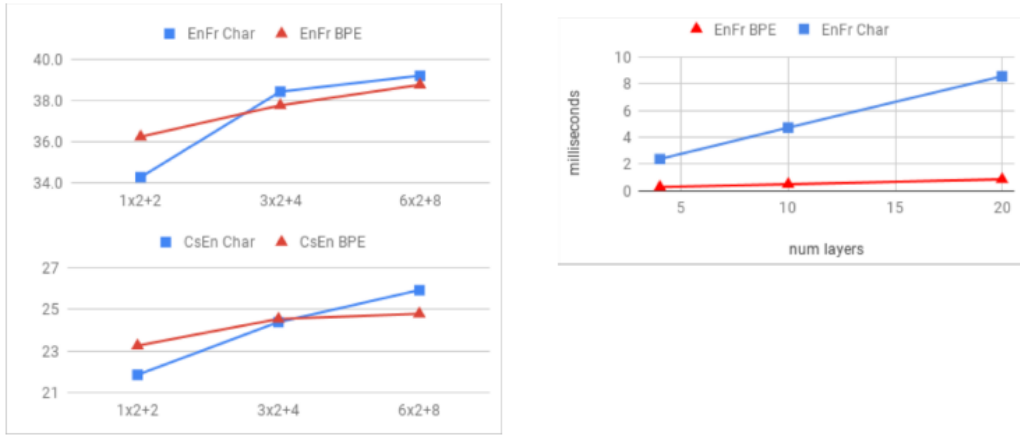
System	BLEU
Word-level model (single; large vocab; UNK replace)	15.7
Character-level model (single; 600-step backprop)	15.9

14

인간과 char level에서는 영어를 체코어로 잘 번역했지만, Word level의 모델에서는 UNK에 대해서 잘 대응하지 못하는 것을 볼 수 있음

(참고)체코는 아주 긴 음절들이 이어져 있어서 word 단위의 모델의 성능이 다른 언어보다 좋지 않다고 함, 그래서 계속 체코번역을 돌리는 거임

Revisiting Character-Based Neural Machine Translation with Capacity and Compression. 2018.
Cherry, Foster, Bapna, Firat, Macherey, Google AI



16

X축은 bi-directional LSTM encode의 layer+one directional LSTM decode의 layer를 의미=models of different size
왼쪽 그림의 y축은 성능, 오른쪽 그림의 y축은 시간

모형이 복잡할수록 Char level의 성능이 좋음, 하지만 매~~우 느리다는 단점이 있음

Sub-word models : two trends

- Subword segmentation

하나의 단어를 여러 서브워드로 분리해서 단어를 인코딩 및 임베딩하겠다는 의도를 가진 전처리
이를 통해 OOV나 희귀 단어, 신조어와 같은 문제를 완화시킬 수 있다.

1. Same architecture as for word-level model

Use smaller units: “word pieces”

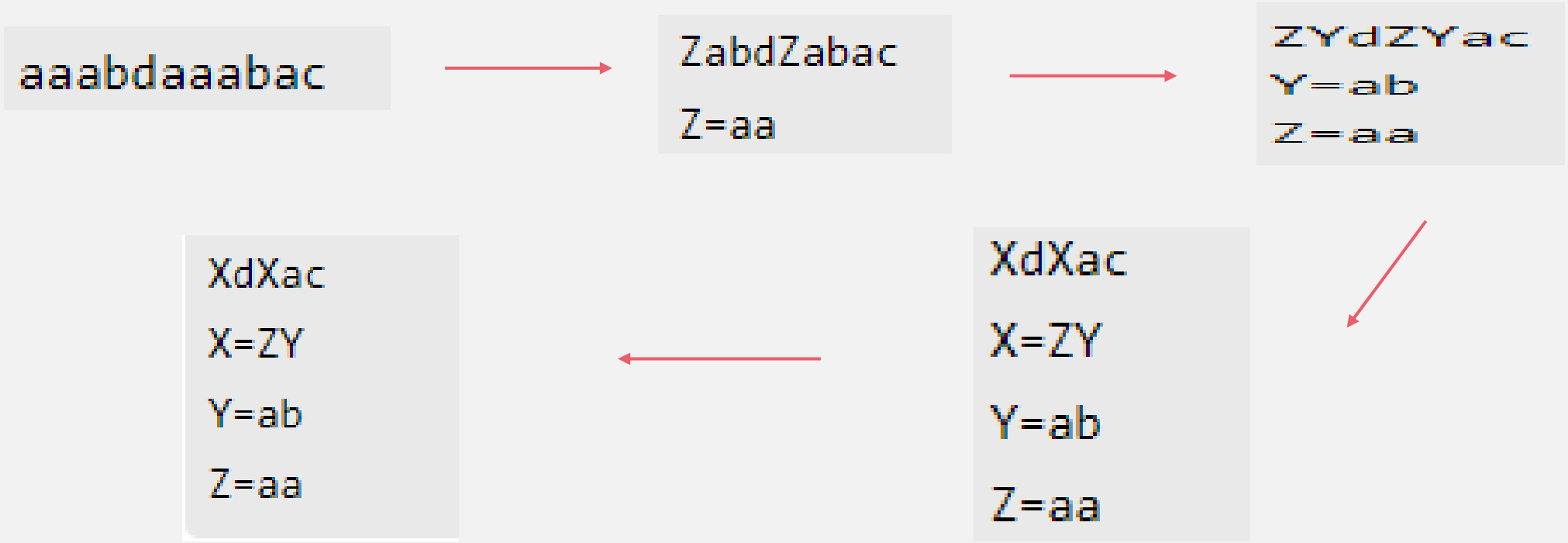
- Byte Pair Encoding
- SentencePiece model
- Character-level to build word-level

