

---

# 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*)

# A Naive View of Language



1

- 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

# Marking of Relationships: Agreement

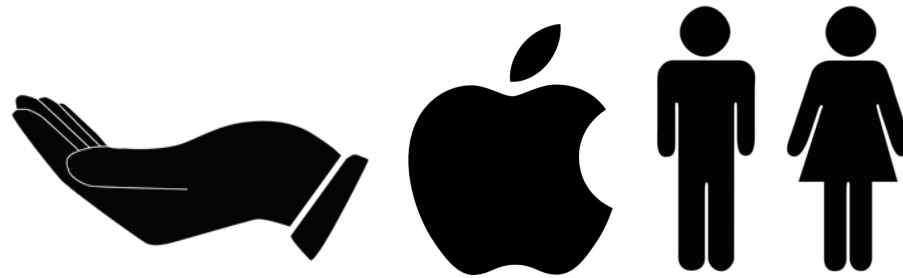
- From Catullus, First Book, first verse (Latin):
- Gender (and case) agreement links adjectives to nouns



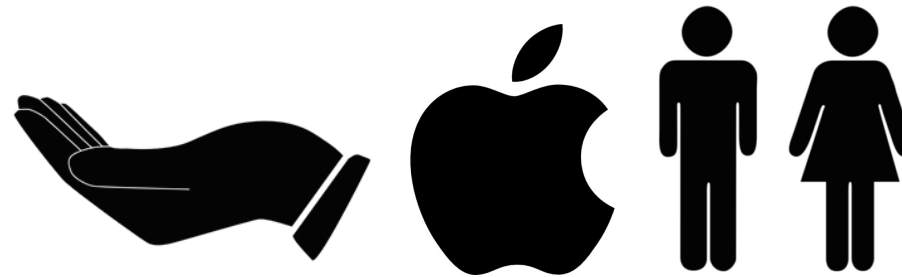
*Cui dono lepidum novum libellum arida modo pumice expolitus ?*  
*Whom I-present lovely new little-book dry manner pumice polished ?*

*(To whom do I present this lovely new little book now polished with a dry pumice?)*

# Marking of Relationships to Verb: Case



# Marking of Relationships to Verb: Case



- German:

<i>Die Frau</i>	<i>gibt</i>	<i>dem Mann</i>	<i>den Apfel</i>
<i>The woman</i>	<i>gives</i>	<i>the man</i>	<i>the apple</i>
subject		indirect object	object

- Case inflection indicates role of noun phrases

# Writingwordstogether



4

- Definition of word boundaries purely an artifact of writing system

# Writingwordstogether



4

- Definition of word boundaries purely an artifact of writing system
- Differences between languages
  - Agglutinative compounding  
*Informatikseminar* vs. *computer science seminar*
  - Function word vs. affix



# Writingwordstogether



- Definition of word boundaries purely an artifact of writing system
- Differences between languages
  - Agglutinative compounding  
*Informatikseminar* vs. *computer science seminar*
  - Function word vs. affix
- Border cases
  - *Joe's* — one token or two?
  - Morphology of affixes often depends on phonetics / spelling conventions  
*dog+s* → *dogs* vs. *pony* → *ponies*  
... but note the English function word *a*:  
*a donkey* vs. *an aardvark*

# Changing Part-of-Speech



- Derivational morphology allows changing part of speech of words

# Changing Part-of-Speech



- Derivational morphology allows changing part of speech of words
- Example:
  - base: *nation*, noun
  - *national*, adjective
  - *nationally*, adverb
  - *nationalist*, noun
  - *nationalism*, noun
  - *nationalize*, verb

# Changing Part-of-Speech



- Derivational morphology allows changing part of speech of words
- Example:
  - base: *nation*, noun
  - *national*, adjective
  - *nationally*, adverb
  - *nationalist*, noun
  - *nationalism*, noun
  - *nationalize*, verb
- Sometimes distinctions between POS quite fluid (enabled by morphology)
  - *I want to integrate morphology*
  - *I want the integration of morphology*

