Deep Neural Networks for Natural Language Processing (AI6127)

JUNG-JAE KIM

LECTURE 10: SUBWORD MODELS

Lecture Plan

- A tiny bit of linguistics
- Purely character-level models
- Subword-models: Byte Pair Encoding and friends
- Hybrid character and word level models
- fastText

https://www.youtube.com/watch?v=EN0wXFN0GO0

1. Human language sounds: Phonetics and phonology

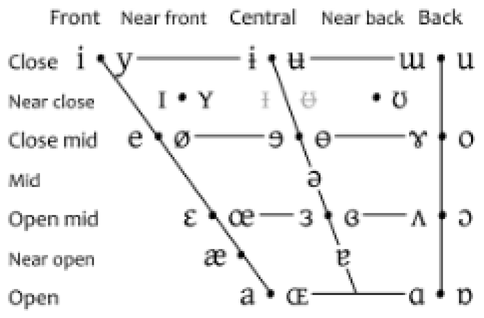
- Phonetics is the sound stream – uncontroversial “physics”
- Phonology posits a small set or sets of distinctive, categorical units: phonemes (significant spoken sounds) or distinctive features
  - A perhaps universal typology but language-particular realization
  - Best evidence of categorical perception comes from phonology
    - Within phoneme differences shrink; between phoneme magnified

CONSONANTS (PULMONIC) © 2005 IPA

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k ɡ	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap		ⱱ		ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Where symbols appear in pairs, the one to the right represents a voiced consonant. Shaded areas denote articulations judged impossible.

vowels



Vowels at right & left of bullets are rounded & unrounded.

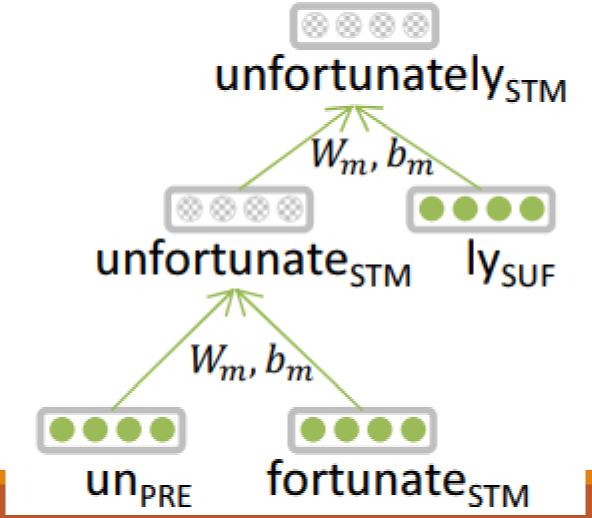
caught / kɔt /
cot / kɒt /

Morphology: Parts of words

- Traditionally, we have morphemes as smallest **semantic** unit
  - [[un [[fortun(e)]$_{ROOT}$ ate]$_{STEM}$] $_{STEM}$ ly]$_{WORD}$

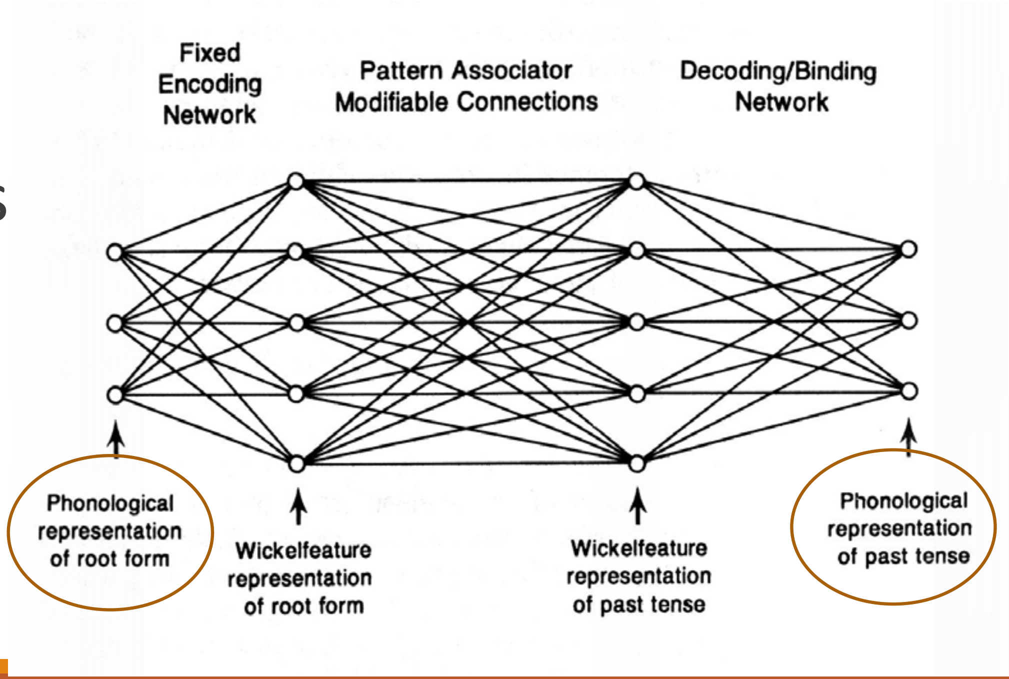
A root is a form which is not further analysable
'fortunate' is the stem of 'unfortunate'

- Deep learning: Morphology little studied; one attempt with recursive neural networks is (Luong, Socher, & Manning 2013)



Morphology

- An easy alternative is to work with character n-grams
  - Wickelphones (Rumelhart & McClelland 1986)
  - Microsoft's DSSM (Huang, He, Gao, Deng, Acero, & Hect 2013)
- Related idea to use of a convolutional layer
- Can give many of the benefits of morphemes more easily??

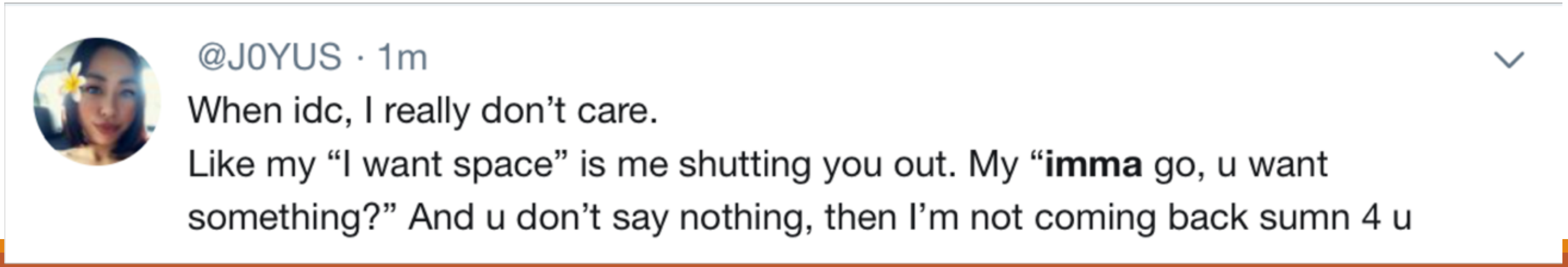


Words in writing systems

- Writing systems vary in how they represent words – or don’t
- 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? (have form of affixes, but distribution of function words; e.g. it’**s**, we’**ve**)
    - Separated **Je vous ai apporté** des bonbons
    - Joined فقلناها = ف+ قال+ نا+ ها = so+said+we+it
  - Compounds?
    - Separated life insurance company employee
    - Joined Lebensversicherungsgesellschaftsangestellter

Models below the word level

- Need to handle large, open vocabulary
  - Rich morphology: nejneobhospodařovávatelnějšímu Czech
    (“to the worst farmable one”)
  - Transliteration: Christopher ↦ Kryštof
  - Informal spelling:



Character-Level Models

- 1. Word embeddings can be composed from character embeddings
  - Generates embeddings for unknown words
  - Similar spellings share similar embeddings
  - Solves OOV problem
- 2. Connected language can be processed as characters
- Both methods have proven to work very successfully!
  - Somewhat surprisingly – traditionally, phonemes/letters weren’t a semantic unit – but DL models compose groups

Below the word: Writing systems

- Most deep learning NLP work begins with language in its written form – it’s the easily processed, found data