2. Hybrid architectures

Main model has words; something else for characters

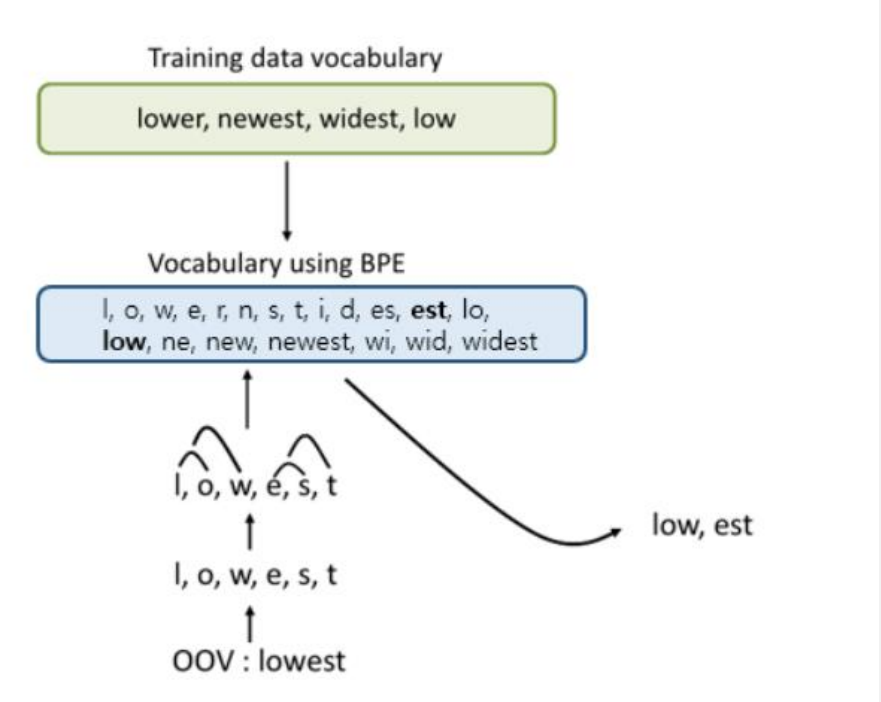
Byte Pair Encoding

- BPE(Byte pair encoding) 알고리즘 : 1994년에 제안된 데이터 압축 알고리즘
- 원래 알고리즘의 작동방법 : 기본적으로 연속적으로 가장 많이 등장한 글자의 쌍을 찾아서 하나의 글자로 병합하는 방식을 수행



Byte Pair Encoding

- 자연어 처리에서의 BPE는 서브워드 분리(subword segmentation) 알고리즘
- BPE(in NLP) : 글자(charcter) 단위에서 점차적으로 단어 집합(vocabulary)을 만들어 내는 Bottom up 방식의 접근
- 원하는 단어 집합에 이를 때까지 알고리즘을 반복함



Byte Pair Encoding

초기에 훈련 데이터에 있는 단어와 등장 빈도수 정보를 포함한 dictionary
초기 구성은 글자 단위로 분리된 상태 : 초기의 단어 집합

```
# dictionary
l o w : 5,  l o w e r : 2,  n e w e s t : 6,  w i d e s t : 3
```

```
# vocabulary
l, o, w, e, r, n, w, s, t, i, d
```

결과

```
# dictionary update!
low : 5,
low e r : 2,
newest : 6,
widest : 3
```

```
# vocabulary update!
l, o, w, e, r, n, w, s, t, i, d, es, est, lo, low, ne, new, newest, wi, wid, wi
dest
```

1회 - 딕셔너리를 참고로 하였을 때 빈도수가 9로 가장 높은 (e, s)의 쌍을 es로 통합합니다.

```
# dictionary update!
l o w : 5,
l o w e r : 2,
n e w es t : 6,
w i d es t : 3
```

```
# vocabulary update!
l, o, w, e, r, n, w, s, t, i, d, es
```

2회 - 빈도수가 9로 가장 높은 (es, t)의 쌍을 est로 통합합니다.

```
# dictionary update!
l o w : 5,
l o w e r : 2,
n e w est : 6,
w i d est : 3
```