# Meaning Altering Affixes



- English

*undo*

*redo*

*hypergraph*

# Meaning Altering Affixes



- English

*undo*

*redo*

*hypergraph*

- German: *zer-* implies action causes destruction

*Er **zer**redet das Thema → He talks the topic **to death***

# Meaning Altering Affixes



- English

*undo*

*redo*

*hypergraph*

- German: *zer-* implies action causes destruction

*Er **zer**redet das Thema → He talks the topic **to death***

- Spanish: *-ito* means object is small

*burro → burrito*

# Adding Subtle Meaning



- Morphology allows adding subtle meaning
  - verb tenses: time action is occurring, if still ongoing, etc.
  - count (singular, plural): how many instances of an object are involved
  - definiteness (*the cat* vs. *a cat*): relation to previously mentioned objects
  - grammatical gender: helps with co-reference and other disambiguation
- Sometimes redundant: same information repeated many times



# how does morphology impact machine translation?

# 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

# Differently Encoded Information

- Languages with different sentence structure

<i>das</i>	<i>behaupten</i>	<i>sie</i>	<i>wenigstens</i>
<i>this</i>	<i>claim</i>	<i>they</i>	<i>at least</i>
<i>the</i>		<i>she</i>	

- Convert from inflected language into configuration language (and vice versa)
- Ambiguities can be resolved through syntactic analysis
  - the meaning *the* of *das* not possible (not a noun phrase)
  - the meaning *she* of *sie* not possible (subject-verb agreement)

- Pronominal anaphora

*I saw the movie and **it** is good.*

- How to translate *it* into German (or French)?
  - *it* refers to *movie*
  - *movie* translates to *Film*
  - *Film* has masculine gender
  - ergo: *it* must be translated into masculine pronoun *er*
- We are not handling pronouns very well

- Example

*Whenever I visit my uncle and his daughters,  
I can't decide who is my favorite **cousin**.*

- How to translate *cousin* into German? Male or female?

# morphological pre-processing schemes

- German sentence with morphological analysis

<i>Er</i>	<i>wohnt</i>	<i>in</i>	<i>einem</i>	<i>großen</i>	<i>Haus</i>
<i>Er</i>	<i>wohnen -en+t</i>	<i>in</i>	<i>ein +em</i>	<i>groß +en</i>	<i>Haus +ε</i>
<i>He</i>	<i>lives</i>	<i>in</i>	<i>a</i>	<i>big</i>	<i>house</i>

- Four inflected words in German, but English...

**also inflected** both English verb *live* and German verb *wohnen*  
inflected for tense, person, count

**not inflected** corresponding English words not inflected (*a* and *big*)  
→ easier to translate if inflection is stripped

**less inflected** English word *house* inflected for count  
German word *Haus* inflected for count and case  
→ reduce morphology to singular/plural indicator

- Reduce German morphology to match English

*Er* | *wohnen+3P-SGL* | *in* | *ein* | *groß* | *Haus+SGL*

- Example
  - Turkish: **Sonuçlarına**<sub>1</sub> **dayanılarak**<sub>2</sub> **bir**<sub>3</sub> **ortaklığı**<sub>4</sub> **oluşturulacaktır**<sub>5</sub>.
  - English: **a**<sub>3</sub> **partnership**<sub>4</sub> *will be* **drawn-up**<sub>5</sub> *on the* **basis**<sub>2</sub> *of* **conclusions**<sub>1</sub> .
- Turkish morphology → English function words (*will, be, on, the, of*)
- Morphological analysis

*Sonuç +lar +sh +na daya +hnhl +yarak bir ortaklık +sh oluş +dhr +hl +yacak +dhr*

- Alignment with morphemes

<i>sonuç</i>	<i>+lar</i>	<i>+sh</i>	<i>+na</i>		<i>daya+hnhl</i>	<i>+yarak</i>		<i>bir</i>		<i>ortaklık</i>	<i>+sh</i>		<i>oluş</i>	<i>+dhr</i>	<i>+hl</i>	<i>+yacak</i>	<i>+dhr</i>	
<i>conclusion</i>	<i>+s</i>		<i>of</i>		<i>the</i>	<i>basis</i>	<i>on</i>		<i>a</i>		<i>partnership</i>			<i>draw up</i>	<i>+ed</i>		<i>will</i>	<i>be</i>