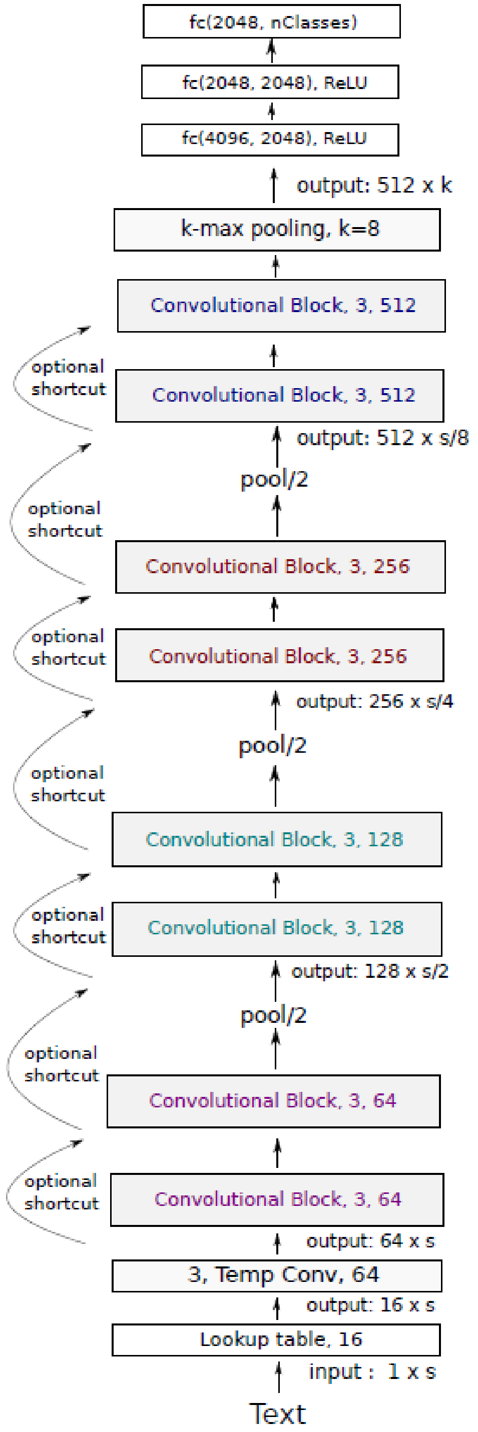
- But human language writing systems aren’t one thing!

Below the word: Writing systems

- Phonemic (maybe digraphs) jiyawu ngabulu Wambaya
  - graphemes (written symbols) correspond to phonemes
- Fossilized phonemic thorough failure English
- Syllabic/moraic ᑐᖑᔪᐊᖓᔪᖅ Inuktitut
  - characters represent syllables and are combined to indicate morphemes
  - Syllable: a sequence of sounds/phonemes with at least one vowel
- Ideographic 去年太空船二号坠毁 Chinese
  - ‘ideogram’ symbols represent elements of language
- Combination of the above インド洋の島 Japanese

2. Purely character-level models

- Strong results via a deep convolutional stack
  - Very Deep Convolutional Networks for Text Classification
  - Conneau, Schwenk, Lecun, Barrault. EACL 2017



Purely character-level NMT models

- Initially, **unsatisfactory** performance
  - (Vilar et al., 2007; Neubig et al., 2013)
- Subword-level encoder + **Character-level decoder** (w/o segmentation)
  - (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)

English-Czech WMT 2015 Results

- Luong and Manning tested as a baseline a pure character-level seq2seq (LSTM) NMT system
- It worked well against word-level baseline
- But it was ssllooooww
  - 3 weeks to train … not that fast at runtime

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

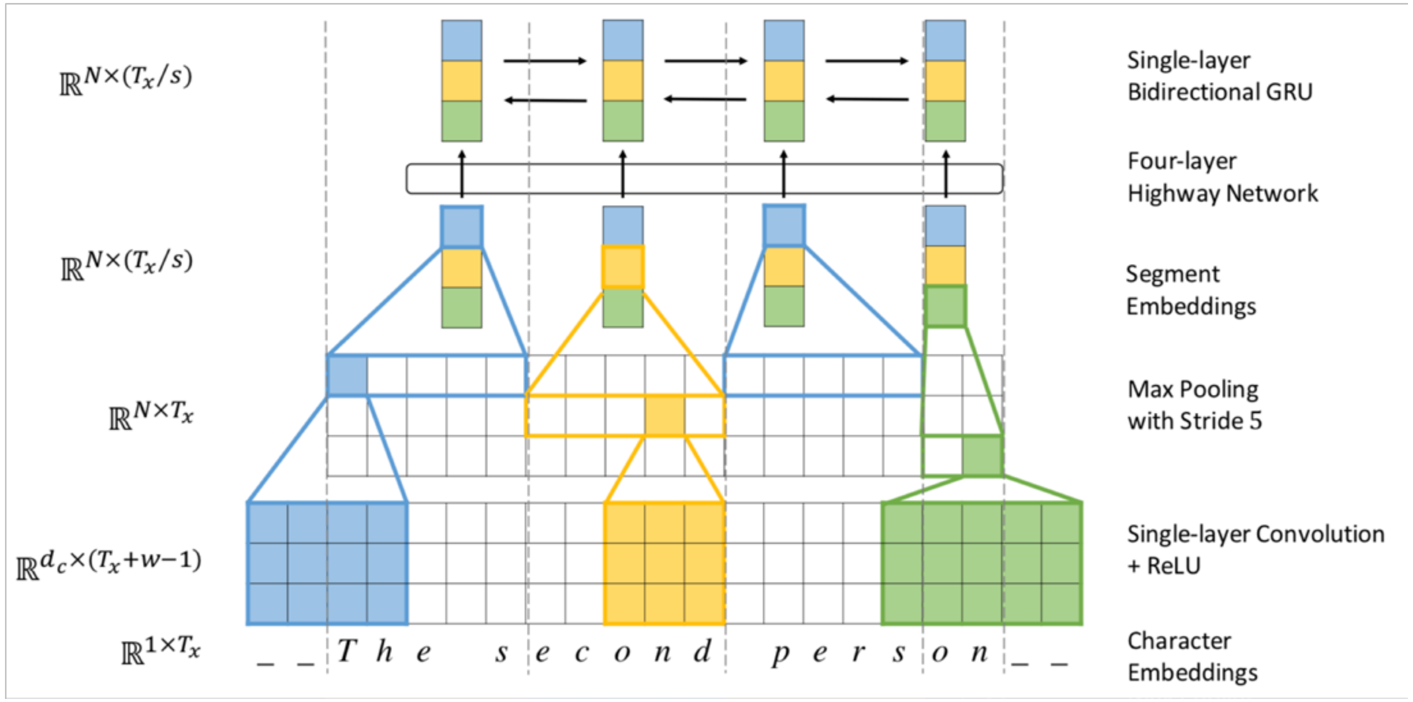
English-Czech WMT 2015 Example

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

Fully Character-Level Neural Machine Translation without Explicit Segmentation

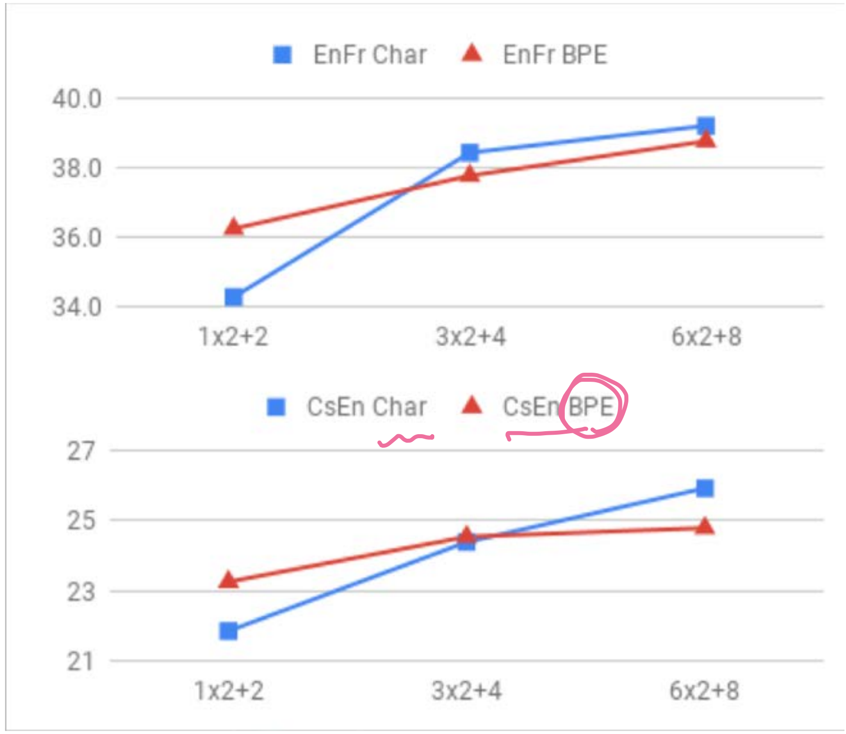
- Jason Lee, Kyunghyun Cho, Thomas Hoffmann. 2017. Encoder as below; decoder is a char-level GRU



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

Stronger character results with depth in LSTM seq2seq model

- Revisiting Character-Based Neural Machine Translation with Capacity and Compression. 2018. Cherry, Foster, Bapna, Firat, Macherey, Google AI
  - **X-axis: E.g. 1x2+2 indicates 1 BiLSTM encoder layer and 2 LSTM decoder layers**
  - **Y-axis: bleu scores**



3. Sub-word models: two trends

- **Same** architecture as for word-level model:
  - But use smaller units: “word pieces”
  - [Sennrich, Haddow, Birch, ACL’16a], [Chung, Cho, Bengio, ACL’16].