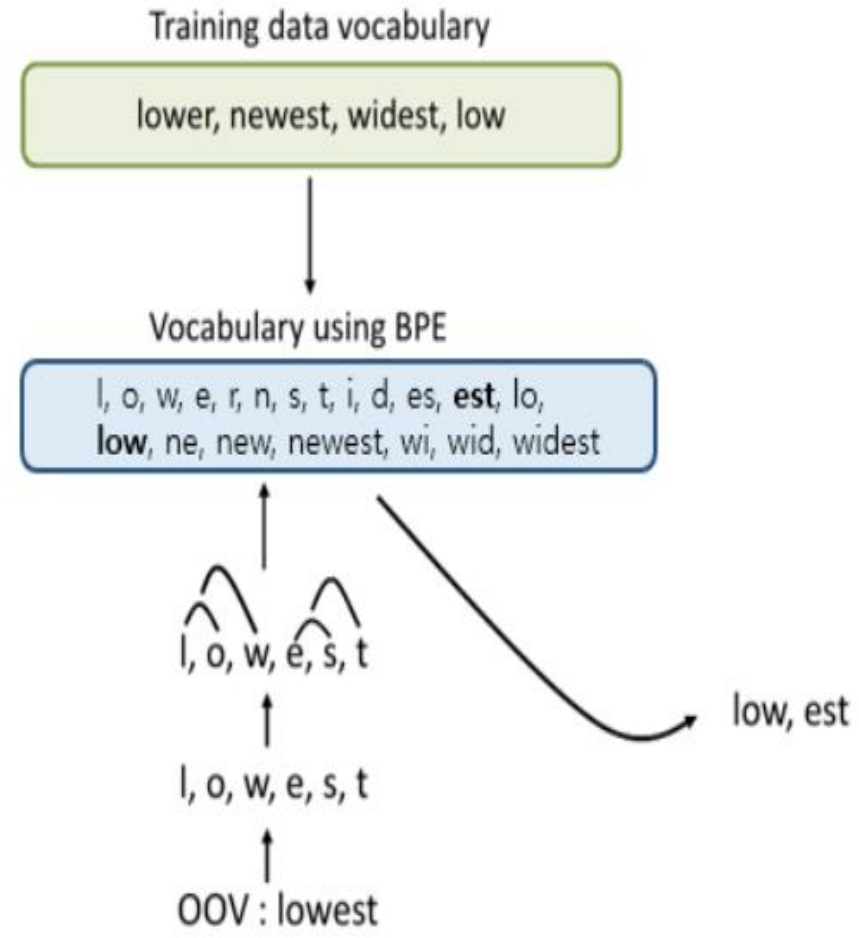
```
# vocabulary update!
l, o, w, e, r, n, w, s, t, i, d, es, est
```

3회 - 빈도수가 7로 가장 높은 (l, o)의 쌍을 lo로 통합합니다.

```
# dictionary update!
lo w : 5,
lo w e r : 2,
n e w est : 6,
w i d est : 3
```

```
# vocabulary update!
l, o, w, e, r, n, w, s, t, i, d, es, est, lo
```

Byte Pair Encoding




```
# dictionary update!
low : 5,
low e r : 2,
newest : 6,
widest : 3
```

```
# vocabulary update!
l, o, w, e, r, n, w, s, t, i, d, es, est, lo, low, ne, new, newest, wi, wid, wi
dest
```

- 테스트 과정에서 'lowest'란 단어가 등장
- 기계는 우선 'lowest'를 전부 글자 단위로 분할->즉, 'l, o, w, e, s, t'
- 기계는 위의 단어 집합을 참고로 하여 'low'와 'est'를 찾아냄->즉, 'lowest'를 기계는 'low'와 'est' 두 단어로 인코딩
- 이 두 단어는 둘 다 단어 집합에 있는 단어이므로 OOV가 아닙니다.

Wordpiece Model(WPM)

- WordPiece Model은 BPE의 변형 알고리즘
- BPE가 빈도수에 기반하여 가장 많이 등장한 쌍을 병합하는 것과는 달리, 병합되었을 때 코퍼스의 우도(Likelihood)를 가장 높이는 쌍을 병합.
- 최신 딥 러닝 모델 BERT를 훈련하기 위해서 사용됨
- 모든 단어의 맨 앞에 _를 붙이고, 단어는 서브 워드(subword)로 통계에 기반하여 띄어쓰기로 분리

WPM을 수행하기 이전의 문장: Jet makers feud over seat width with big orders at stake

WPM을 수행한 결과(wordpieces): _J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

WPM이 수행된 결과로부터 다시 수행 전의 결과로 돌리는 방법은 현재 있는 모든 띄어쓰기를 전부 제거하고, 언더바를 띄어쓰기로 바꾸면 됨

기존에 없던 띄어쓰기가 추가되어 서브 워드(subwords)들을 구분하는 구분자 역할

기존에 있던 띄어쓰기와 구분자 역할의 띄어쓰기를 구별하는 것은 단어들 앞에 붙은 언더바 _

Unigram Language Model Tokenizer

- 각각의 서브워드들에 대해서 손실(loss)을 계산
- 서브 단어의 손실이라는 것은 해당 서브워드가 단어 집합에서 제거되었을 경우,
  코퍼스의 우도(Likelihood)가 감소하는 정도를 말합니다.
- 이렇게 측정된 서브워드들을 손실의 정도로 정렬하여, 최악의 영향을 주는 10~20%의 토큰을 제거
- 이를 원하는 단어 집합의 크기에 도달할 때까지 반복

