Natural Language Processing with Deep Learning

Information from parts of words: Subword Models
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 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
Morphology: Parts of words

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

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

A possible way of dealing with a larger vocabulary – most unseen words are new morphological forms (or numbers)
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

- No word segmentation
- Words (mainly) segmented: *This is a sentence with words*
  - Clitics?
    - Separated: *Je vous ai apporté des bonbons*
    - Joined: *ف+ قال + نا + ها = فقالناها* = *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: nejneob hospodařovávatelně jšímu
    (“to the worst farmable one”)
  
- Transliteration: Christopher $\mapsto$ Kryštof

- Informal spelling:

```plaintext
Brianna @_parsimonia_ · 24h
Gooooood Vibessssssss
```

```plaintext
@J0YUS · 1m
When idc, I really don’t care.
Like my “I want space” is me shutting you out. My “imma go, u want something?” And u don’t say nothing, then I’m not coming back sumn 4 u
```
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
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!

- Phonemic (maybe digraphs)  jiyawu ngabulu
- Fossilized phonemic  thorough failure
- Syllabic/moraic  とうほく
- Ideographic (syllabic)  去年太空船二号坠毁
- Combination of the above  インド洋の島
Purely character-level models

• one good example of a purely character-level model for sentence classification:
  • Very Deep Convolutional Networks for Text Classification
  • Conneau, Schwenk, Lecun, Barrault. EACL 2017
• Strong results via a deep convolutional stack
Purely character-level NMT models

- Initially, **unsatisfactory** performance
  - (Vilar et al., 2007; Neubig et al., 2013)
- **Decoder only**
- Then **promising** results
  - (Thang Luong, Christopher Manning, ACL 2016)
  - (Marta R. Costa-Jussà, José A. R. Fonollosa, ACL 2016)
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 slow. 3 weeks to train ... not that fast at runtime.

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word-level model (single; large vocab; UNK replace)</td>
<td>15.7</td>
</tr>
<tr>
<td>Character-level model (single; 600-step backprop)</td>
<td>15.9</td>
</tr>
</tbody>
</table>
Fully Character-Level Neural Machine Translation without Explicit Segmentation

Encoder as below; decoder is a char-level GRU

<table>
<thead>
<tr>
<th>CS-En</th>
<th>WMT 15</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Target</td>
<td>BLEU</td>
</tr>
<tr>
<td>Bpe</td>
<td>Bpe</td>
<td>20.3</td>
</tr>
<tr>
<td>Bpe</td>
<td>Char</td>
<td>22.4</td>
</tr>
<tr>
<td>Char</td>
<td>Char</td>
<td>22.5</td>
</tr>
</tbody>
</table>

*byte pair encoding
*2-layer with attention

CS = Czech
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

6*2+8:
6 bi-LSTM encoder
8 decoder
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 $\mapsto$ 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:
  - Start with a vocabulary of characters
  - Most frequent ngram pairs $\mapsto$ a new ngram

Start with all characters in vocab

(Example from Sennrich)
Byte Pair Encoding

• A word segmentation algorithm:
  • Start with a vocabulary of characters
  • Most frequent ngram pairs $\mapsto$ a new ngram

**Dictionary**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>low</td>
</tr>
<tr>
<td>2</td>
<td>lower</td>
</tr>
<tr>
<td>6</td>
<td>newest</td>
</tr>
<tr>
<td>3</td>
<td>widest</td>
</tr>
</tbody>
</table>

**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

• 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
- Sentencepiece model works from raw text
  - Whitespace is retained as special token (_) and grouped normally
  - You can reverse things at end by joining pieces and recoding them to spaces

- [https://github.com/google/sentencepiece](https://github.com/google/sentencepiece)
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

*all subwords start with ## except the first one

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

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

Prediction

LSTM

Highway Network

CNN

Character Embeddings

absurdity is recognized

moment the absurdity is recognized
Convolutional Layer

- Convolutions over character-level inputs.
- Max-over-time pooling (effectively n-gram selection).
Highway Network (Srivastava et al. 2015)

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

\[
t = \sigma(W_T y + b_T)
\]

\[
z = t \odot g(W_H y + b_H) + (1 - t) \odot y
\]
Long Short-Term Memory Network

- Hierarchical Softmax to handle large output vocabulary.
- Trained with truncated backprop through time.
# Quantitative Results

Comparable performance with fewer parameters!

