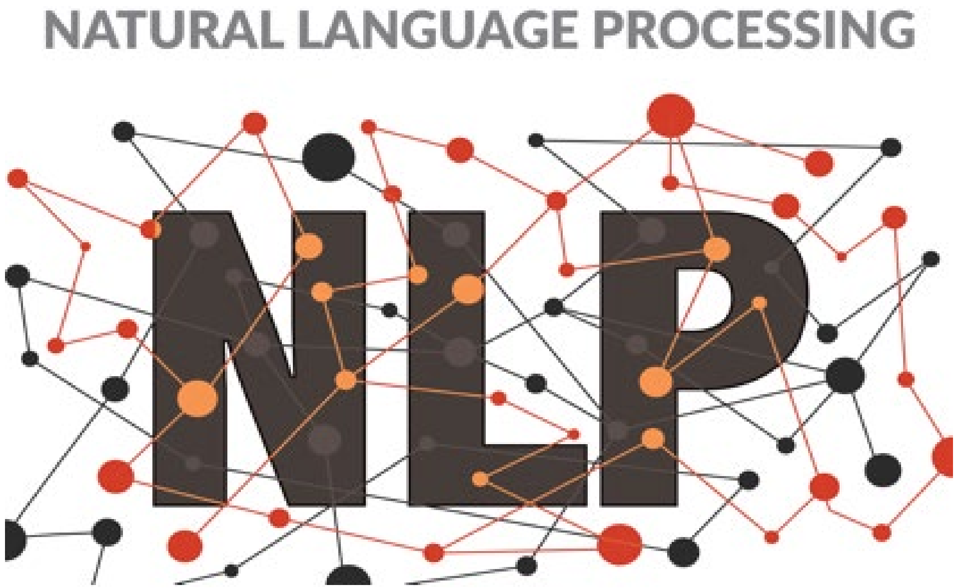
Natural Language Processing (CS224N)

Lecture 12. Information from parts of words: Subword Models

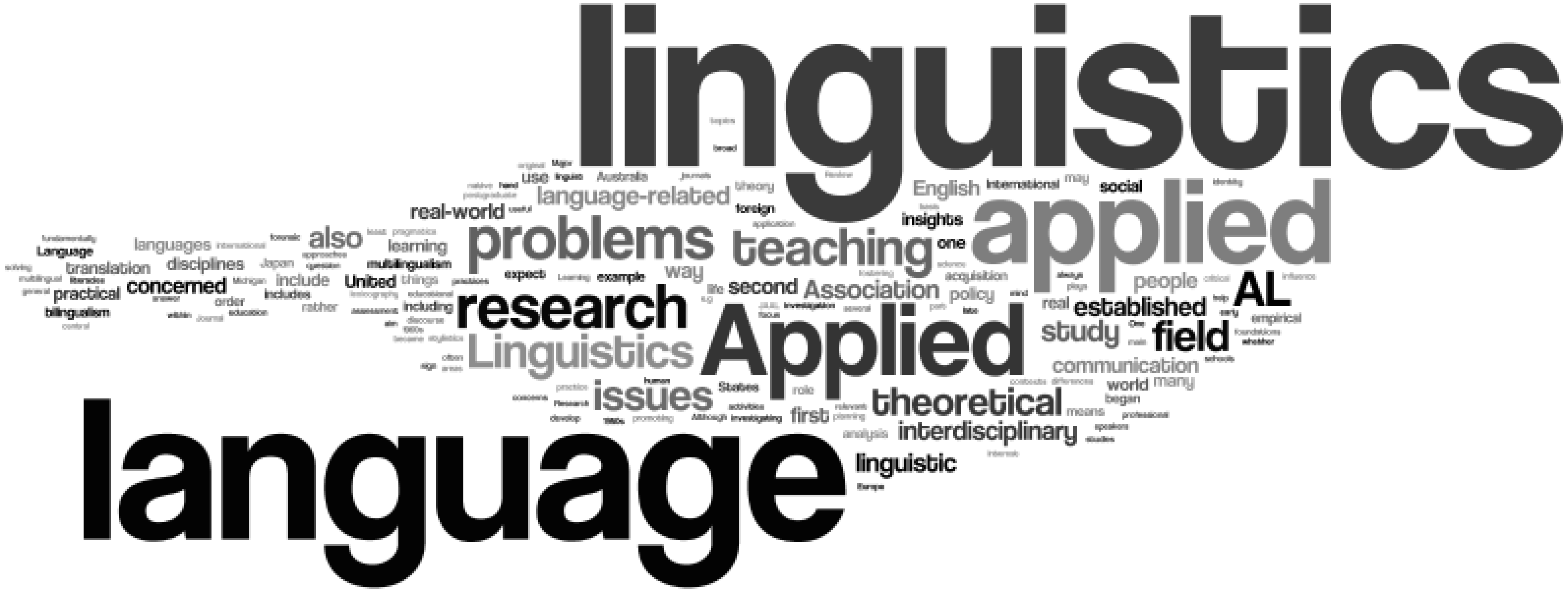
강동인, 김서영, 김수연, 손지우, 조민주



개요

1. 언어학(Linguistics)
2. Purely Character-level Model
3. Sub-word models: two trends
4. Character-level
To build word-level
5. FastText

1. 언어학(Linguistics)

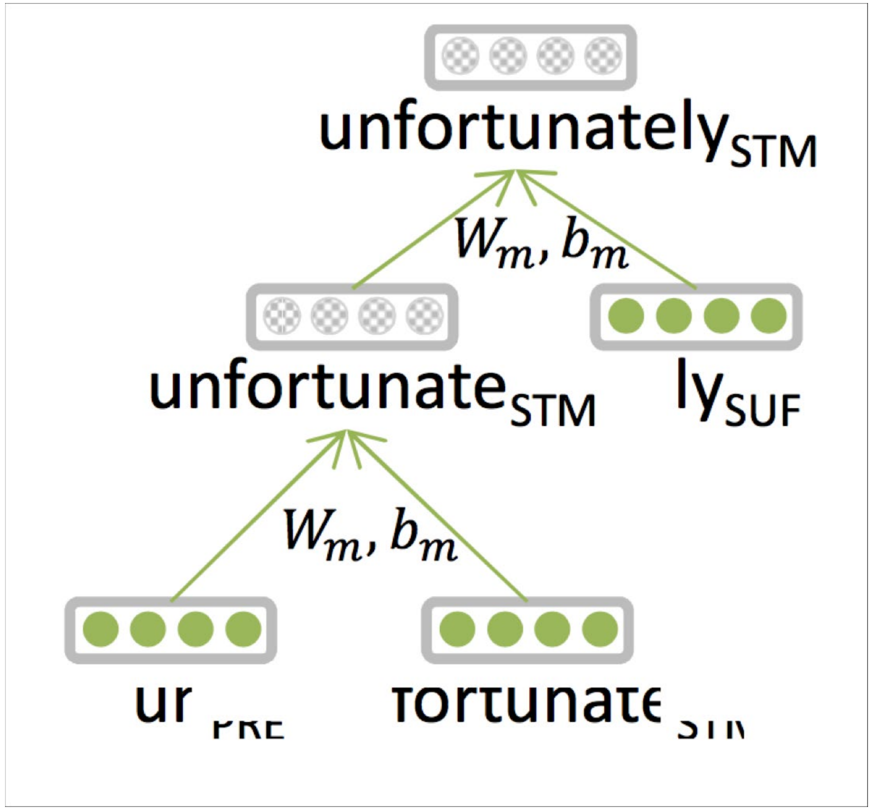


주요개념

1. Phonemes(음소)
2. Morpheme(형태소)

| 1. 언어학(Linguistics)

most unseen words are new morphological forms!