- **Hybrid** architectures:
  - Main model has *words*; something else for *characters*
  - [Costa-Jussà & Fonollosa, ACL’16], [Luong & Manning, ACL’16].

Byte Pair Encoding

- Originally a compression algorithm:
  - Most frequent byte pair ↦ a new byte.

Replace bytes with character ngrams

(though, actually, some people have done interesting things with bytes)

  - Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. ACL 2016.
    - https://arxiv.org/abs/1508.07909
    - https://github.com/rsennrich/subword-nmt
    - https://github.com/EdinburghNLP/nematus

Byte Pair Encoding

A word segmentation algorithm: Though done as bottom up clustering

- Start with a unigram vocabulary of all (Unicode) characters in data
- Most frequent ngram pairs $\mapsto$ a new ngram

Byte Pair Encoding

A word segmentation algorithm:

- Start with a unigram vocabulary of all (Unicode) characters in data
- Most frequent ngram pairs ↦ a new ngram

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

Start with all characters in vocab

(Example from Sennrich)

Byte Pair Encoding

A word segmentation algorithm:

- Start with a unigram vocabulary of all (Unicode) characters in data
- Most frequent ngram pairs ↦ a new ngram

Dictionary

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

Vocabulary

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

Add a pair (e, s) with freq 9

(Example from Sennrich)

Byte Pair Encoding

A word segmentation algorithm:

- Start with a unigram vocabulary of all (Unicode) characters in data
- Most frequent ngram pairs ↦ a new ngram

Dictionary

5	l o w
2	l o w e r
6	n e w **est**
3	w i d **est**

Vocabulary

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

Add a pair (es, t) with freq 9

(Example from Sennrich)

Byte Pair Encoding

A word segmentation algorithm:

- Start with a unigram vocabulary of all (Unicode) characters in data
- Most frequent ngram pairs ↦ a new ngram

Dictionary

5	**lo** w
2	**lo** w e r
6	n e w est
3	w i d est

Vocabulary

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

Add a pair (l, o) with freq 7

(Example from Sennrich)

Byte Pair Encoding

- Have a target vocabulary size and stop when you reach it
- Do deterministic longest piece segmentation of words
- Segmentation is only within words identified by some prior tokenizer (commonly Moses tokenizer for MT)
- Automatically decides vocab for system
  - No longer strongly "word" based in conventional way

Top places in WMT 2016!
Still widely used in WMT 2018

https://github.com/rsennrich/nematus

Wordpiece/Sentencepiece model

- Google NMT (GNMT) uses a variant of this
  - V1: wordpiece model
  - V2: sentencepiece model
- Rather than char *n*-gram count, uses a greedy approximation to maximizing language model log likelihood to choose the pieces
  - Add *n*-gram that maximally reduces perplexity

Wordpiece/Sentencepiece model

- Wordpiece model tokenizes inside words
  - Issue: original input and tokenized sequence are NOT reversibly convertible
    - E.g. “World.” vs “World .”
- Sentencepiece model works from raw text
  - Treats raw text just as a sequence of Unicode characters
  - Whitespace is handled as normal symbol
  - You can reverse things at end by joining pieces

  - https://github.com/google/sentencepiece
  - https://arxiv.org/pdf/1804.10959.pdf

Wordpiece/Sentencepiece model

- BERT uses a variant of the wordpiece model
  - (Relatively) common words are in the vocabulary:
    - at, fairfax, 1910s
  - Other words are built from wordpieces:
    - hypatia = h ##yp ##ati ##a

Have word vectors for four word pieces.

- If you're using BERT in an otherwise word based model, you have to deal with this

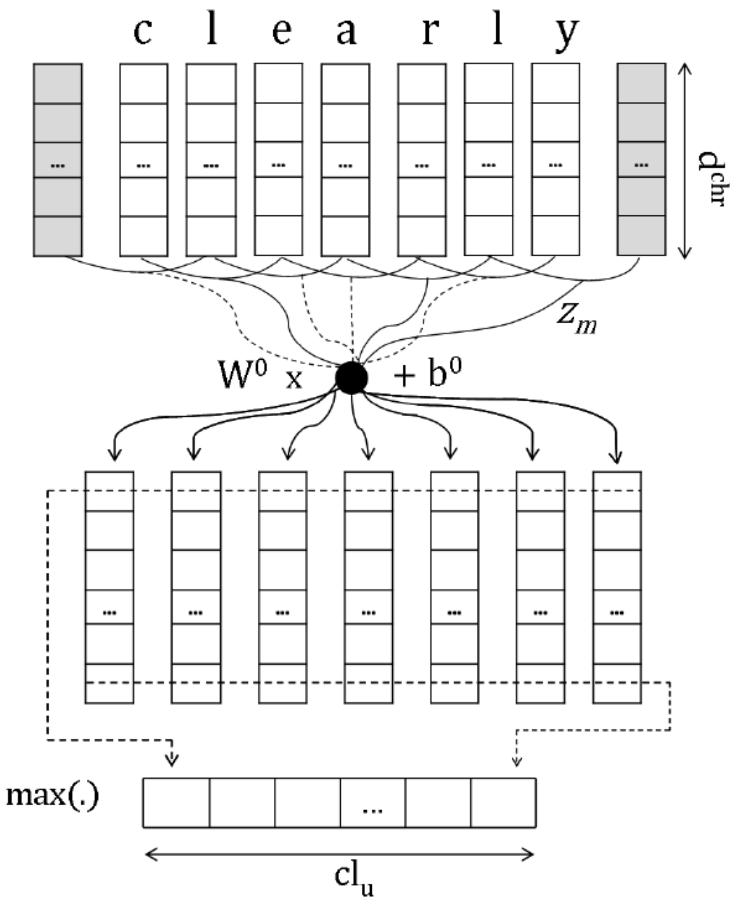
The simplest and quite common way is: averagine these four.

Also can ConvNet & maxpool / LSTM

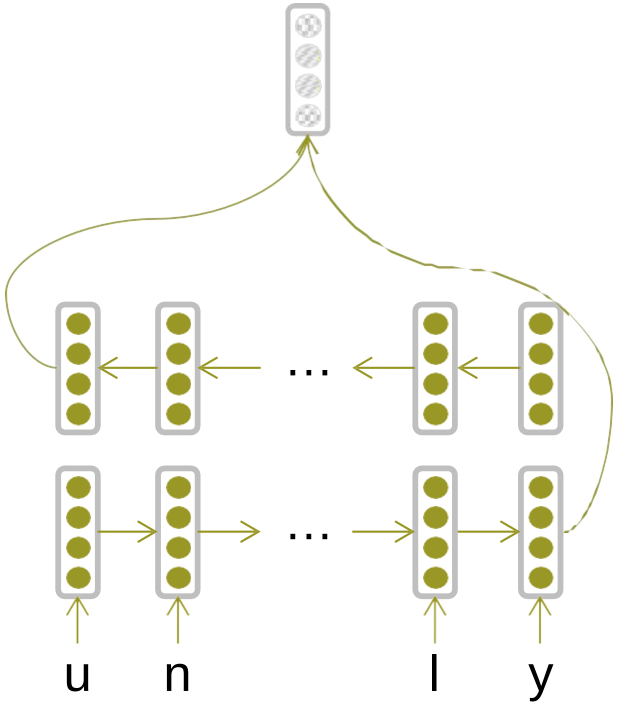
4. Character-level to build word-level

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 to build word representations

