Words and Morphology

Philipp Koehn

20 October 2020

A Naive View of Language

- Language needs to name
  - nouns: objects in the world (*dog*)
  - verbs: actions (*jump*)
  - adjectives and adverbs: properties of objects and actions (*brown*, *quickly*)

- Relationship between these have to specified
  - word order
  - morphology
  - function words

Unknown Words

- Ratio of unknown words in WMT 2013 test set:

Source language	Ratio unknown
Russian	2.0%
Czech	1.5%
German	1.2%
French	0.5%
English (to French)	0.5%

- Caveats:
  - corpus sizes differ
  - not clear which unknown words have known morphological variants

Large Vocabularies

- Zipf's law tells us that words in a language are very unevenly distributed.
  - large tail of rare words
    (e.g., new words *retweeting, website, woke, lit*)
  - large inventory of names, e.g., *eBay, Yahoo, Microsoft*

- Neural methods not well equipped to deal with such large vocabularies
  (ideal representations are continuous space vectors → word embeddings)

- Large vocabulary
  - large embedding matrices for input and output words
  - prediction and softmax over large number of words

- Computationally expensive, both in terms of memory and speed

Special Treatment for Rare Words

- Limit vocabulary to 20,000 to 80,000 words

- First idea
  - map other words to unknown word token (UNK)
  - model learns to map input UNK to output UNK
  - replace with translation from backup dictionary

- Not used anymore, except for numbers and units
  - numbers: English *540,000*, Chinese *54* TENTHOUSAND, Indian *5.4 lakh*
  - units: map *25cm* to *10 inches*

Some Causes for Large Vocabularies

- Morphology

  tweet, *tweets*, *tweeted*, *tweeting*, *retweet*, ...

  → morphological analysis?

- Compounding

  homework, *website*, ...

  → compound splitting?

- Names

  Netanyahu, *Jones*, *Macron*, *Hoboken*, ...

  → transliteration?

⇒ Breaking up words into **subwords** may be a good idea

Byte Pair Encoding

- Start by breaking up words into characters

```
t h e ␣ f a t ␣ c a t ␣ i s ␣ i n ␣ t h e ␣ t h i n ␣ b a g
```

- Merge frequent pairs

```
t h→th    th e ␣ f a t ␣ c a t ␣ i s ␣ i n ␣ th e ␣ th i n ␣ b a g
a t→at    th e ␣ f at ␣ c at ␣ i s ␣ i n ␣ th e ␣ th i n ␣ b a g
i n→in    th e ␣ f at ␣ c at ␣ i s ␣ in ␣ th e ␣ th in ␣ b a g
th e→the  the ␣ f at ␣ c at ␣ i s ␣ in ␣ the ␣ th in ␣ b a g
```

- Each merge operation increases the vocabulary size
  - starting with the size of the character set (maybe 100 for Latin script)
  - stopping after, say, 50,000 operations

Byte Pair Encoding

Obama receives Net@@ any@@ ahu

the relationship between Obama and Net@@ any@@ ahu is not exactly friendly . the two wanted to talk about the implementation of the international agreement and about Teheran ’s destabil@@ ising activities in the Middle East . the meeting was also planned to cover the conflict with the Palestinians and the disputed two state solution . relations between Obama and Net@@ any@@ ahu have been stra@@ ined for years . Washington critic@@ ises the continuous building of settlements in Israel and acc@@ uses Net@@ any@@ ahu of a lack of initiative in the peace process . the relationship between the two has further deteriorated because of the deal that Obama negotiated on Iran ’s atomic programme . in March , at the invitation of the Republic@@ ans , Net@@ any@@ ahu made a controversial speech to the US Congress , which was partly seen as an aff@@ ront to Obama . the speech had not been agreed with Obama , who had rejected a meeting with reference to the election that was at that time im@@ pending in Israel .

Subwords

- Byte pair encoding induces subwords

- But: only accidentally along linguistic concepts of morphology
  - morphological: `critic@@ ises`, `im@@ pending`
  - not morphological: `aff@@ ront`, `Net@@ any@@ ahu`

- Still: Similar to unsupervised morphology (frequent suffixes, etc.)

Sentence Piece

_Obama _receives _Net any ahu

_the _relationship _between _Obama _and _Net any ahu _is _not _exactly
_friendly _. _the _two _wanted _to _talk _about _the _implementation _of
_the _international _agreement _and _about _Teheran _'s _destabil ising
_activities _in _the _Middle _East _. _the _meeting _was _also _planned
_to _cover _the _conflict _with _the _Palestinians _and _the _disputed
_two _state _solution _. _relations _between _Obama _and Net _any _ahu
_have _been _stra ined _for _years _. _Washington _critic ises _the
_continuous _building _of _settlements _in _Israel _and _acc uses _Net any
ahu _of _a _lack _of _initiative _in _the _peace _process _. _the
_relationship _between _the _two _has _further _deteriorated _because _of
_the _deal _that _Obama _negotiated _on _Iran _'s _atomic _programme _.
_in _March _, _at _the _invitation _of _the _Republic ans _, _Net any ahu
_made _a _controversial _speech _to _the _US _Congress _, _which _was
_partly _seen _as _an _aff ront _to _Obama _. _the _speech _had _not
_been _agreed _with _Obama _, _who _had _rejected _a _meeting _with
_reference _to _the _election _that _was _at _that _time _im pending _in
_Israel .