1. Phonemes(음소)

어떤 언어에서 의미 구별 기능을 갖는 음성상의 최소 단위

Categorical perception의 근거가 되는 주요 단위

2. Morpheme(형태소)

뜻을 갖는 최소 언어 단위

semantic units

문제점: 언어별 표기 체계의 차이가 존재한다

예시1. 띄어쓰기 없음 (ex.중국어)

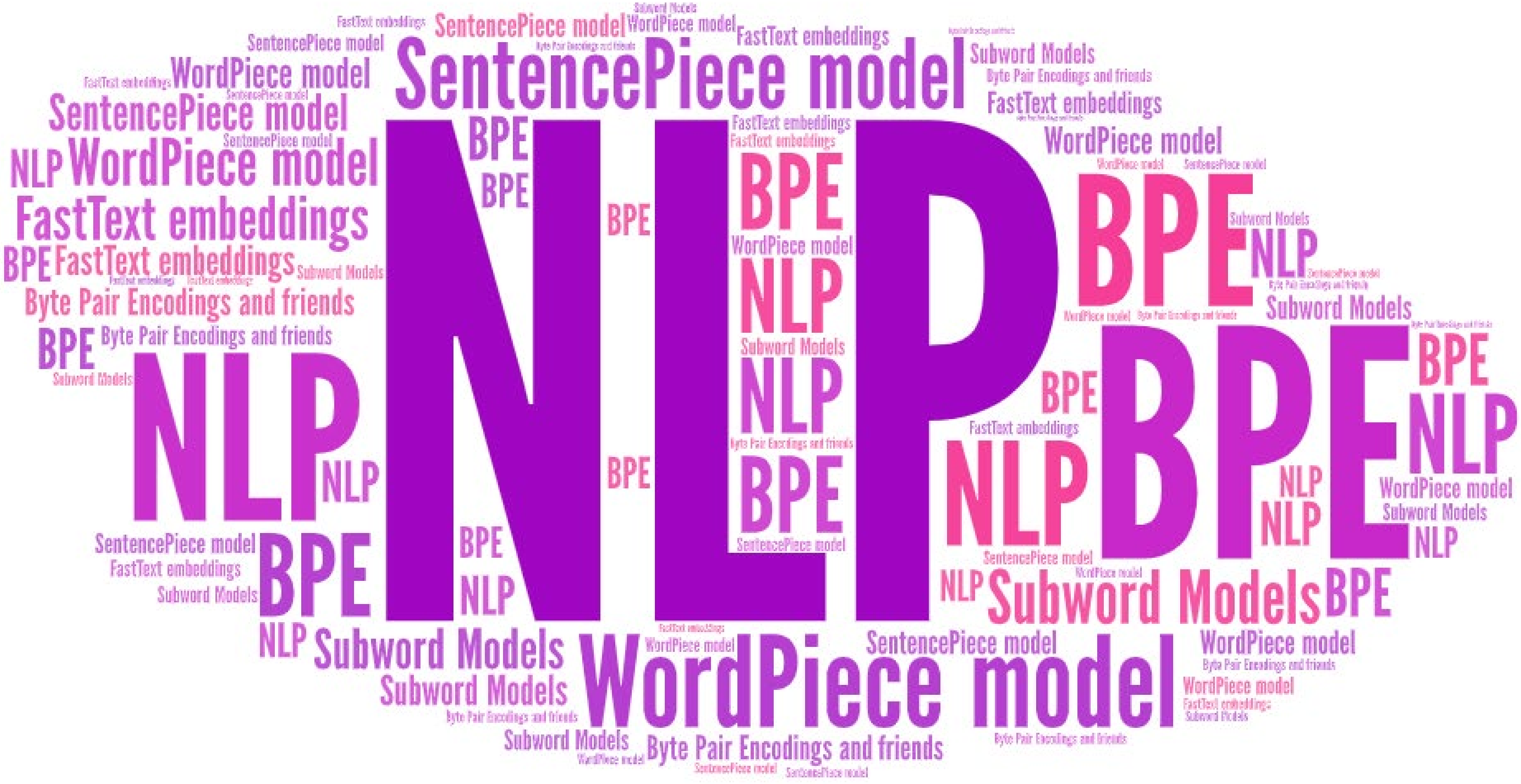
美固美島固阮机玩及其亦公室均接萩

예시2. 접어(Clitics) (ex.프랑스어, 아랍어)

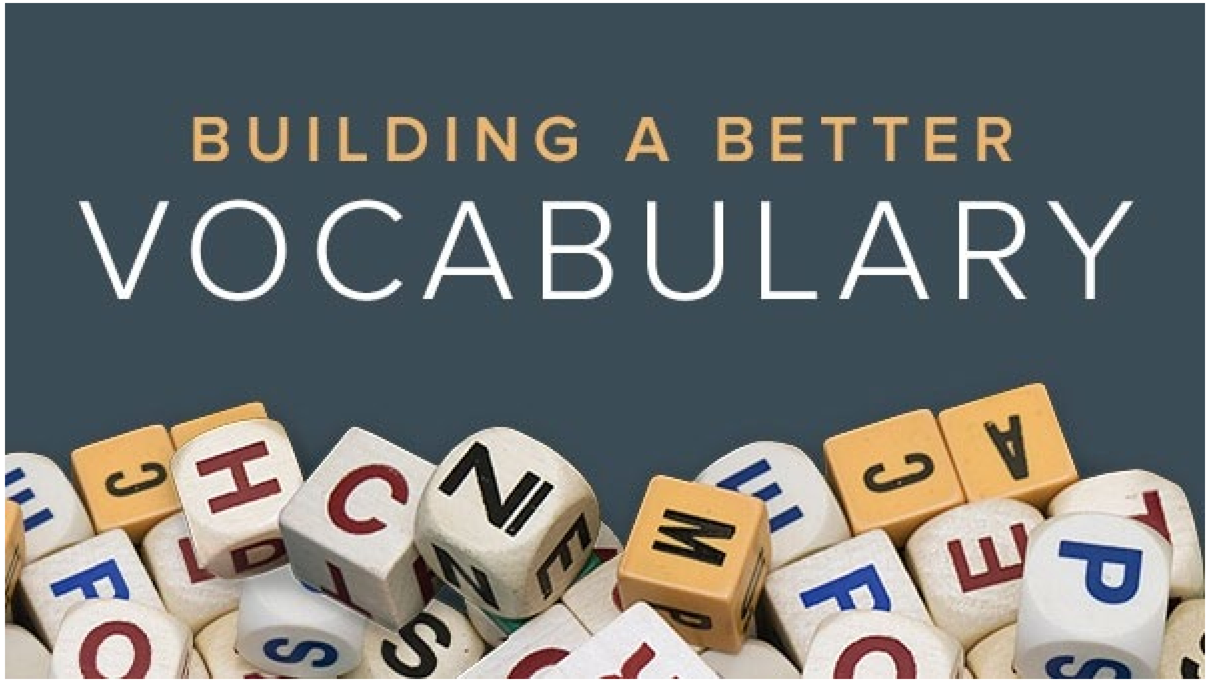
Je vous ai apporté des bonbons
vs.
فلقناه ا = so+said+we+it

예시3. 복합어(ex. 영어, 독일어)

life insurance company employee
vs.
Lebensversicherungsgesellschaftsangestellter



Word Level vs. Character Level



Word Level 모델의 문제점
< 방대한 크기의 단어장을 가져야 한다. >

1) 다양한 복합어의 가능성

2) Transliteration
- 최대한 들리는 대로 유사하게 옮겨적기
- 들리는대로 번역하더라도 어느 정도는 맞아야!

ex. Christopher -> Krystof

3) 비표준어의 사용도 증가
- 인터넷 용어

ex. Gooood vibesss, idk

Character Level 모델의 특징

1. Word Embedding이 Character Embedding으로 이루어질 수 있다.

=> OOV(out of vocabulary) 문제를 해결 할 수 있다! 어떻게?

- 모르는 단어에 대한 embedding을 형성할 수 있다.
- 비슷한 맞춤법이면 비슷한 embedding을 갖는다.

2. 언어에 따라 다양할 수 밖에 없다.

- 딥러닝은 방대한 데이터를 요구하기에, written form이 필요하다.
- 글로 쓰여진 형태의 데이터가 가장 처리하기 쉽고, 구하기도 쉽다!

p.s. 교수 왈, 음소를 기준으로 딥러닝이 시도된 적은 없다.