<table>
<thead>
<tr>
<th>Data-S</th>
<th>CS</th>
<th>DE</th>
<th>ES</th>
<th>FR</th>
<th>RU</th>
<th>AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Botha</td>
<td>KN-4</td>
<td>545</td>
<td>366</td>
<td>241</td>
<td>274</td>
<td>396</td>
</tr>
<tr>
<td></td>
<td>MLBL</td>
<td>465</td>
<td>296</td>
<td>200</td>
<td>225</td>
<td>304</td>
</tr>
<tr>
<td>Small</td>
<td>Word</td>
<td>503</td>
<td>305</td>
<td>212</td>
<td>229</td>
<td>352</td>
</tr>
<tr>
<td></td>
<td>Morph</td>
<td>414</td>
<td>278</td>
<td>197</td>
<td>216</td>
<td>290</td>
</tr>
<tr>
<td></td>
<td>Char</td>
<td>401</td>
<td>260</td>
<td>182</td>
<td>189</td>
<td>278</td>
</tr>
<tr>
<td>Large</td>
<td>Word</td>
<td>493</td>
<td>286</td>
<td>200</td>
<td>222</td>
<td>357</td>
</tr>
<tr>
<td></td>
<td>Morph</td>
<td>398</td>
<td>263</td>
<td>177</td>
<td>196</td>
<td>271</td>
</tr>
<tr>
<td></td>
<td>Char</td>
<td>371</td>
<td>239</td>
<td>165</td>
<td>184</td>
<td>261</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data-L</th>
<th>CS</th>
<th>DE</th>
<th>ES</th>
<th>FR</th>
<th>RU</th>
<th>EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Botha</td>
<td>KN-4</td>
<td>862</td>
<td>463</td>
<td>219</td>
<td>243</td>
<td>390</td>
</tr>
<tr>
<td></td>
<td>MLBL</td>
<td>643</td>
<td>404</td>
<td>203</td>
<td>227</td>
<td>300</td>
</tr>
<tr>
<td>Small</td>
<td>Word</td>
<td>701</td>
<td>347</td>
<td>186</td>
<td>202</td>
<td>353</td>
</tr>
<tr>
<td></td>
<td>Morph</td>
<td>615</td>
<td>331</td>
<td>189</td>
<td>209</td>
<td>331</td>
</tr>
<tr>
<td></td>
<td>Char</td>
<td>578</td>
<td>305</td>
<td>169</td>
<td>190</td>
<td>313</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>LSTM-Word</th>
<th>In Vocabulary</th>
<th>LSTM-Char (before highway)</th>
<th>LSTM-Char (after highway)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>while</td>
<td>his</td>
<td>you</td>
<td>richard</td>
</tr>
<tr>
<td></td>
<td>although</td>
<td>your</td>
<td>conservatives</td>
<td>jonathan</td>
</tr>
<tr>
<td></td>
<td>letting</td>
<td>her</td>
<td>we</td>
<td>robert</td>
</tr>
<tr>
<td></td>
<td>though</td>
<td>my</td>
<td>guys</td>
<td>neil</td>
</tr>
<tr>
<td></td>
<td>minute</td>
<td>their</td>
<td>i</td>
<td>nancy</td>
</tr>
<tr>
<td></td>
<td>chile</td>
<td>this</td>
<td>your</td>
<td>hard</td>
</tr>
<tr>
<td></td>
<td>whole</td>
<td>hhs</td>
<td>young</td>
<td>rich</td>
</tr>
<tr>
<td></td>
<td>meanwhile</td>
<td>is</td>
<td>four</td>
<td>richer</td>
</tr>
<tr>
<td></td>
<td>white</td>
<td>has</td>
<td>youth</td>
<td>richter</td>
</tr>
<tr>
<td></td>
<td>meanwhile</td>
<td>hhs</td>
<td>we</td>
<td>eduard</td>
</tr>
<tr>
<td></td>
<td>whole</td>
<td>this</td>
<td>your</td>
<td>gerard</td>
</tr>
<tr>
<td></td>
<td>though</td>
<td>their</td>
<td>doug</td>
<td>edward</td>
</tr>
<tr>
<td></td>
<td>nevertheless</td>
<td>your</td>
<td>i</td>
<td>carl</td>
</tr>
</tbody>
</table>

- trading
- advertised
- advertising
- turnover
- heading
- training
- reading
- leading
- trade
- training
- traded
- trader
## Qualitative Insights