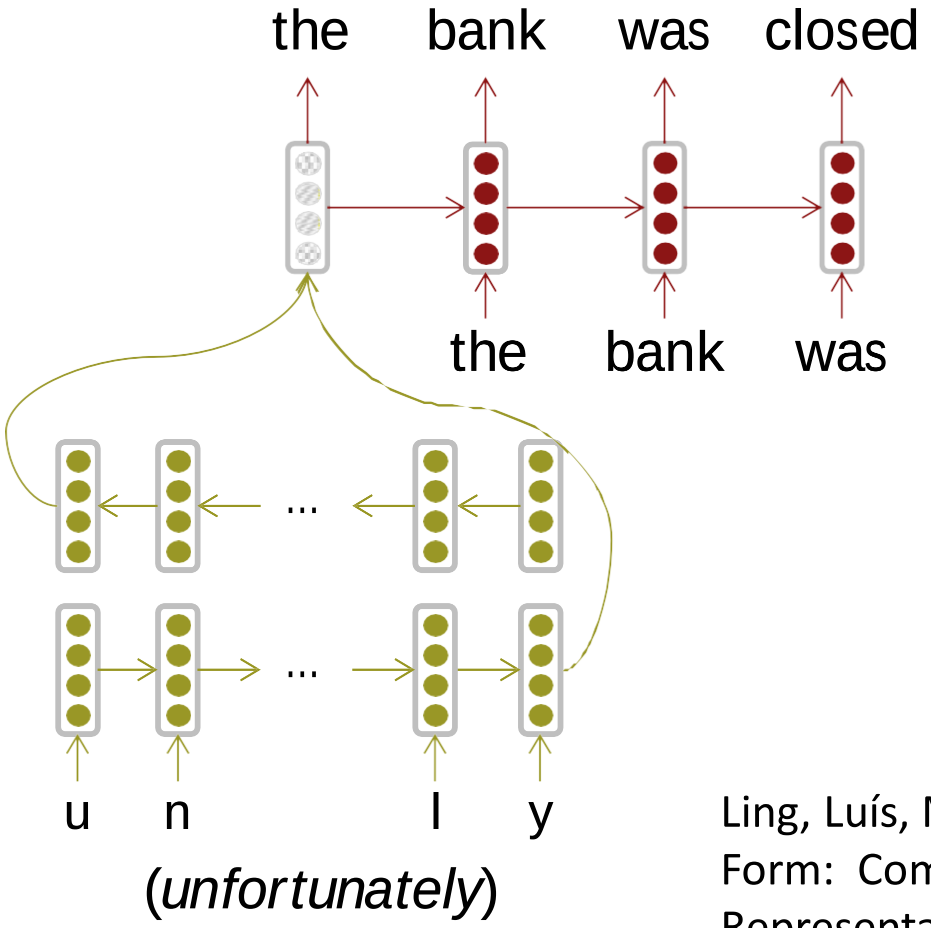


(unfortunately)

Bi-LSTM builds word representations

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

Character-based LSTM



Recurrent Language Model

Bi-LSTM builds word representations

Use 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-Aware Neural Language Models

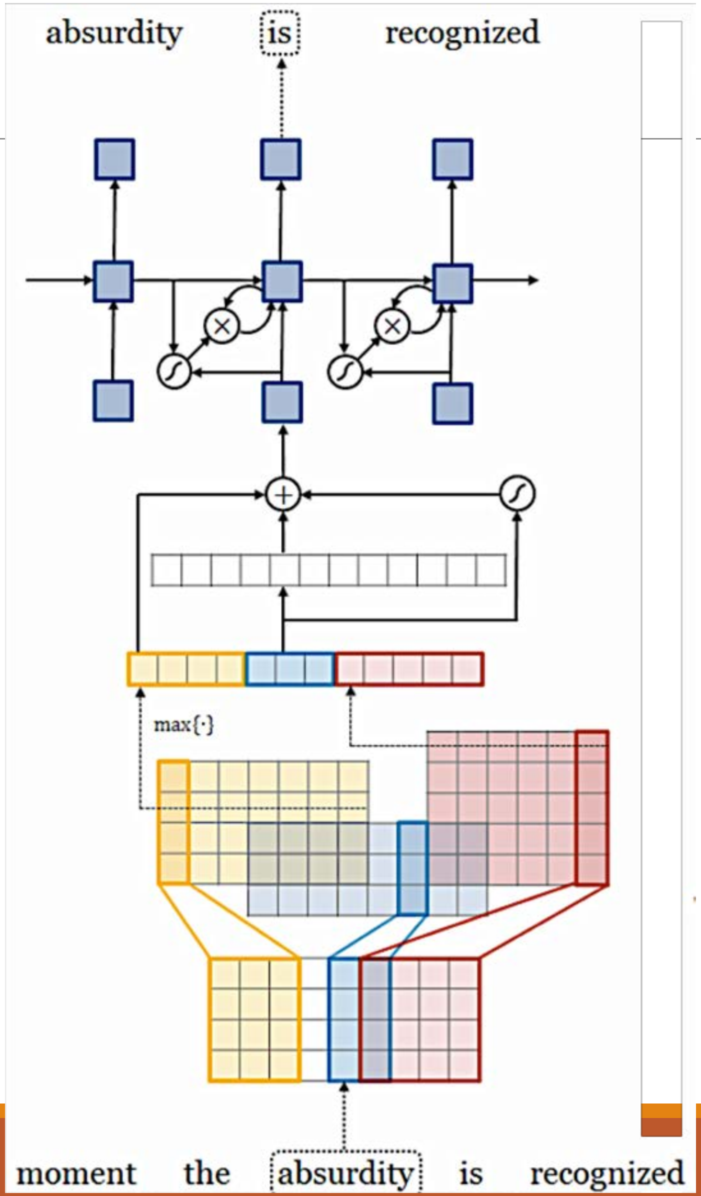
Yoon Kim, Yacine Jernite, David Sontag, Alexander M. Rush. 2015

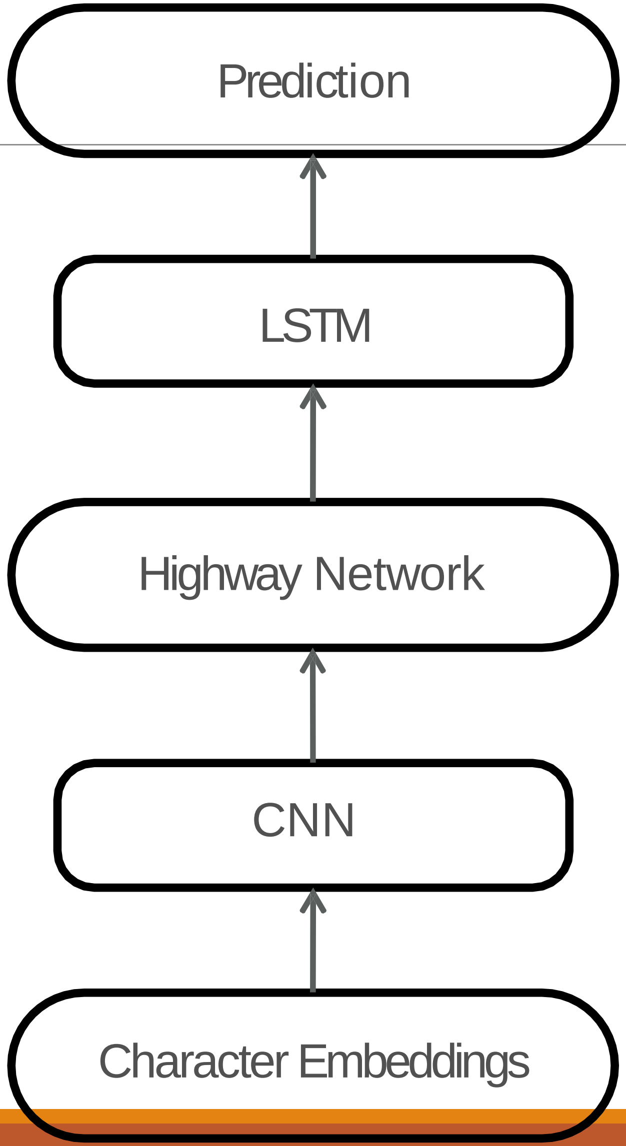
A more complex/sophisticated approach

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.