⇒ Split Turkish into morphemes, drop some



- Basic structure of Arabic morphology

[CONJ+ [PART+ [al+ BASE +PRON]]]

- Examples for clitics (prefixes or suffixes)
  - definite determiner *al+* (English *the*)
  - pronominal morpheme *+hm* (English *their/them*)
  - particle *l+* (English *to/for*)
  - conjunctive pro-clitic *w+* (English *and*)
- Same basic strategies as for German and Turkish
  - morphemes akin to English words → separated out as tokens
  - properties (e.g., tense) also expressed in English → keep attached to word
  - morphemes without equivalence in English → drop

# Arabic Preprocessing Schemes

**ST** Simple tokenization (punctuations, numbers, remove diacritics)

*wsynhY Alr}ys jwlth bzyArp AlY trkyA .*

**D1** Decliticization: split off conjunction clitics

*w+ synhy Alr}ys jwlth bzyArp <lY trkyA .*

**D2** Decliticization: split off the class of particles

*w+ s+ ynhy Alr}ys jwlth b+ zyArp <lY trkyA .*

**D3** Decliticization: split off definite article (Al+) and pronominal clitics

*w+ s+ ynhy Al+ r}ys jwlp +P<sub>3MS</sub> b+ zyArp <lY trkyA .*

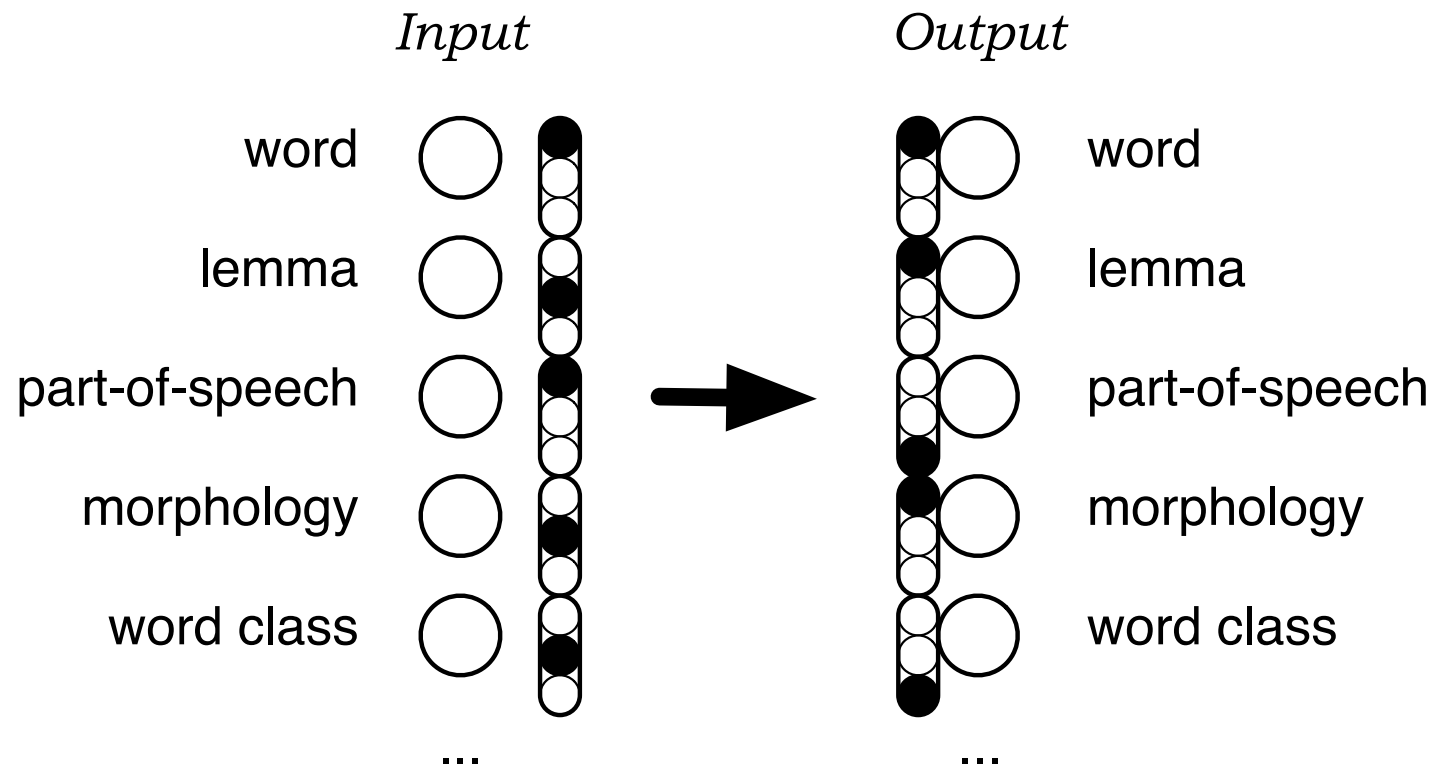
**MR** Morphemes: split off any remaining morphemes