SentencePiece model

- 내부 단어 분리 알고리즘을 사용하기 위해서,
  데이터에 단어 토큰화를 먼저 진행해야 하므로 모든 언어에 사용하는 것은 쉽지않음
- 특히 한국어의 경우 단어 토큰화가 어려움
- 사전 토큰화 작업(pretokenization)없이 전처리를 하지 않은 데이터(raw data)에
  바로 단어 분리 토크나이저를 사용할 수 있도록 만든 것이 SentencePiece model
- BPE 알고리즘과 Unigram Language Model Tokenizer를 구현한 센텐스피스를 깃허브에 공개

네이버 영화 리뷰 토큰화하기!! 실습
Sentencepiece library와 tensorflow library

SentencePiece model 실습

- 실습 참고 자료

https://wikidocs.net/86657
https://wikidocs.net/24992

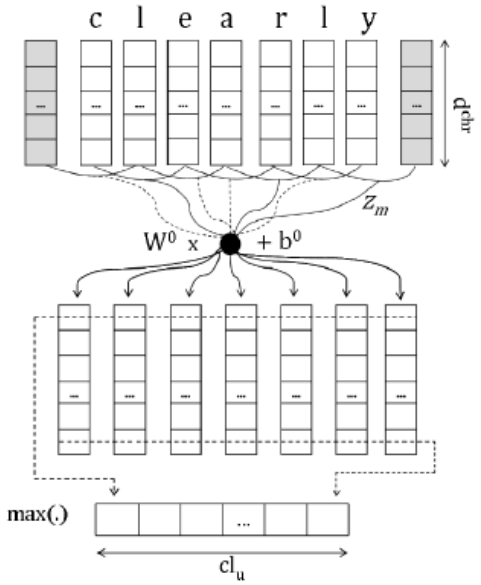
- 사용하는 라이브러리

Sentencepiece와 Tensorflow에서 제공하는 SubwordTextencoder

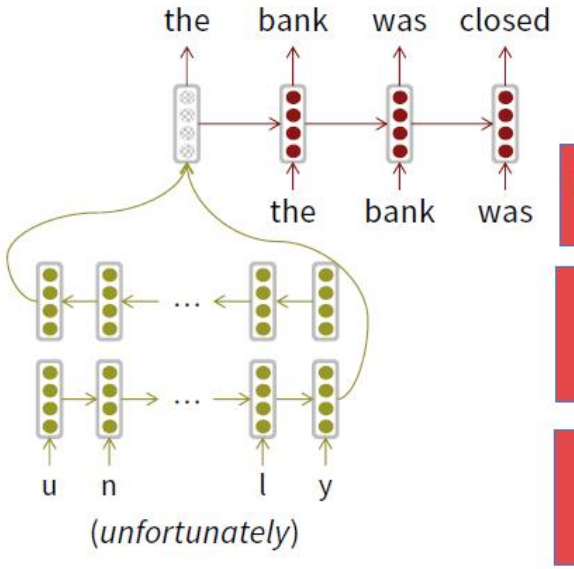
Character-level to build word-level

Learning Character-level Representations for Part-of-Speech Tagging (Dos Santos and Zadrozny 2014)

- **Convolution** over characters to generate word embeddings
- Fixed window of word embeddings used for PoS tagging