character-based models

Character-Based Models

- Explicit word models that yield word embeddings

- Standard methods for frequent words
  - distribution of `beautiful` in the data

  → embedding for `beautiful`

- Character-based models
  - create sequence embedding for character string `b e a u t i f u l`
  - training objective: match word embedding for `beautiful`

- Induce embeddings for unseen morphological variants
  - character string `b e a u t i f u l l y`

  → embedding for `beautifully`

- Hope that this learns morphological principles

Character Sequence Models

- Same model as for words
- Tokens = single characters, incl. special space symbol
- But: generally poor performance
- With some refinements, use in output shown competitive

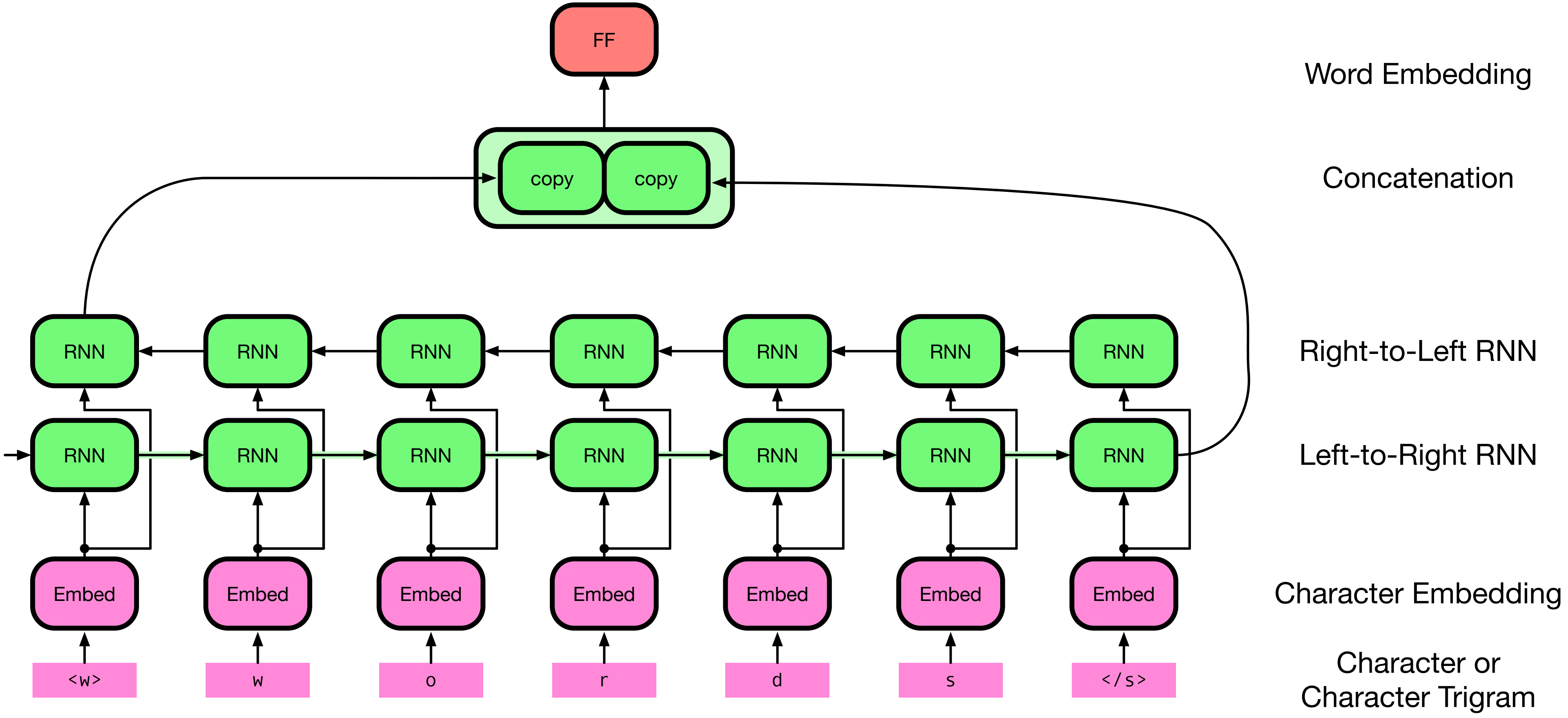
Character Based Word Models

- Word embeddings as before

- Compute word embeddings based on character sequence

- Typically, interpolated with traditional word embeddings

Recurrent Neural Networks


FF
copy
copy
RNN
RNN
Embed
<w>
w
o
r
d
s
</s>
Word Embedding
Concatenation
Right-to-Left RNN
Left-to-Right RNN
Character Embedding
Character or
Character Trigram