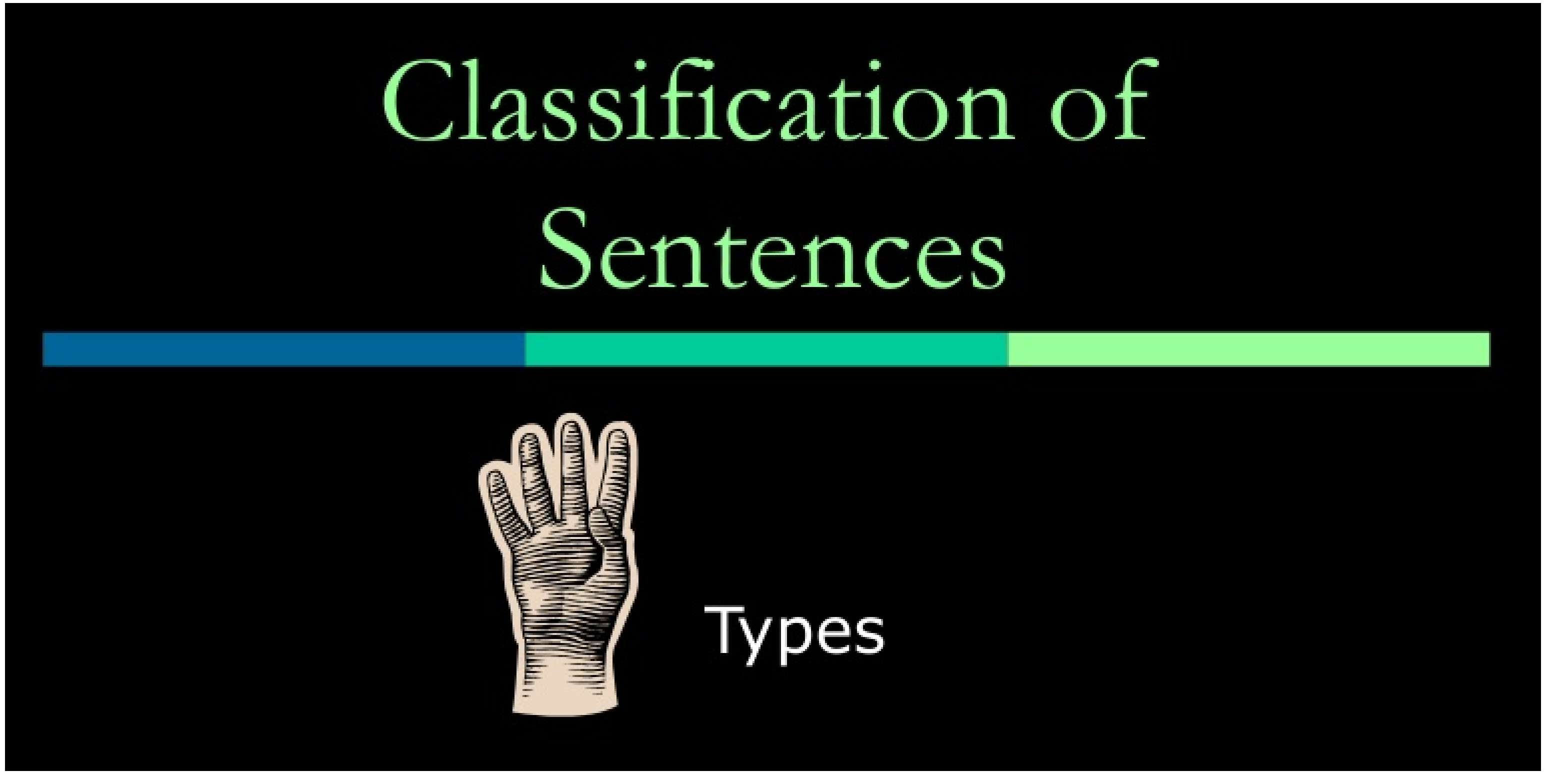
2. Purely Character-Level Model

l 2. Purely Character-Level 모델

Character-level 모델은 크게 두 가지로 나뉘어진다.

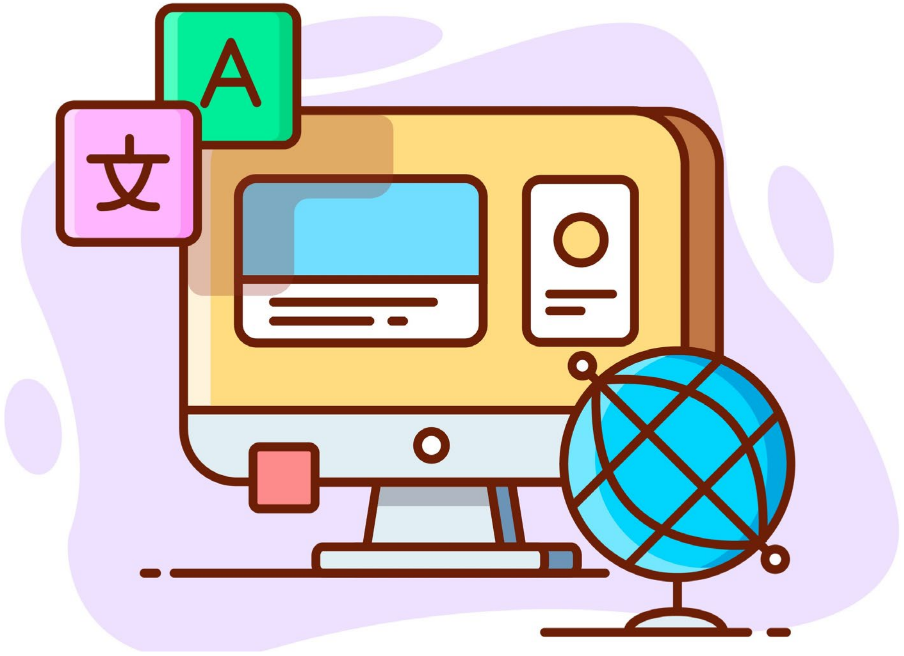
1) 순수하게 character 그 자체만을 쓴 모델
2) Character를 이어붙이는 등으로 활용하여 word level처럼 사용하는 모델

| 2. Purely Character-Level 모델



Deep Convolutional Network 모델에서
Sentence Classification을 하는 데에 좋은 성능을 보였다.
(Conneau, Schwenk, Lecun, Barrault, EACL 2017)

| 2. Purely Character-Level 모델



NMT에서는 초반 2007~2013년에 안 좋은 성과를 보였으나
2015년경부터는 좋은 성과를 내기 시작했다.
(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)

| 2. Purely Character-Level 모델

높을수록 좋은거~

〈 English-Czech WMT 2015 〉
(Workshop on Statistical Machine Translation)

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

하지만 학습하는 데에 3주나 걸리는 치명적인 단점이…

Cf. 체코어는 character level에서 연구하기 좋은 언어이다.
(길고 이상한 복합어가 엄청 많기 때문….)

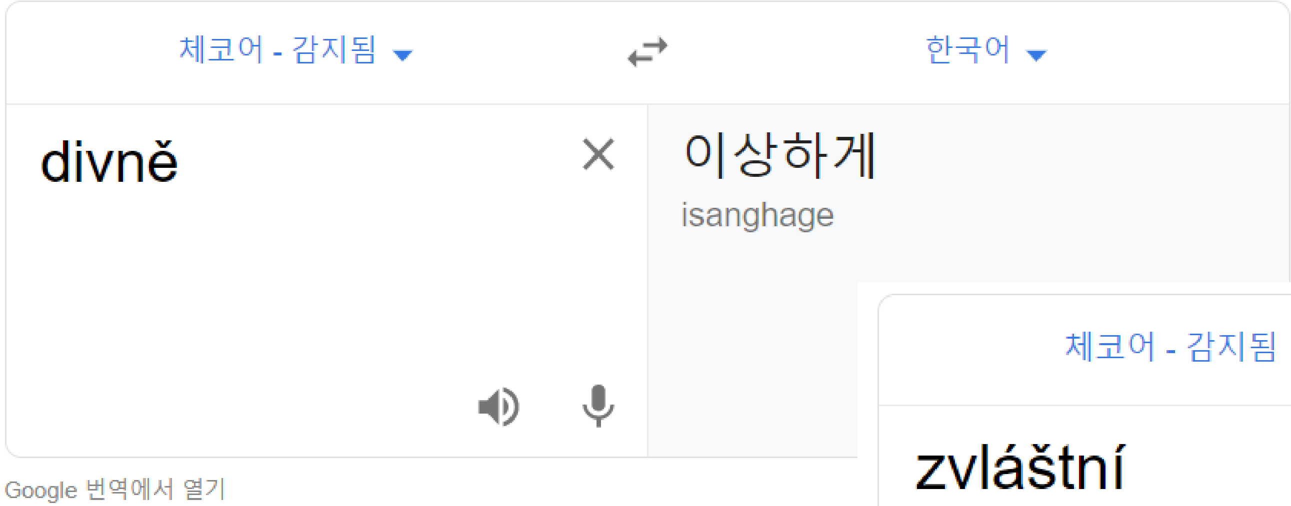
I 2. Purely Character-Level 모델

〈 English-Czech WMT 2015 〉
(Workshop on Statistical Machine Translation)

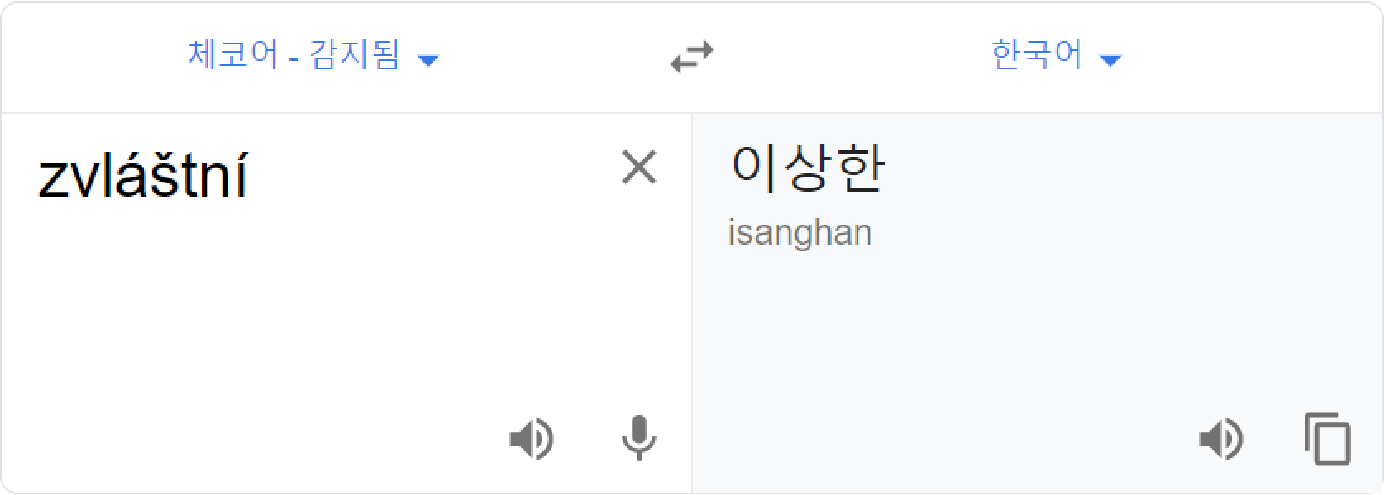
source	Her *11-year-old* daughter , *Shani Bart* , said it felt a little bit *weird*	
정답 human	Její **jedenáctiletá** dcera **Shani Bartová** prozradila , že je to trochu **zvláštní**	
char	Její **jedenáctiletá** dcera , **Shani Bartová** , říkala , že cítí trochu ***divně***	별로 차이 안 남
word	Její <unk> dcera <unk> <unk> řekla , že je to trochu divné	꽤 차이 남 (1)
word	Její **11-year-old** dcera **Shani** **,** řekla , že je to trochu ***divné***	꽤 차이 남 (2)