*w+ s+ y+ nhy Al+ r}ys jwl +p +h b+ zyAr +p <lY trkyA .*

**EN** English-like: use lexeme and English-like POS tags, indicates pro-dropped verb subject as a separate token

*w+ s+ >nhY<sub>VBP</sub> +S<sub>3MS</sub> Al+ r}ys<sub>NN</sub> jwlp<sub>NN</sub> +P<sub>3MS</sub> b+ zyArp<sub>NN</sub> <lY trky<sub>NNP</sub>*

- Factored representation of words

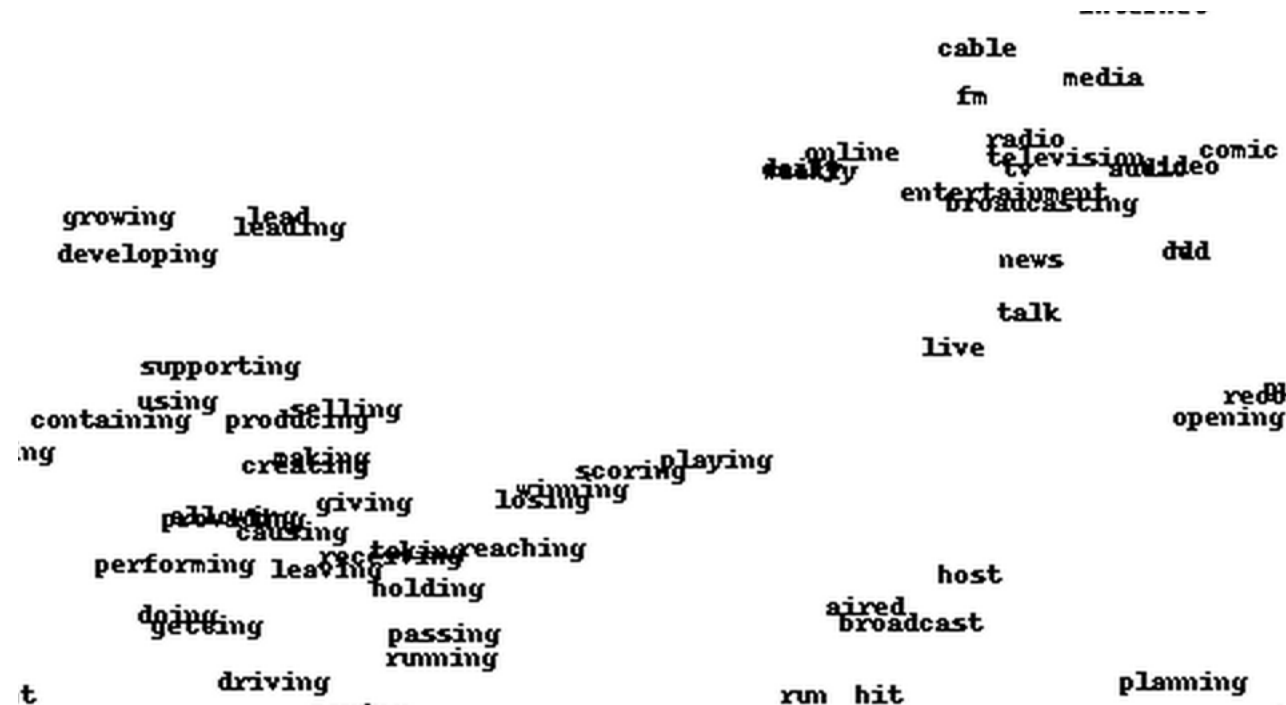


- Encode each factor with a one-hot vector

# word embeddings

# Word Embeddings

20



- In neural translation models words are mapped into, say, 500-dimensional continuous space
- Contextualized in encoder layers