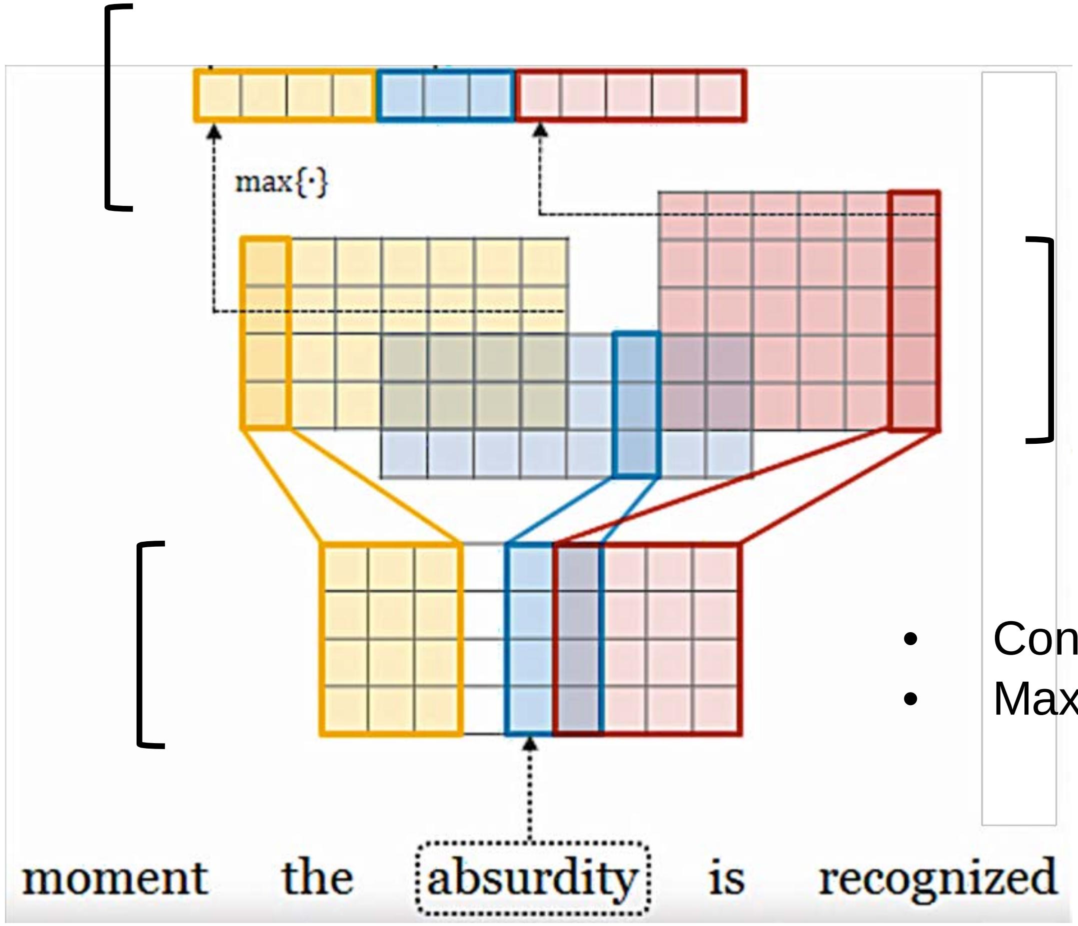
Technical Approach





Convolutional Layer

Feature Representation



Filters

Character Embeddings

- Convolutions over character-level inputs.
- Max-over-time pooling (effectively n-gram selection).

conv1d, padded with max pooling over time

∅	**0.0**	**0.0**	**0.0**	**0.0**
tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3
∅	**0.0**	**0.0**	**0.0**	**0.0**

∅,t,d	−0.6	0.2	1.4
t,d,r	−1.0	1.6	−1.0
d,r,t	−0.5	−0.1	0.8
r,t,k	−3.6	0.3	0.3
t,k,g	−0.2	0.1	1.2
k,g,o	0.3	0.6	0.9
g,o,∅	−0.5	−0.9	0.1

max p	0.3	1.6	1.4

Apply 3 **filters** of size 3

3	1	2	−3
−1	2	1	−3
1	1	−1	1

1	0	0	1
1	0	−1	−1
0	1	0	1

1	−1	2	−1
1	0	−1	3
0	2	2	1

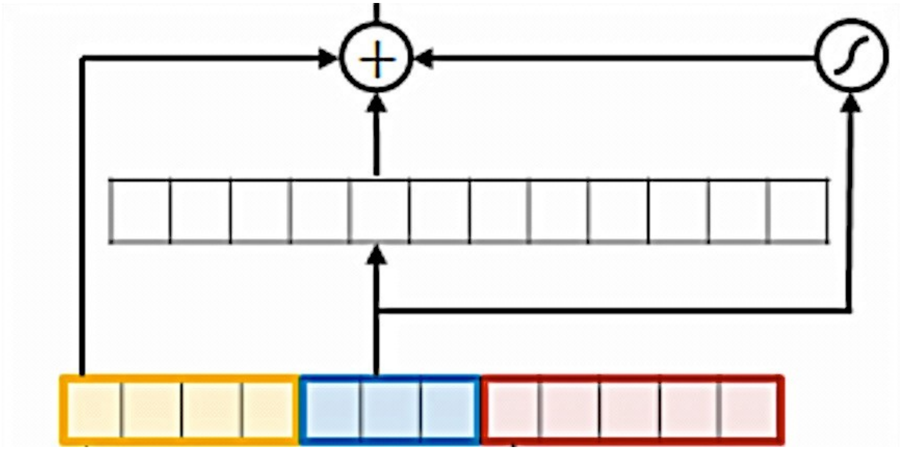
Highway Network (Srivastava et al. 2015)

- Apply transformation while carrying over original information.
- Functions akin to an LSTM memory cell.