Character-based LSTM



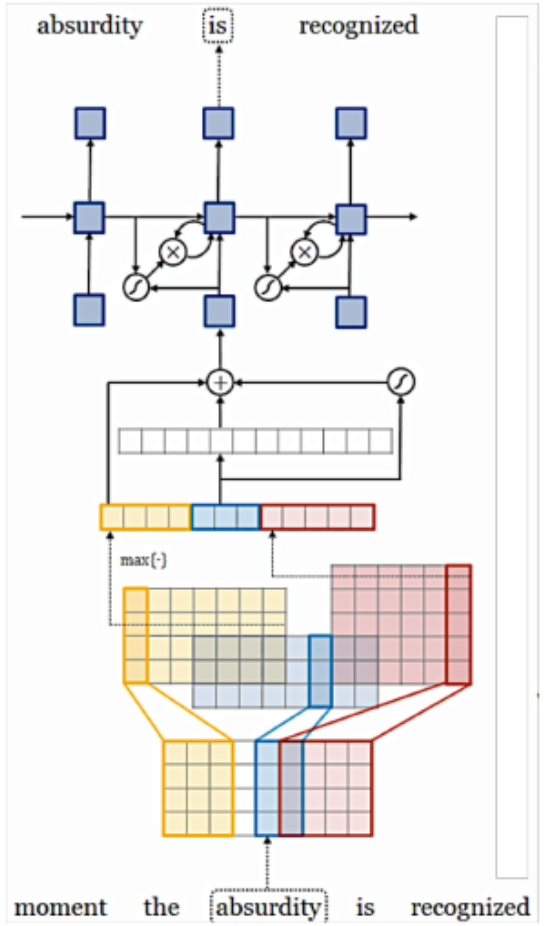
Recurrent Language Model

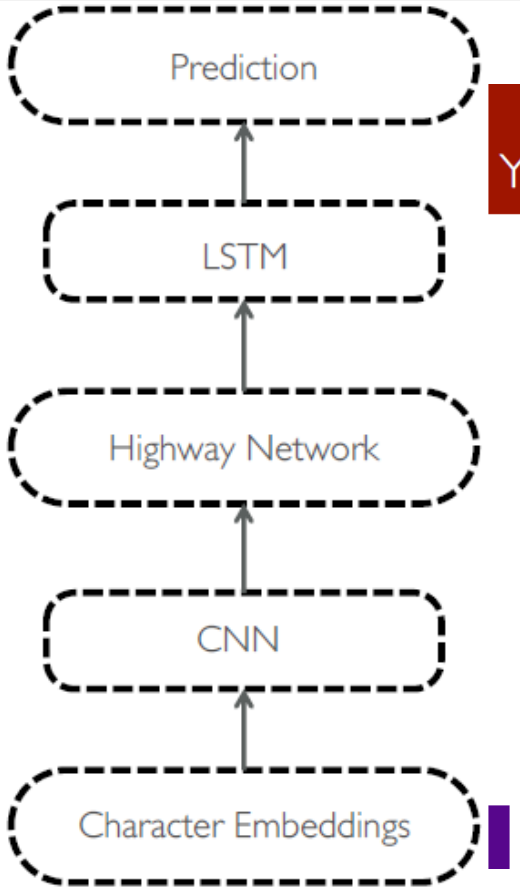
Bi-LSTM builds word representations

Used as LM and for POS tagging

Ling, Luís, Marujo, Astudillo, Amir, Dyer, Black, Trancoso. ***Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation****. EMNLP'15.*

Character-level to build word-level





Character-Aware Neural Language Models
Yoon Kim, Yacine Jernite, David Sontag, Alexander M. Rush. 2015

A more complex/sophisticated approach

Motivation

- Derive a powerful, robust language model effective across a variety of languages.
- Encode subword relatedness: *eventful, eventfully, uneventful…*
- Address rare-word problem of prior models.
- Obtain comparable expressivity with fewer parameters.

Output representation for character n-grams->highway network->word level의 output이 나오게 됨

->LSTM model at word level ->minimize perplexity like for the neural language models we saw earlier.

Character-level to build word-level

Comparable performance with fewer parameters!

	PPL	Size
LSTM-Word-Small	97.6	5 m
LSTM-Char-Small	92.3	5 m
LSTM-Word-Large	85.4	20 m
LSTM-Char-Large	78.9	19 m
KN-5 (Mikolov et al. 2012)	141.2	2 m
RNN† (Mikolov et al. 2012)	124.7	6 m
RNN-LDA† (Mikolov et al. 2012)	113.7	7 m
genCNN† (Wang et al. 2015)	116.4	8 m
FOFE-FNNLM† (Zhang et al. 2015)	108.0	6 m
Deep RNN (Pascanu et al. 2013)	107.5	6 m
Sum-Prod Net† (Cheng et al. 2014)	100.0	5 m
LSTM-1† (Zaremba et al. 2014)	82.7	20 m
LSTM-2† (Zaremba et al. 2014)	78.4	52 m

다른 language model과 비교했을 때, 더 적은 모수들을 가지고 비슷한 성능을 내는 것을 확인 할 수 있음!

In Vocabulary

	while	*his*	*you*	*richard*	*trading*
LSTM-Word	*although*	*your*	*conservatives*	*jonathan*	*advertised*
	letting	*her*	*we*	*robert*	*advertising*
	though	*my*	*guys*	*neil*	*turnover*
	minute	*their*	*i*	*nancy*	*turnover*
LSTM-Char (before highway)	*chile*	*this*	*your*	*hard*	*heading*
	whole	*hhs*	*young*	*rich*	*training*
	meanwhile	*is*	*four*	*richer*	*reading*
	white	*has*	*youth*	*richter*	*leading*
LSTM-Char (after highway)	*meanwhile*	*hhs*	*we*	*eduard*	*trade*
	whole	*this*	*your*	*gerard*	*training*
	though	*their*	*doug*	*edward*	*traded*
	nevertheless	*your*	*i*	*carl*	*trader*