I 2. Purely Character-Level 모델

〈 English-Czech WMT 2015 〉
(Workshop on Statistical Machine Translation)

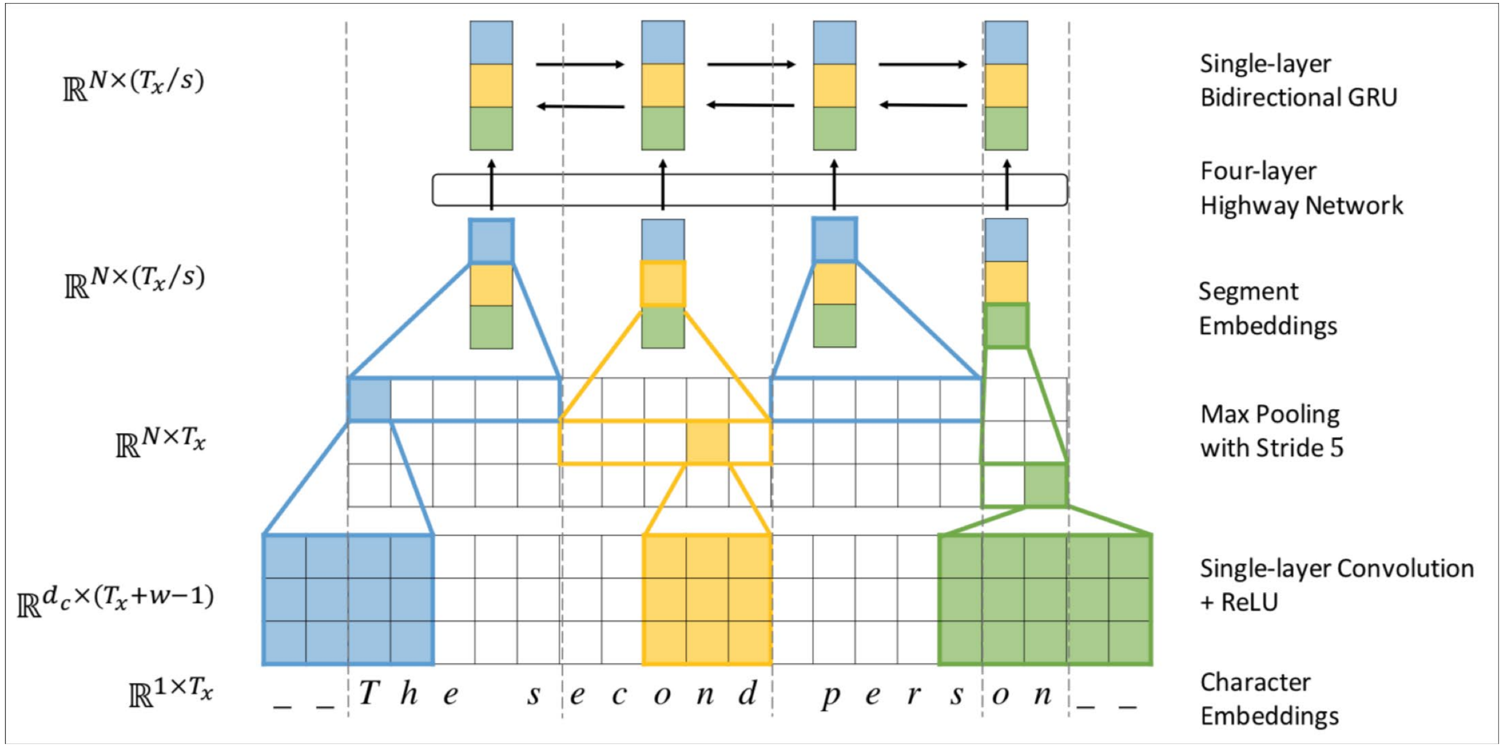


Character-level에서 번역한 거랑,
인간이 직접 번역한 거랑 의미는 거의 같다!



I 2. Purely Character-Level 모델

Fully Character-Level Neural Machine Translation Without Explicit Segmentation

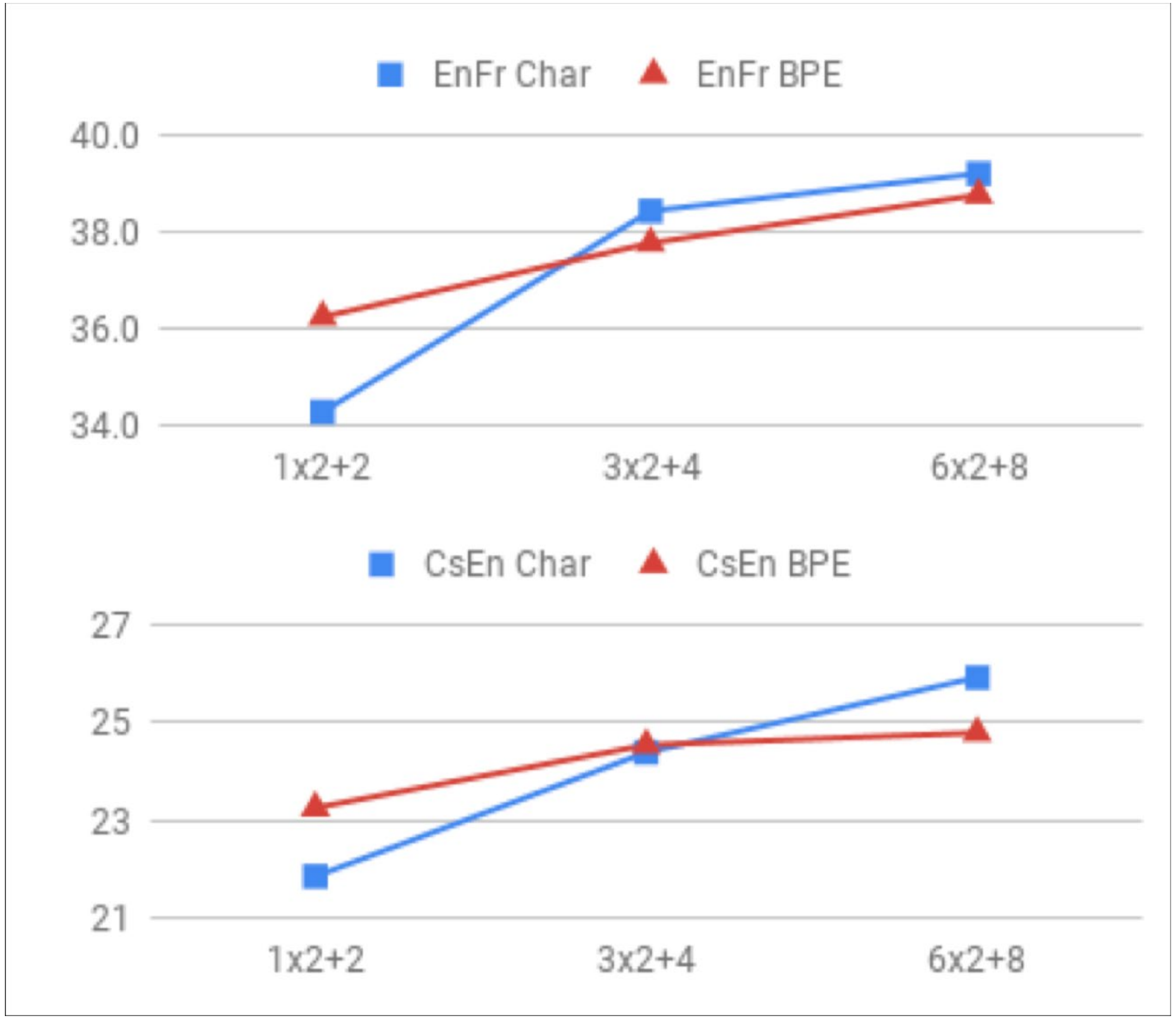


체코어->영어
*Bpe가 뭔지는 뒤에 나와요~

Encoder	Target	BLEU
Bpe	Bpe	20.3
Bpe	Char	22.4
Char	Char	22.5

Encoder & Decoder 모두 **Character-level**일 때 가장 성능이 좋았다.