# Latent Semantic Analysis

- Word embeddings not a new idea
- Representing words based on their context has long tradition in natural language processing

# Latent Semantic Analysis

- Word embeddings not a new idea
- Representing words based on their context has long tradition in natural language processing
- Co-occurrence statistics

word	context			
	<i>cute</i>	<i>fluffy</i>	<i>dangerous</i>	<i>of</i>
<i>dog</i>	231	76	15	5767
<i>cat</i>	191	21	3	2463
<i>lion</i>	5	1	79	796

- But: large counts of function words misleading

# Pointwise Mutual Information

- Pointwise mutual information

$$\text{PMI}(x; y) = \log \frac{p(x, y)}{p(x)p(y)}$$

- Intuition: measures how much more frequent than chance

word	context			
	<i>cute</i>	<i>fluffy</i>	<i>dangerous</i>	<i>of</i>
<i>dog</i>	9.4	6.3	0.2	1.1
<i>cat</i>	8.3	3.1	0.1	1.0
<i>lion</i>	0.1	0.0	12.1	1.0

- Similar words have similar vectors



# Singular Value Decomposition

- Raw co-occurrence statistics matrix is very sparse

⇒ Reduce into lower dimensional matrix

# Singular Value Decomposition

- Raw co-occurrence statistics matrix is very sparse

⇒ Reduce into lower dimensional matrix

- Factorize the PMI matrix  $P$  into
  - two orthogonal matrices  $U$  and  $V$   
(i.e.  $UU^T$  and  $VV^T$  are an identity matrix)
  - diagonal matrix  $\Sigma$   
(i.e., it only has non-zero values on the diagonal)

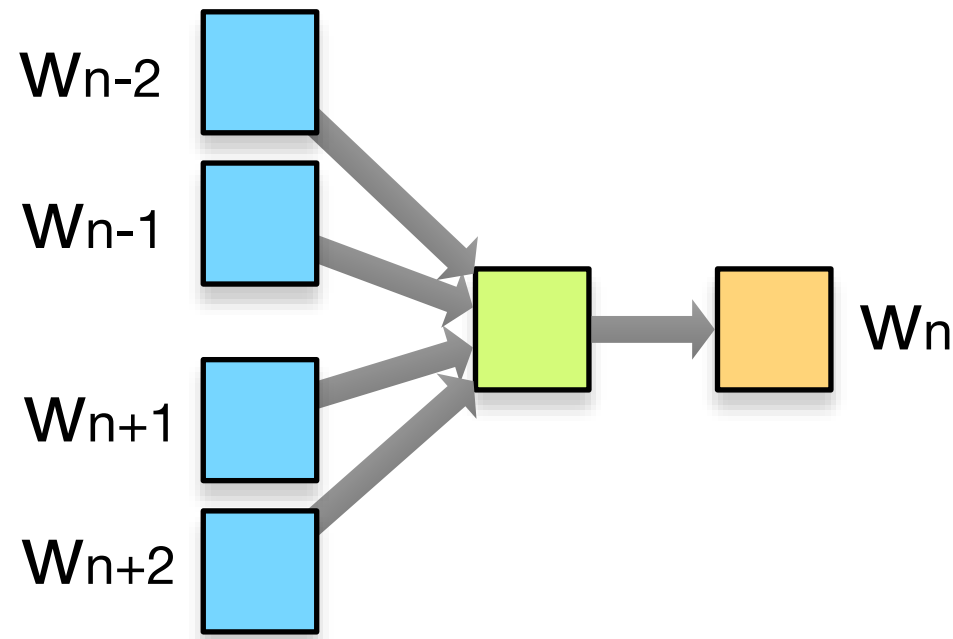
$$P = U\Sigma V^T$$

# Singular Value Decomposition

$$\begin{array}{ccccccc} P & & U & & \Sigma & & V^T \\ \boxed{\text{dots}} & = & \boxed{\text{dots}} & \times & \boxed{\text{diag}} & \times & \boxed{\text{dots}} \\ & & \approx & & & & \\ & & \boxed{\text{dots}} & \times & \boxed{\text{diag}} & \times & \boxed{\text{dots}} \end{array}$$