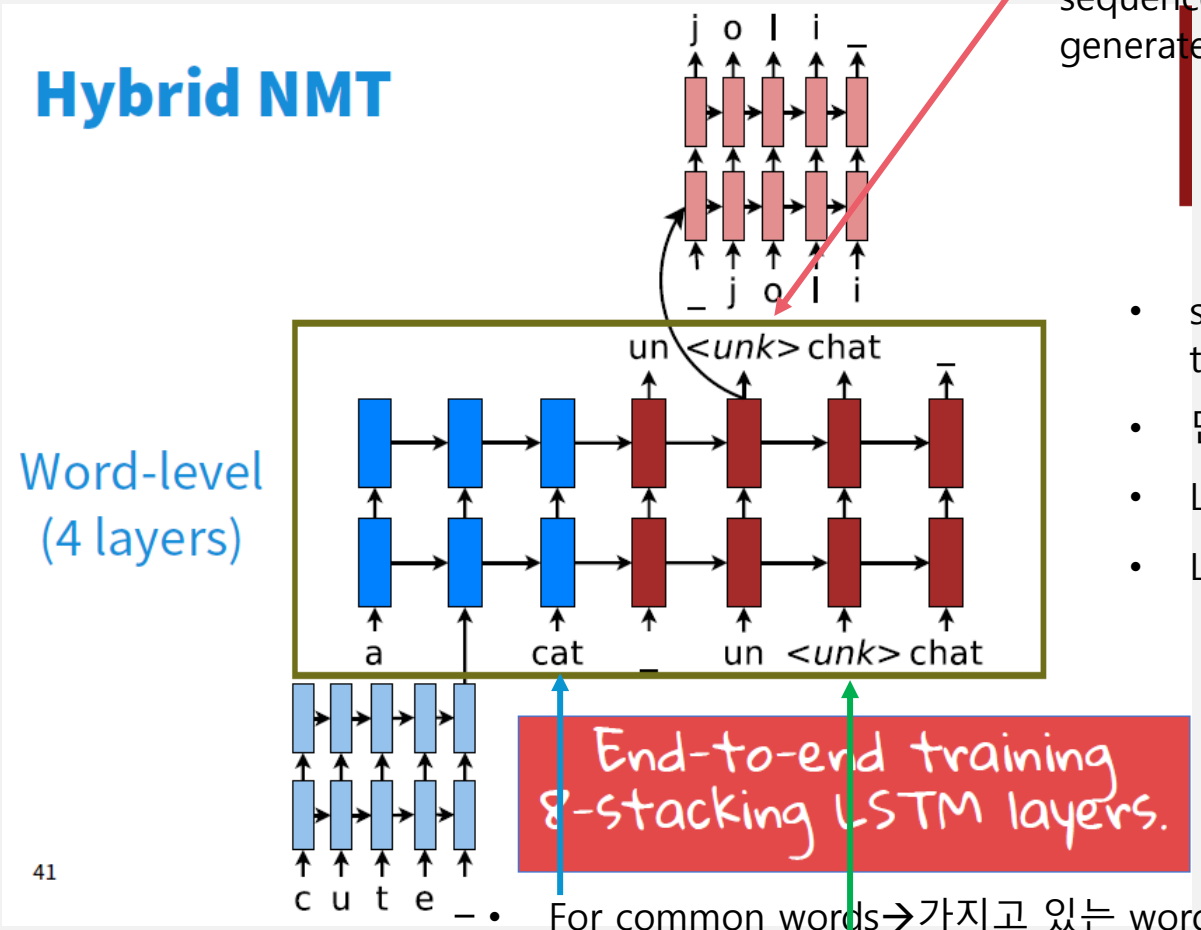
- 단어 단위는 의미상 비슷한 단어를 잘 표현하지만,
- Char 단위는 그렇지 못함
- 하지만, highway를 적용하면 의미상 비슷하게 단어를 표현할 수 있는 것을 확인

Out-of-Vocabulary

computer-aided	*misinformed*	*looooook*
–	–	–
–	–	–
–	–	–
–	–	–
computer-guided	*informed*	*look*
computerized	*performed*	*cook*
disk-drive	*transformed*	*looks*
computer	*inform*	*shook*
computer-guided	*informed*	*look*
computer-driven	*performed*	*looks*
computerized	*outperformed*	*looked*
computer	*transformed*	*looking*

- UNK에 대해서 대응하지 못함
- Char 단위는 잘 대응함
- Highway를 적용한 char 단위도 잘 대응함

Hybrid NMT



- 만약 unk symbol을 생성한다면 hidden representation을 작동하고 and feed it in as the initial input into a character level LSTm and then we have the character level LSTM generate a character sequence until it generates a stop symbol and we use that to generate words.