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

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

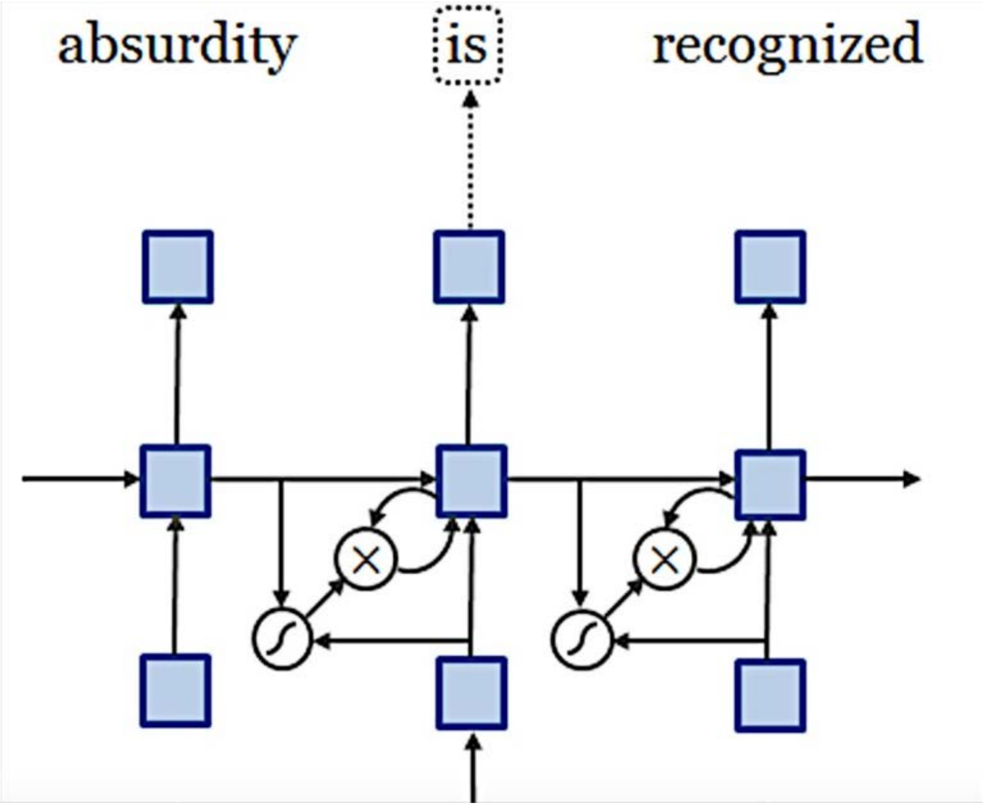
Transform Gate Input Carry Gate



CNN Output

Long Short-Term Memory Network

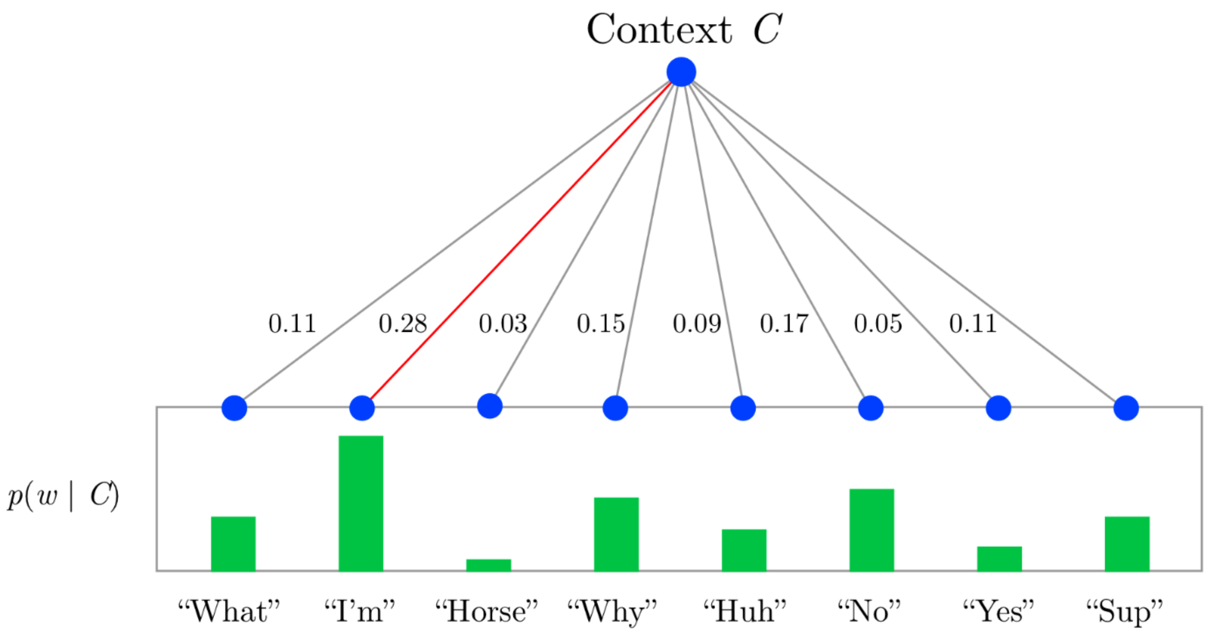
- Hierarchical Softmax to handle large output vocabulary (V).
- Trained with truncated backpropagation through time.

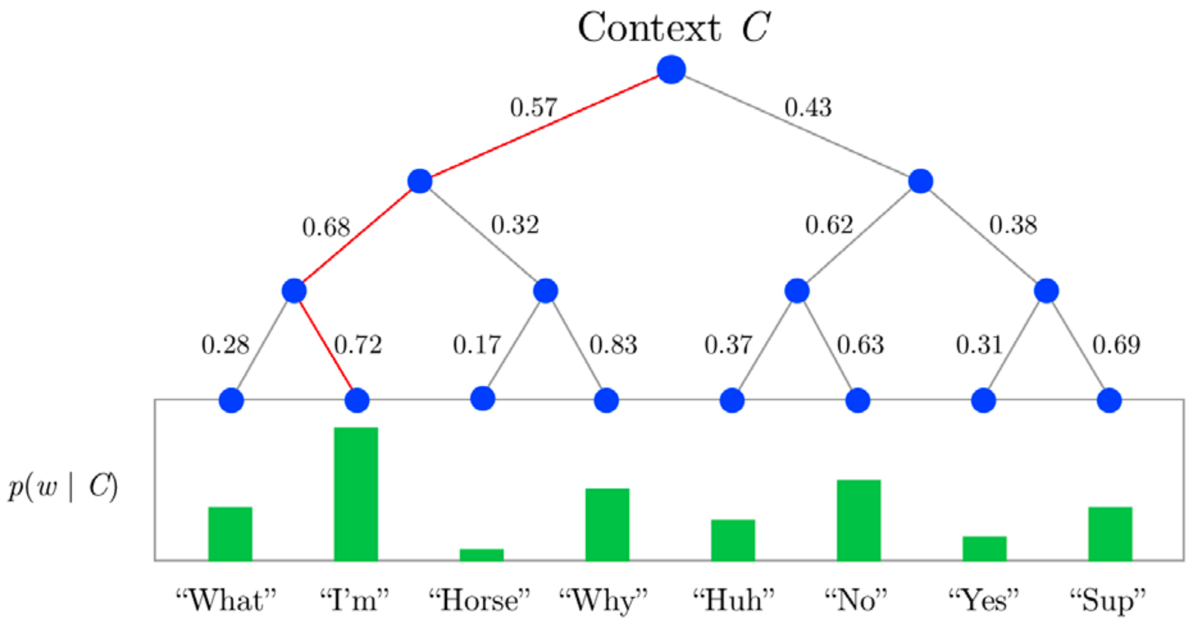


Highway Network Output

Long Short-Term Memory Network

- Hierarchical Softmax to handle large output vocabulary (V).
  - O(V) -> O($\log_2$V)



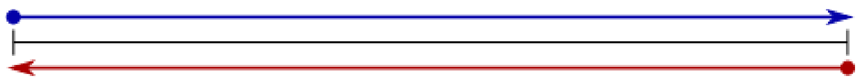


https://www.quora.com/What-is-hierarchical-softmax

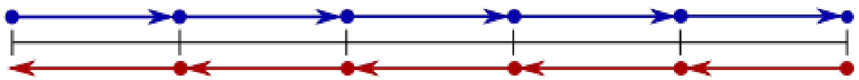
Long Short-Term Memory Network

- Trained with truncated backpropagation through time.

A modified version of the BPTT training algorithm for recurrent neural networks where the sequence is processed one timestep at a time and periodically and the BPTT update is performed back for a fixed number of timesteps



(a) BPTT



(b) Truncated BPTT

Quantitative 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**

Test set perplexity

Comparable performance with fewer parameters!

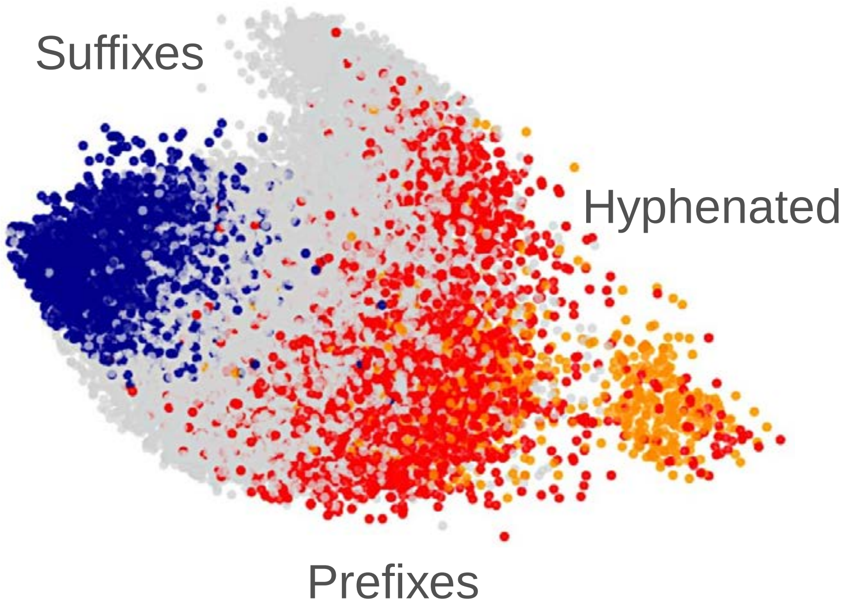
	PPL	Size
LSTM-Word-Small	97.6	5 m
LSTM-Char-Small	92.3	5 m
LSTM-Word-Large	85.4	20 m
→ LSTM-Char-Large	78.9	19 m
KN-5 (Mikolov et al. 2012)	141.2	2 m
RNN† (Mikolov et al. 2012)	124.7	6 m
RNN-LDA† (Mikolov et al. 2012)	113.7	7 m
genCNN† (Wang et al. 2015)	116.4	8 m
FOFE-FNNLM† (Zhang et al. 2015)	108.0	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

Qualitative Insights

	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*

Qualitative Insights

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*



Plot of character n-gram representations via PCA for English. Prefixes (red), suffixes (blue), hyphenated (orange), and all others (grey). Prefixes refer to character n-grams which start with the start-of-word character. Suffixes like-wise refer to character n-grams which end with the end-of-word character

Take-aways

- Paper questioned the necessity of using word embeddings as inputs for neural language modeling.
- CNNs + Highway Network over characters can extract rich semantic and structural information.
- Key thinking: you can compose "building blocks" to obtain nuanced and powerful models!