LSTM seq2seq 모델



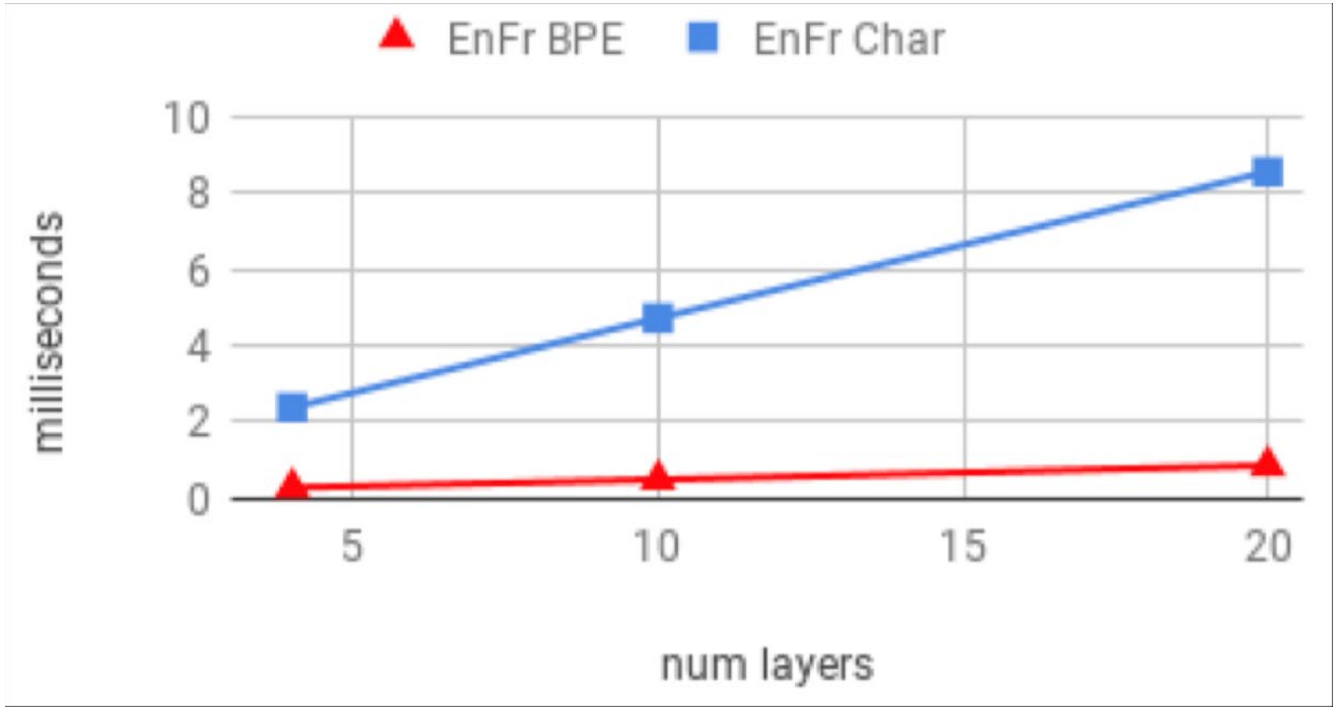
En: English
Fr: French
Cs: Czech

<x축의 의미>
3x2 + 4
Bidirectional LSTM Encoder 3층
LSTM Decoder 4층

< 그래프 해석 >
간단한 모델을 쓸거면 word-level
복잡한 모델을 쓸거면 char-level
특히, 체코어처럼 형태적으로 복잡한 언어의 경우!
Char-level이 유리하다

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

LSTM seq2seq 모델



단, char-level 모델의 경우
학습하는 데에 있어서 시간이
오래 걸린다는 단점이 있다.

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

3. Sub-word models: two trends

Bottom-up Clustering (character -> vocabulary)

Dictionary

5 l o w
2 l o w e r
6 n e w e s t
3 w i d e s t

Vocabulary

l, o, w, e, r, n, w, s, t, i, d

characters

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, **es**

(e,s) -> es [freq : 9]

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, **est**

(es, t) -> est [freq : 9]

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, est, **lo**

(l, o) -> lo [freq : 7]

I 3-1. Byte Pair Encoding

Dictionary

5 l o w
2 l o w e r
6 n e w e s t
3 w i d e s t

10번 반복 →

l, o, w, e, r, n, w, s, t, i, d, es, est, lo, low, ne, new, newest, wi, wid, widest

Test

Lowest → **Low + est**

I 3-1. Byte Pair Encoding

Dictionary

5 l o w
2 l o w e r
6 n e w e s t
3 w i d e s t

10번 반복

l, o, w, e, r, n, w, s, t, i, d, es, est, lo, low, ne, new, newest, wi, wid, widest

Test

Lowest

Low + est

더이상 OOV 아니다

| 3-2. Wordpiece/Sentencepiece model

Wordpiece model

BPE

variant

단순 n-gram count(x),
greedy approximation to maximizing language model log likelihood

Sentencepiece model

HOW TO:

Jet makers feud over seat width with big orders at stake

↓

_J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

모든 단어의 맨 앞에 _를 붙이고, 단어를 subword로 통계에 기반하여 띄어쓰기로 분리

Ex) Jet -> _J et : 띄어쓰기 추가되어 subword 구분하는 구분자 역할

문장 복원 방법 : 모든 띄어쓰기 제거 & _를 띄어쓰기로 바꾸면 된다

CODE:

```
lines = [
  "I didn't at all think of it this way.",
  "I have waited a long time for someone to film"
]
for line in lines:
  print(line)
  print(sp.encode_as_pieces(line))
  print(sp.encode_as_ids(line))
  print()
```

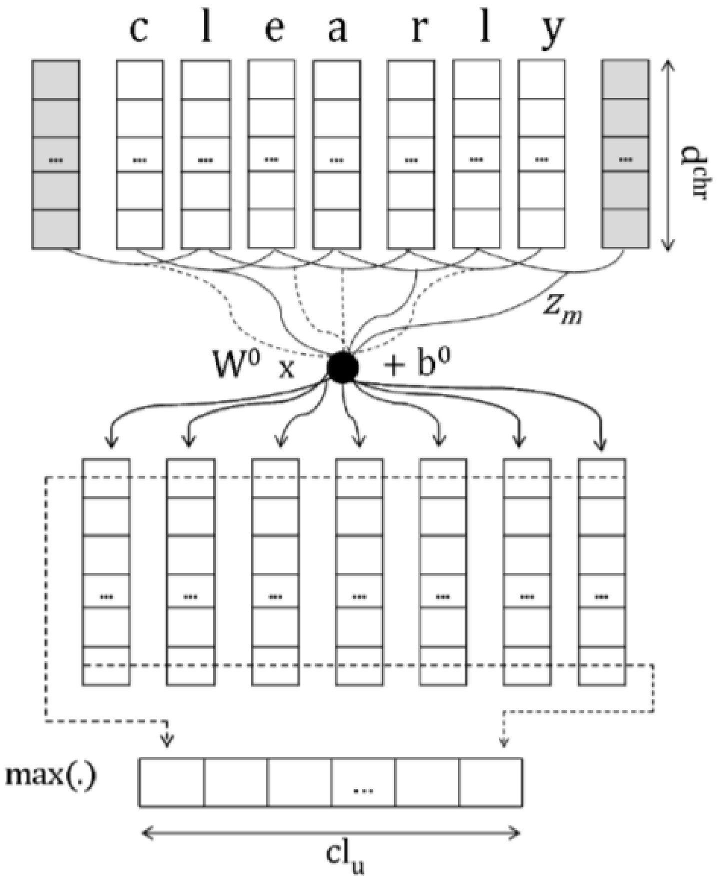
```
I didn't at all think of it this way.
['_I', '_didn', "'", 't', '_at', '_all', '_think', '_of', '_it', '_thi
s', '_way', '.']
[41, 623, 4950, 4926, 138, 169, 378, 30, 58, 73, 413, 4945]

I have waited a long time for someone to film
['_I', '_have', '_wa', 'ited', '_a', '_long', '_time', '_for', '_someon
e', '_to', '_film']
[41, 141, 1364, 1120, 4, 666, 285, 92, 1078, 33, 91]
```