### Out-of-Vocabulary

<table>
<thead>
<tr>
<th>computer-aided</th>
<th>misinformed</th>
<th>looooook</th>
</tr>
</thead>
<tbody>
<tr>
<td>computer-guided</td>
<td>informed</td>
<td>look</td>
</tr>
<tr>
<td>computerized</td>
<td>performed</td>
<td>cook</td>
</tr>
<tr>
<td>disk-drive</td>
<td>transformed</td>
<td>looks</td>
</tr>
<tr>
<td>computer</td>
<td>inform</td>
<td>shook</td>
</tr>
<tr>
<td>computer-driven</td>
<td></td>
<td></td>
</tr>
<tr>
<td>computerized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>computer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>computerized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>computer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Suffixes
- 
- 
- 
- 

### Hyphenated
- 
- 
- 
- 

### Prefixes
- 
- 
- 
- 

38
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

- More than 2 BLEU improvement over a copy mechanism to try to fill in rare words

Hybrid NMT

Word-level (4 layers)

End-to-end training 8-stacking LSTM layers.
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

<table>
<thead>
<tr>
<th>Systems</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winning WMT’15 (Bojar &amp; Tamchyna, 2015)</td>
<td>18.8</td>
</tr>
<tr>
<td><strong>Word-level</strong> NMT (Jean et al., 2015)</td>
<td>18.3</td>
</tr>
<tr>
<td><strong>Hybrid</strong> NMT (Luong &amp; Manning, 2016)*</td>
<td>20.7</td>
</tr>
</tbody>
</table>

*30x data
3 systems
Large vocab + copy mechanism

Then SOTA!

But cf. Cherry et al. 2018: ~26 BLEU
# Sample English-Czech translations

<table>
<thead>
<tr>
<th>source</th>
<th>The author <em>Stephen Jay Gould</em> died 20 years after <em>diagnosis</em>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
<td>Autor <em>Stephen Jay Gould</em> zemřel 20 let po <em>diagnóze</em>.</td>
</tr>
<tr>
<td>char</td>
<td>Autor <em>Stephen Stephen</em> zemřel 20 let po <em>diagnóze</em>.</td>
</tr>
<tr>
<td>word</td>
<td>Autor Stephen Jay <em>&lt;unk&gt;</em> zemřel 20 let po <em>&lt;unk&gt;</em>.</td>
</tr>
<tr>
<td>hybrid</td>
<td>Autor <em>Stephen Jay Gould</em> zemřel 20 let po <em>diagnóze</em>.</td>
</tr>
</tbody>
</table>

Perfect translation!
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 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:
- $\textit{where} = \langle \textit{wh}, \textit{whe}, \textit{her}, \textit{ere}, \textit{re} \rangle, \langle \textit{where} \rangle$
  - Prefix, suffixes and whole words are special
  - Represent word as sum of these representations. Word in context score is:
  - $s(w, c) = \sum_{g \in G(w)} z_g^T v_c$
    - $n$-gram vector $z_g$: skip-gram loss function
    - Detail: rather than sharing representation for all $n$-grams, use “hashing trick” to have fixed number of vectors
## FastText embeddings

Word similarity dataset scores (correlations)

<table>
<thead>
<tr>
<th></th>
<th>sg</th>
<th>cbow</th>
<th>sisg-</th>
<th>sisg</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AR</strong></td>
<td>WS353</td>
<td>51</td>
<td>52</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>GUR350</td>
<td>61</td>
<td>62</td>
<td>64</td>
</tr>
<tr>
<td><strong>DE</strong></td>
<td>GUR65</td>
<td>78</td>
<td>78</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>ZG222</td>
<td>35</td>
<td>38</td>
<td>41</td>
</tr>
<tr>
<td><strong>EN</strong></td>
<td>RW</td>
<td>43</td>
<td>43</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>WS353</td>
<td>72</td>
<td>73</td>
<td>71</td>
</tr>
<tr>
<td><strong>ES</strong></td>
<td>WS353</td>
<td>57</td>
<td>58</td>
<td>58</td>
</tr>
<tr>
<td><strong>FR</strong></td>
<td>RG65</td>
<td>70</td>
<td>69</td>
<td>75</td>
</tr>
<tr>
<td><strong>RO</strong></td>
<td>WS353</td>
<td>48</td>
<td>52</td>
<td>51</td>
</tr>
<tr>
<td><strong>RU</strong></td>
<td>HJ</td>
<td>59</td>
<td>60</td>
<td>60</td>
</tr>
</tbody>
</table>

*sisg*: OOV as null vector, sisg (fasttext)