Hybrid NMT

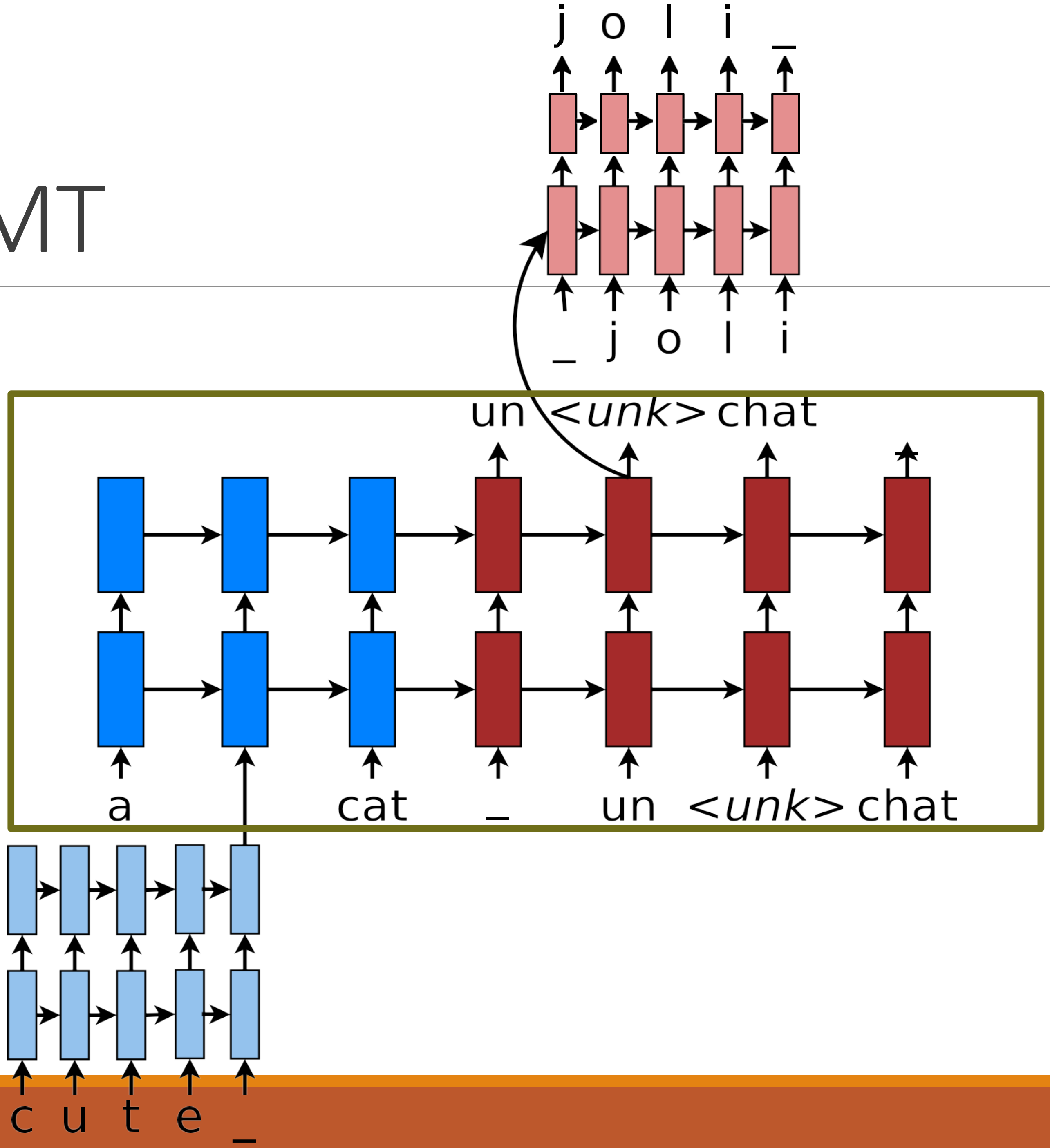
- A best-of-both-worlds architecture:
  - Translate mostly at the word level
  - Only go to the character level when needed (rare words)

- More than 2 BLEU improvement over a copy mechanism (exact word string from source to target sentence) to try to fill in unknown words

Thang Luong and Chris Manning. ***Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models****. ACL 2016.*

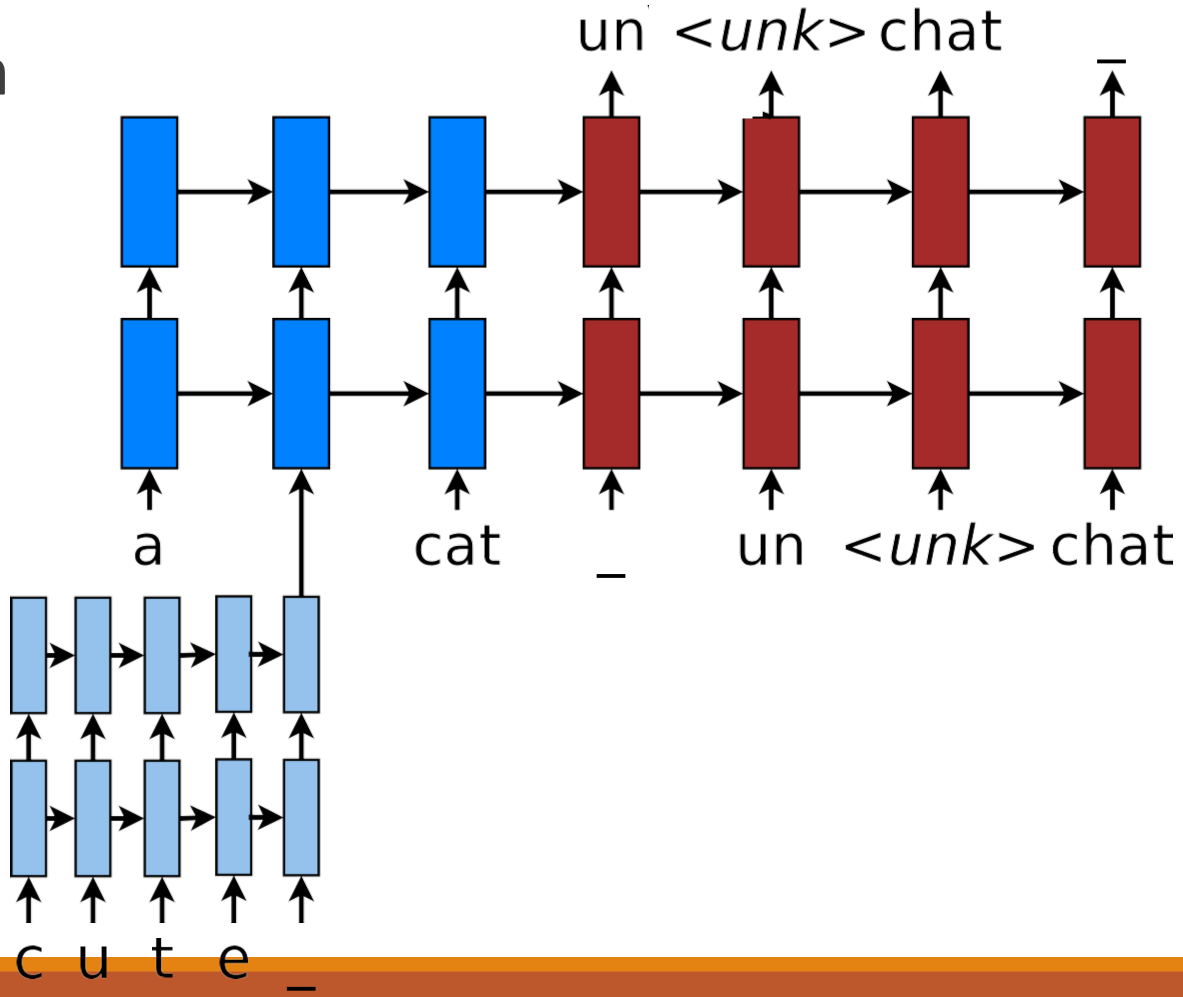
Hybrid NMT

Word-level
(4 layers)