https://wikidocs.net/86657

4. Character-level to build word-level

Character-based LSTM

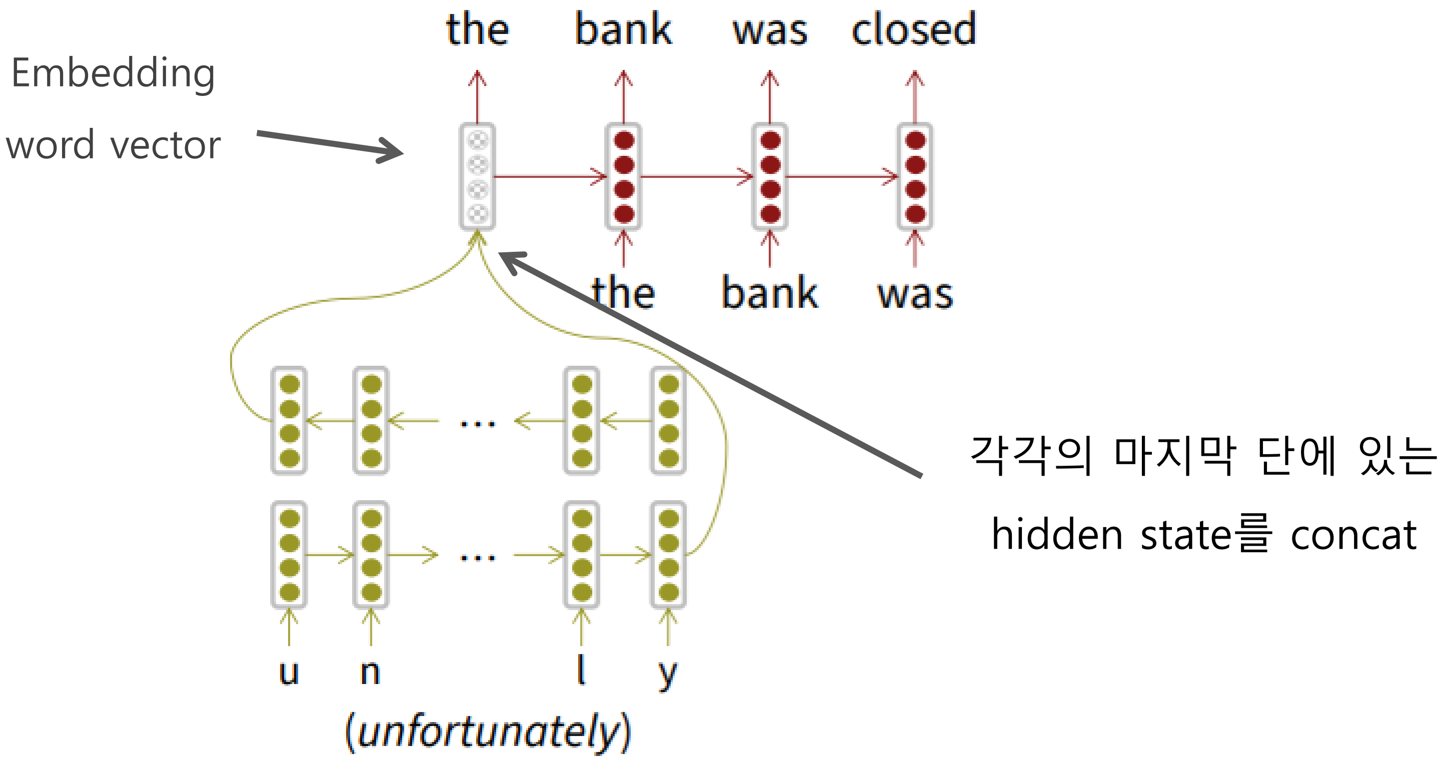


- Convolution over characters to generate word embeddings

- Fixed window of word embeddings used for PoS tagging

ㅣ4. Character-level to build word-level

Character-based LSTM



Character-Aware Neural Language Models

- Yoon Kim, Yacine Jernite, David Sontag, Alexander M.Rush. 2015

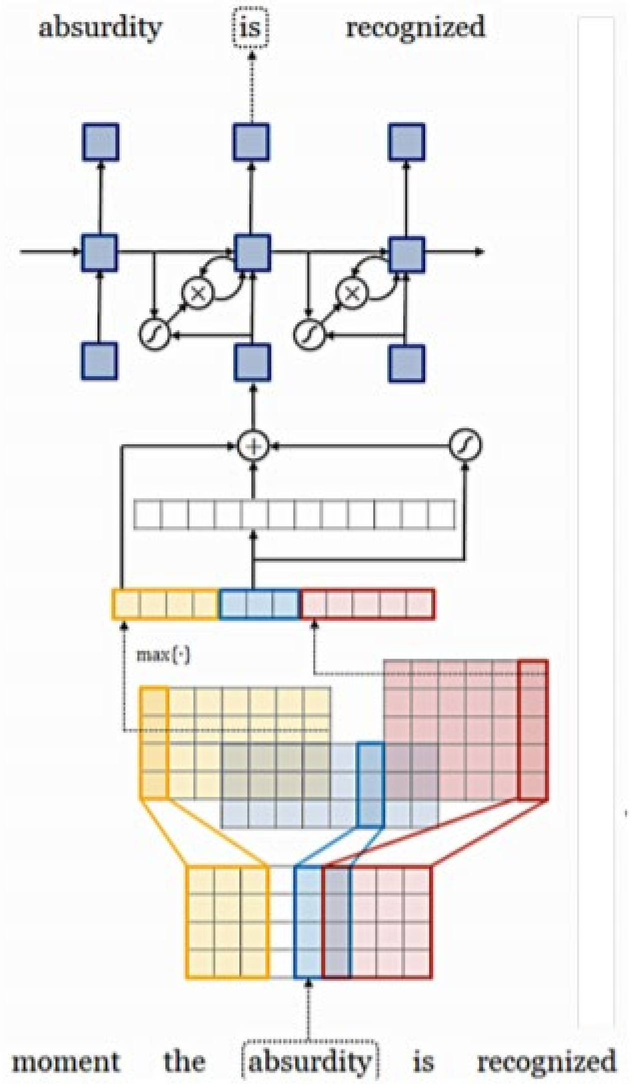
- 다양한 언어에 대해서 좀 더 효율적이고 강력한 언어모델을 구성하기 위해 더 복잡하고 정교한 접근법을 취함

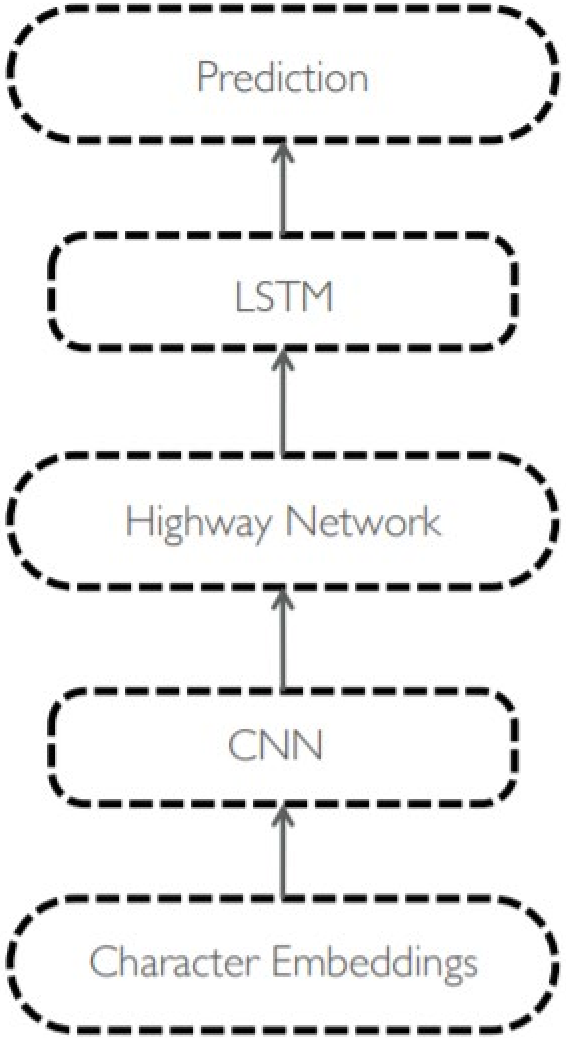
- 적은 매개변수로도 비슷한 결과 도출

- 사전모델을 사용해서 효율적으로 단어 처리 (rare-word problem 해결)