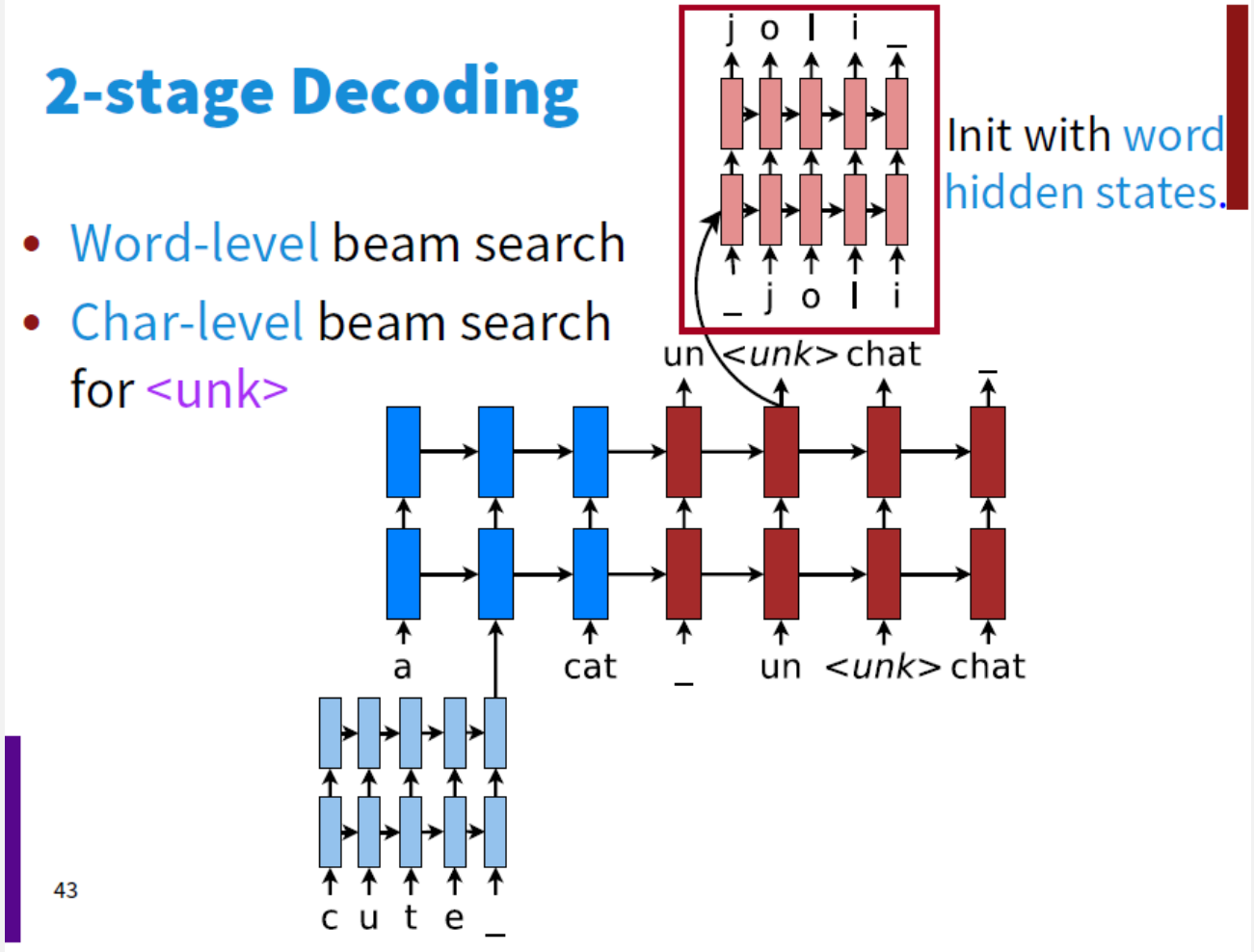
- sequence to sequence with attention LSTM neural machine translation system(4 layers)
- 단어를 생성할 때, 1600개의 단어에 대해서 softmax 이용
- Loss at the word level->softmax로 확률에 대한 손실함수를 구하면 됨
- Loss at the character level

- For common words→가지고 있는 word representation을 모델에 그냥 넣음
- For rare or unseen words->work out a word representation for them by using a character level LSTM

Hybrid NMT

2-stage Decoding

- Word-level beam search
- Char-level beam search for <unk>



Hybrid NMT의 성능

source	The author ***Stephen Jay Gould*** died 20 years after ***diagnosis*** .
human	Autor **Stephen Jay Gould** zemřel 20 let po **diagnóze** .
char	Autor **Stepher Stephe**r zemřel 20 let po **diagnóze**
word	Autor Stephen Jay <unk> zemřel 20 let po <unk> .
	Autor **Stephen Jay Gould** zemřel 20 let po **po** .
hybrid	Autor Stephen Jay <unk> zemřel 20 let po <unk> .
	Autor **Stephen Jay Gould** zemřel 20 let po **diagnóze** .

Perfect translation!

- Char-based : wrong name translation
- Word-based : incorrect alignment
- Char-based &hybrid : correct translation of diagnose

source	Her ***11-year-old*** daughter , ***Shani Bart*** , said it felt a little bit ***weird***
human	Její **jedenáctiletá** dcera **Shani Bartová** prozradila , že je to trochu **zvláštní**
word	Její <unk> dcera <unk> <unk> řekla , že je to trochu divné
	Její **11-year-old** dcera **Shani ,** řekla , že je to trochu ***divné***
hybrid	Její <unk> dcera , <unk> <unk> , řekla , že je to <unk> <unk>
	Její **jedenáctiletá** dcera **Graham *Bart*** , řekla , že cítí trochu ***divný***

- Word-based : identity copy fails
- Hybrid : Shani 이름 wrong

Hybrid NMT의 성능->Successful

source	The author *Stephen Jay Gould* died 20 years after *diagnosis* .
human	Autor **Stephen Jay Gould** zemřel 20 let po **diagnóze** .
char	Autor **Stepher Stepher** zemřel 20 let po **diagnóze**
word	Autor Stephen Jay <unk> zemřel 20 let po <unk> .
	Autor **Stephen Jay Gould** zemřel 20 let po **po** .
hybrid	Autor Stephen Jay <unk> zemřel 20 let po <unk> .
	Autor **Stephen Jay Gould** zemřel 20 let po **diagnóze** .