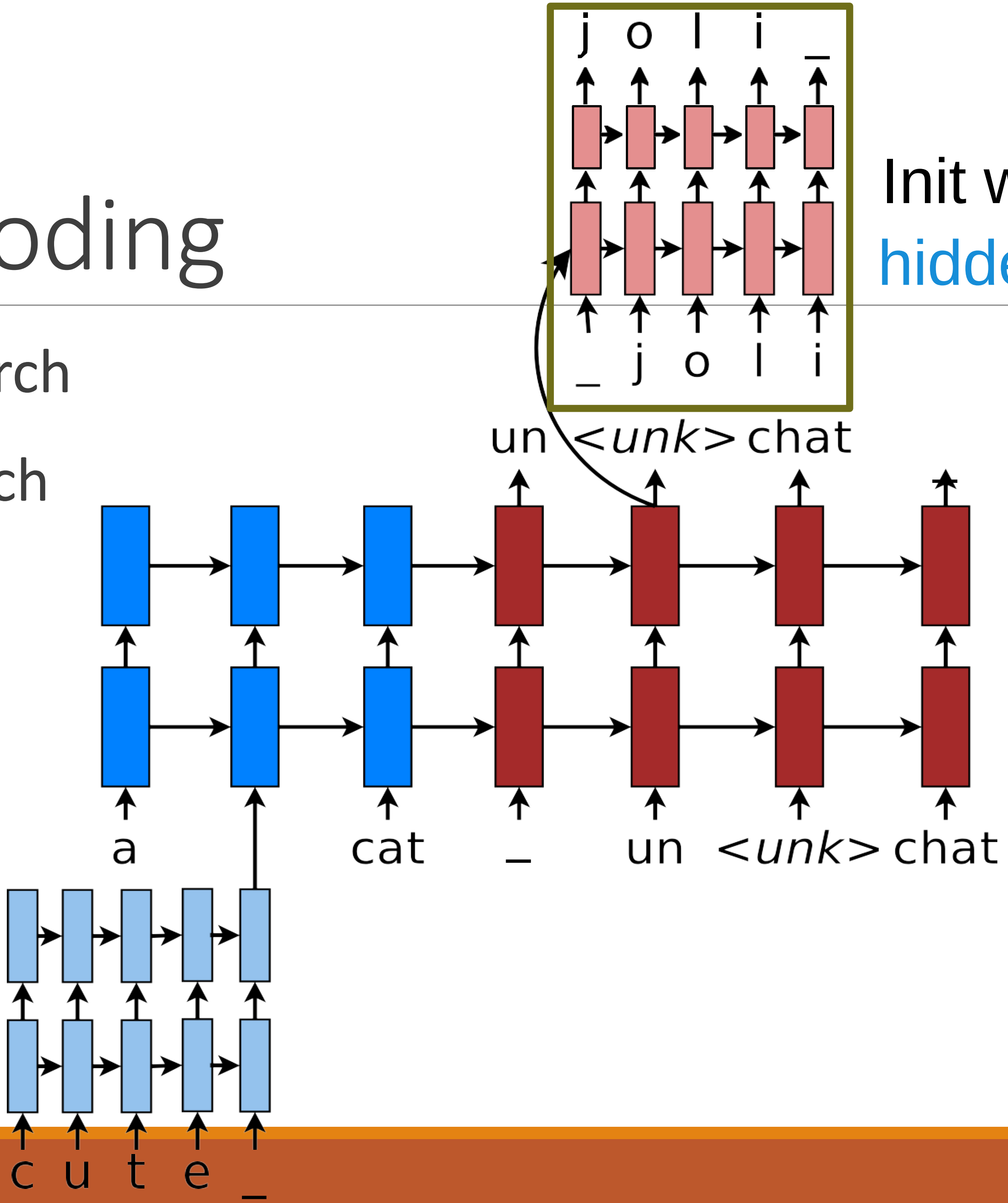
2-stage Decoding

- Word-level beam search



2-stage Decoding

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



Init with word hidden states.

English-Czech Results

- Train on WMT’15 data (12M sentence pairs)
  - newstest2015

Systems	BLEU
Winning WMT’15 (Bojar & Tamchyna, 2015)	18.8
Word-level NMT (Jean et al., 2015)	18.3

30x additional data
3 systems (2 MT, 1 post-editing)

Large vocab
+ copy mechanism

English-Czech Results

- Train on WMT'15 data (12M sentence pairs)
  - newstest2015

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**

Sample English-Czech translations

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!

Sample English-Czech translations

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

Sample English-Czech translations

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

Sample English-Czech translations

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

Sample English-Czech translations

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

when generate words, although feeding the hidden representation of the word level model in as the starting hidden representation of the character level model, it doesn't have any representation further back than that of what's in the word level model.

Sample English-Czech translations

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***

it tends to not always do a good job at representing

<Hybrid>

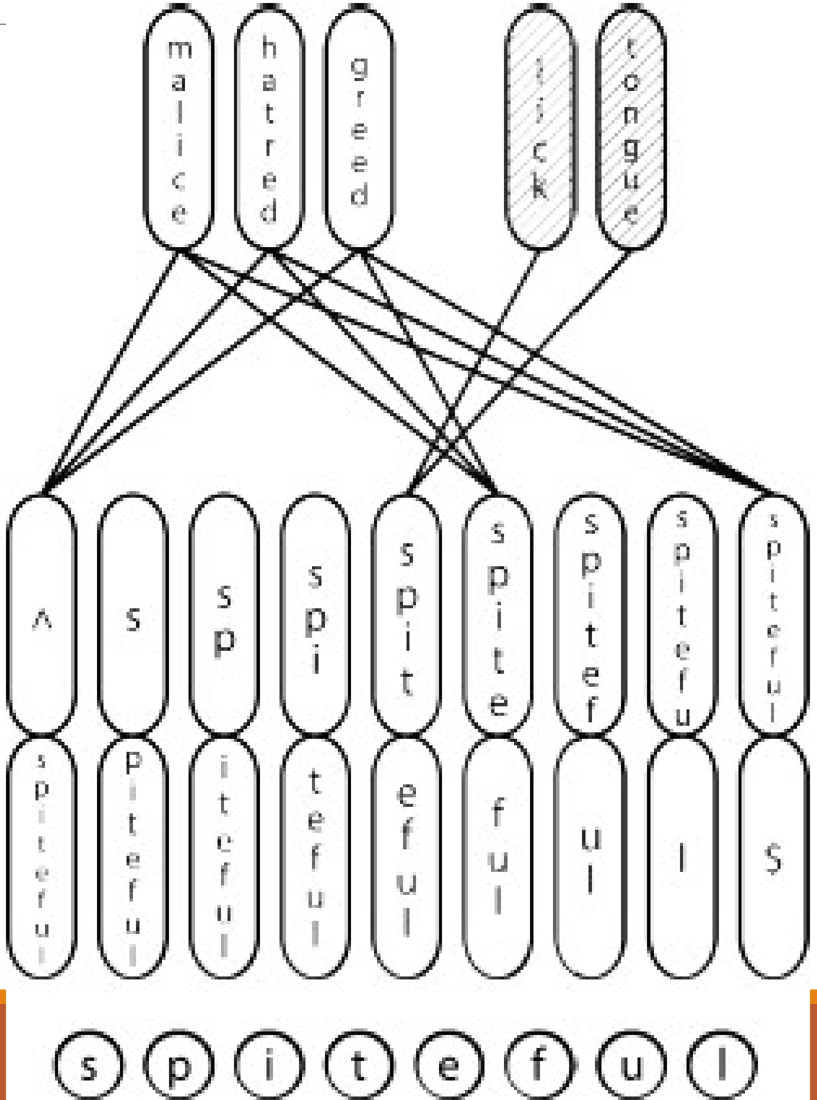
Hybrid: correct, 11-year-old – jedenáctiletá

Wrong: Shani Bartová

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

Enriching Word Vectors with Subword Information Bojanowski, Grave, Joulin and Mikolov. FAIR. 2016.

- https://arxiv.org/pdf/1607.04606.pdf
- https://fasttext.cc

- 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

FastText embeddings

- Represent word as char *n*-grams augmented with boundary symbols and as whole word, designated as $G(w)$:
- *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 (z_g).
- Word in context (c) score is: $s(w,c) = \sum_{g \in G(w)} {z_g}^T v_c$

FastText embeddings

- Correlation between human judgement and similarity scores on word similarity datasets

		sg	cbow	sisg-	sisg
AR	WS353	51	52	54	**55**
DE	GUR350	61	62	64	**70**
	GUR65	78	78	**81**	**81**
	ZG222	35	38	41	**44**
EN	RW	43	43	46	**47**
	WS353	72	**73**	71	71
ES	WS353	57	58	58	**59**
FR	RG65	70	69	**75**	**75**
RO	WS353	48	52	51	**54**
RU	HJ	59	60	60	**66**

FastText embeddings

- Spearman's rank correlation coefficient between human judgement and model scores for different methods using morphology to learn (OOV) word representations

	DE		EN		ES	FR
	GUR350	ZG222	WS353	RW	WS353	RG65
Luong et al. (2013)	-	-	64	34	-	-
Qiu et al. (2014)	-	-	65	33	-	-
Soricut and Och (2015)	64	22	71	42	47	67
sisg	73	43	73	48	54	69
Botha and Blunsom (2014)	56	25	39	30	28	45
sisg	66	34	54	41	49	52

Trained on different datasets