I 4. Character-level to build word-level

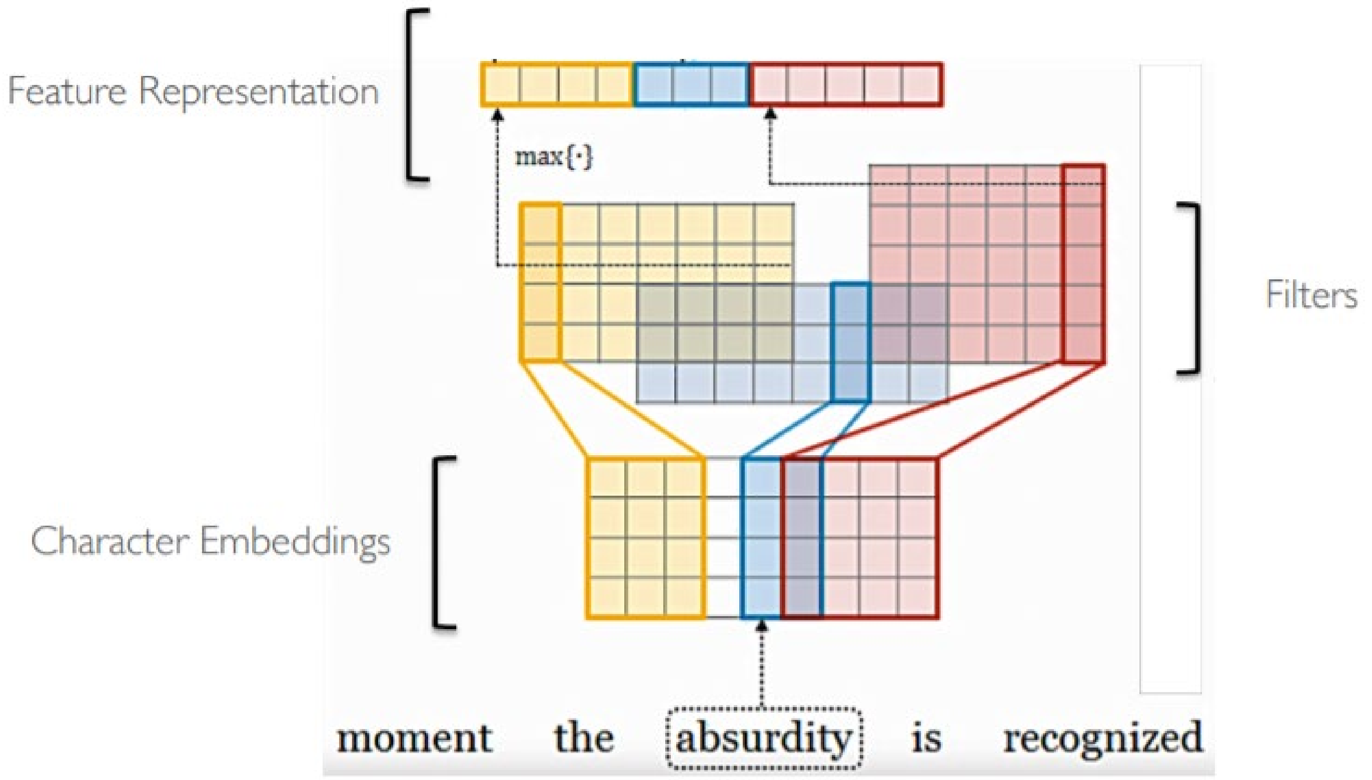
Technical Approach





l 4. Character-level to build word-level

Convolutional Layer



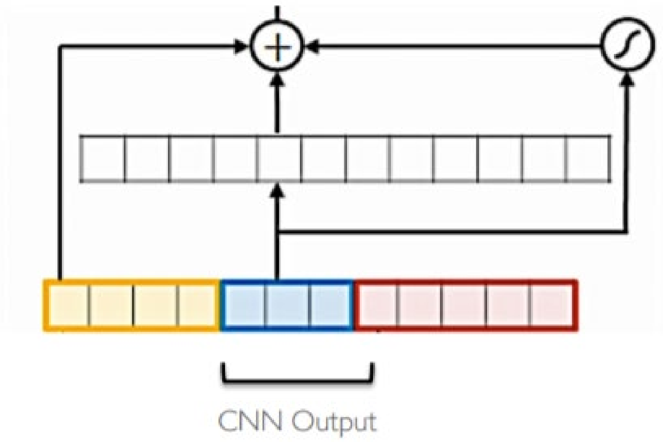
- Convolutions over character-level inputs
- Max-over-time pooling (effectively n-gram selection)
- Various filter size / not using word embedding

Highway Network

$$\mathbf{t} = \sigma(\mathbf{W}_T \mathbf{y} + \mathbf{b}_T)$$

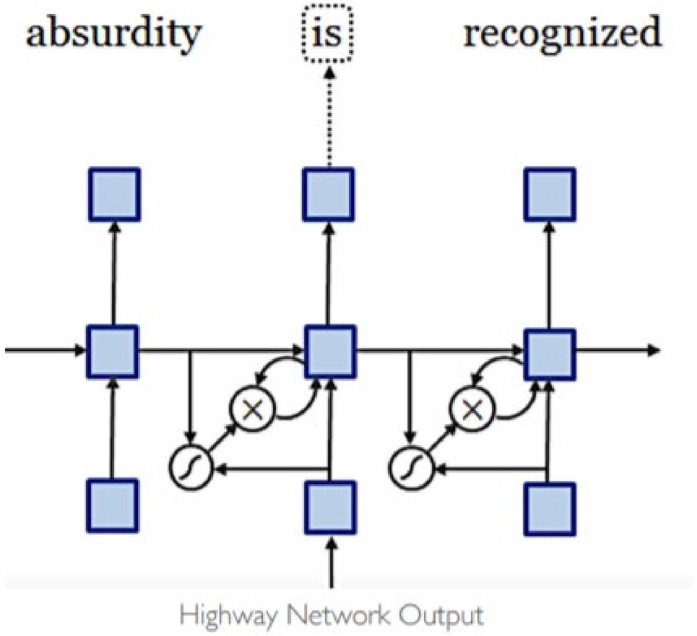
Output $$\mathbf{z} = \mathbf{t} \odot g(\mathbf{W}_H \mathbf{y} + \mathbf{b}_H) + (\mathbf{1} - \mathbf{t}) \odot \mathbf{y}$$

Transform Gate　Input　Carry Gate



- Model n-gram interactions
- Apply transformation while carrying over original information 'y'
- Functions akin to an LSTM memory cell

LSTM Network



- Hierarchical Softmax to handle large output vocabulary
- Trained with truncated backprop through time

| 4. Character-level to build word-level

Results

		DATA-S					
		Cs	De	Es	Fr	Ru	Ar
Botha	KN-4	545	366	241	274	396	323
	MLBL	465	296	200	225	304	–
Small	Word	503	305	212	229	352	216
	Morph	414	278	197	216	290	230
	Char	401	260	182	189	278	196
Large	Word	493	286	200	222	357	172
	Morph	398	263	177	196	271	**148**
	Char	**371**	**239**	**165**	**184**	**261**	**148**

		DATA-L					
		Cs	De	Es	Fr	Ru	En
Botha	KN-4	862	463	219	243	390	291
	MLBL	643	404	203	227	**300**	273
Small	Word	701	347	186	202	353	236
	Morph	615	331	189	209	331	233
	Char	**578**	**305**	**169**	**190**	313	**216**

적은 매개변수로도 비슷한 결과 도출

	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

| 4. Character-level to build word-level

Results

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

- **Word - 고유명사인 사람이름에 대해서는 유사한 사람이름을 순서대로**
- **Char(before highway) - 고유명사에 대해서는 잘못된 결과**
- **Char(after highway) - 문자단위라도 highway를 거치면 사람이름과 유사한 결과**

Results

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*

- Word - 처리불가. 그래서 보통은 원본 그대로 copy
- Char(before highway) - 처리가능하긴 하나 결과가 좋진 않음
- Char(after highway) - 가장 유사한 결과

Results

- Paper questioned the necessity of using word embeddings as inputs for neural language modeling.

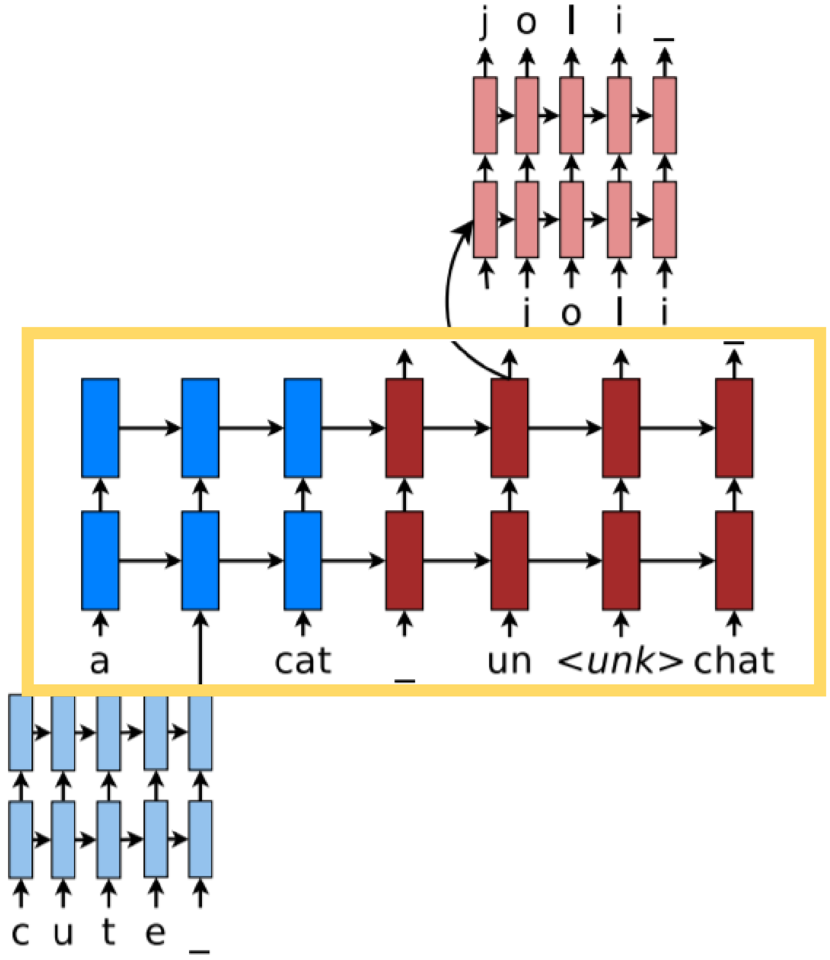
- 'CNNs + Highway Network over characters' is better

- Key thinking

⇒ You can compose "building blocks" to obtain nuanced and powerful models!