Perfect translation!

source	Her *11-year-old* daughter , *Shani Bart* , said it felt a little bit *weird*
human	Její **jedenáctiletá** dcera **Shani Bartová** prozradila , že je to trochu **zvláštní**
word	Její <unk> dcera <unk> <unk> řekla , že je to trochu divné
	Její **11-year-old** dcera **Shani ,** řekla , že je to trochu *divné*
hybrid	Její <unk> dcera , <unk> <unk> , řekla , že je to <unk> <unk>
	Její **jedenáctiletá** dcera **Graham *Bart*** , řekla , že cítí trochu *divný*

- Char-based : wrong name translation
- Word-based : incorrect alignment
- Char-based &hybrid : correct translation of diagnose

Hybrid는 it doesn't have any further back than that of what's in the word level model.->word단위의 문맥을 고려할 수 없다는 한계

- Word-based : identity copy fails
- Hybrid : Shani 이름 wrong

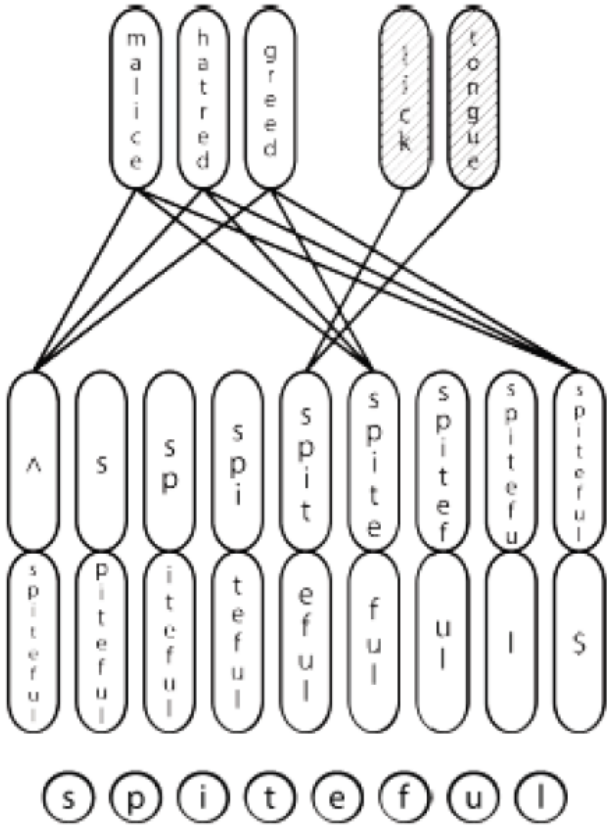
- word-level에서 unk에 대처하는 자세는 두가지로
- unigram translation of the word that it's maximally putting attention on or it could copy the word that it's maximally putting attention on

Chars for word embeddings

5. Chars for word embeddings

A Joint Model for Word Embedding and Word Morphology
(Cao and Rei 2016)

- Same objective as w2v, but using characters
- Bi-directional LSTM to compute embedding
- Model attempts to capture morphology
- Model can infer roots of words



FastText embeddings

- Aim: a next generation efficient word2vec-like word representation library, but better for rare words and languages with lots of morphology
- An extension of the w2v skip-gram model with character *n*-grams
- Represent word as char *n*-grams augmented with boundary symbols and as whole word:
- *where* = <*wh, whe, her, ere, re*>, <*where*>
  - Note that <*her*> or <*her* is different from *her*
    - Prefix, suffixes and whole words are special
- Represent word as sum of these representations. Word in context score is:
  - $s(w, c) = \sum_{g \in G(w)} \mathbf{z}_g^{\mathrm{T}} \mathbf{v}_c$
    - Detail: rather than sharing representation for all *n*-grams, use "hashing trick" to have fixed number of vectors

54

•Word embedding (Distributed vector represenatation of words)에는 다양한 방법이 있지만, 대부분의 방법들은 언어의 형태학적(Morpological)인 특성을 반영하지 못하고, 또 희소한 단어에 대해서는 Embedding이 되지 않음

•본 연구에서는 단어를 Bag-of-Characters로 보고, 개별 단어가 아닌 n-gram의 Charaters를 Embedding함 (Skip-gram model 사용)

•최종적으로 각 단어는 Embedding된 n-gram의 합으로 표현됨, 그 결과 빠르고 좋은 성능을 나타냈음