Hybrid NMT

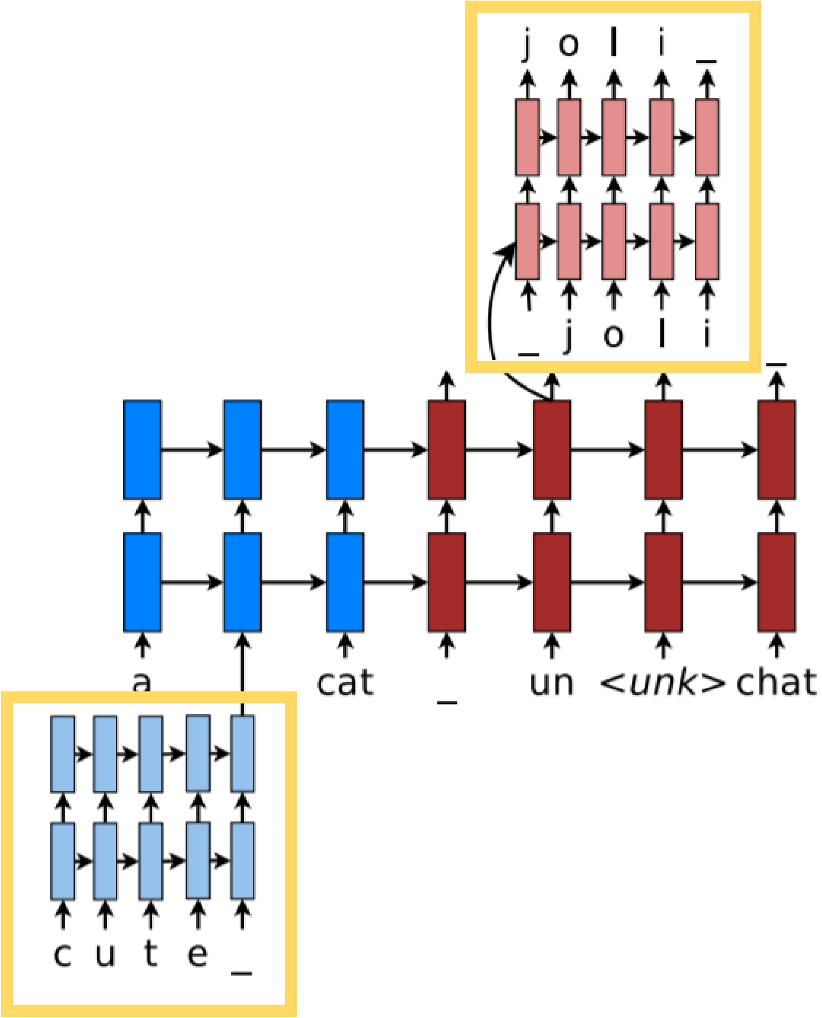
2-stage Decoding



메인 모델은 Word-level (4 layers)

Hybrid NMT

2-stage Decoding



<unknown> 토큰에 대해서는

Character-level로 다시 수행 (2 layers)

English-Czech Results

WMT'15 data (12M sentence pairs)

Systems	BLEU
Winning WMT'15 (Bojar & Tamchyna, 2015)	18.8
Word-level NMT (Jean et al., 2015)	18.3
Hybrid NMT (Luong & Manning, 2016)*	**20.7**

I 4. Character-level to build word-level

English-Czech Results

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!

| 4. Character-level to build word-level

English-Czech Results

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** .

- *Char*-based: wrong name translation

I 4. Character-level to build word-level

English-Czech Results

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** .

- *Word*-based: incorrect alignment

| 4. Character-level to build word-level

English-Czech Results

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** .

- *Char*-based & hybrid: correct translation of **diagnóze**

ㅣ4. Character-level to build word-level

English-Czech Results

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

I 4. Character-level to build word-level

English-Czech Results

source	Her *11-year-old* daughter , *Shani Bart* , said it felt a little bit *weird*
human	Její $\boxed{\textbf{jedenáctiletá}}$ 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 $\boxed{\textbf{jedenáctiletá}}$ dcera , **Graham** ***Bart*** , řekla , že cítí trochu ***divný***

- Hybrid: correct, ***11-year-old*** – **jedenáctiletá**
- Wrong: **Shani Bartová**

5. Chars for word embeddings

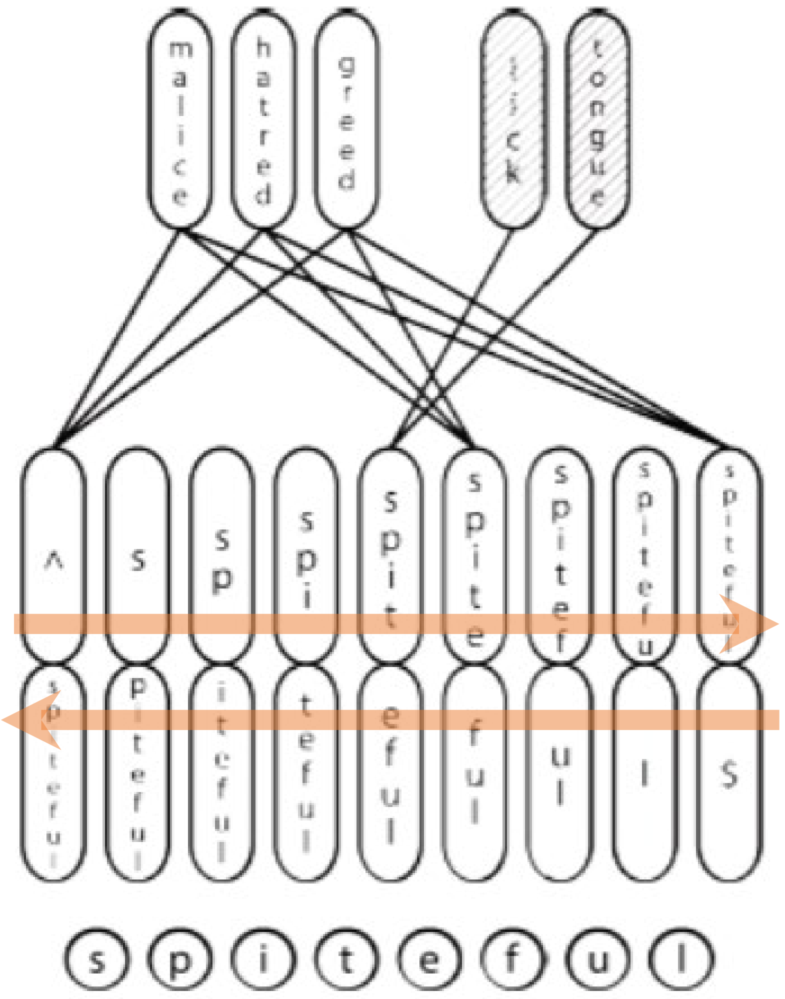
| 5. Chars for word embeddings

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

ex) spit → lick, tongue

spite, spiteful → malice, hatred, greed



FastText embeddings

Enriching Word Vectors with Subword Information

Bojanowski, Grave, Joulin and Mikolov. FAIR. 2016.

Aim : a next generation efficient word2vec-like word representation library.

but <u>better for rare words and languages with lots of morphology</u>

→ An extension of the w2v skip-gram model <u>with character n-grams</u>

ex) where = <wh, whe, her, ere, re>, <where> : subwords for embedding

FastText embeddings

Represent word as sum of the subwords.

Word in context score is $s(w,c) = \sum_{g \in G(w)} \mathbf{z}_g^{\mathrm{T}} \mathbf{v}_c$

'hashing trick' 차원을 축소해서 원래 유사도를 구하는 방식

THE END