- Not going into details how to compute this
- Geometric interpretation: rotation  $U$ , a stretching  $\Sigma$ , and another rotation  $V^T$
- Matrices  $U$  and  $V^T$  play similar role as embedding matrices

# Continuous Bag of Words (CBOW)



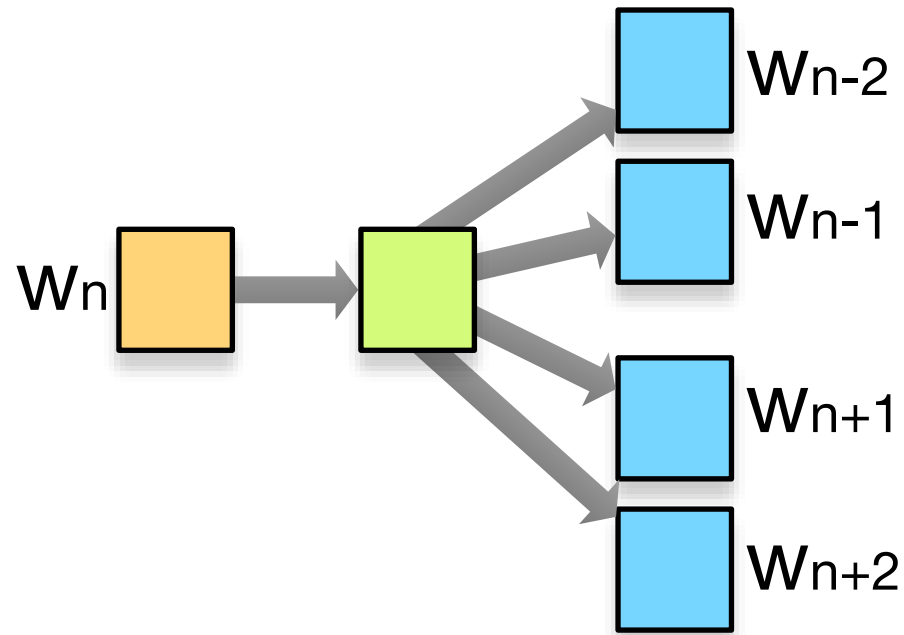
- Predict word from context

$$h_t = \frac{1}{2n} \sum_{j \in \{-n, \dots, -1, 1, \dots, n\}} Cw_{t+j}$$

$$y_t = \text{softmax}(Uh_t)$$

- Similar to n-gram language model

# Skip Gram



- Predict context from word

$$y_t = \text{softmax}(UCw_t)$$

- $C$  input word embedding matrix,  $U$  output word embedding matrix

- Global Vectors: use co-occurrence statistics

word	context			
	<i>cute</i>	<i>fluffy</i>	<i>dangerous</i>	<i>of</i>
<i>dog</i>	231	76	15	5767
<i>cat</i>	191	21	3	2463
<i>lion</i>	5	1	79	796

- Predict the values in this matrix  $X$ , using target word embeddings  $v_i$  and context word embeddings  $\tilde{v}_j$

$$\text{cost} = \sum_i \sum_j \tilde{v}_j^T |v_i - \log X_{ij}|$$

- Training: loop over all words, and their context words

# Refinements

- Bias terms  $b$  and  $\tilde{b}$

$$\text{cost} = \sum_i \sum_j |b_i + \tilde{b}_j + \tilde{v}_j^T v_i - \log X_{ij}|$$

# Refinements

- Bias terms  $b$  and  $\tilde{b}$

$$\text{cost} = \sum_i \sum_j |b_i + \tilde{b}_j + \tilde{v}_j^T v_i - \log X_{ij}|$$

- Most word pairs  $(i, j)$  meaningless, especially for rare words



# Refinements

- Bias terms  $b$  and  $\tilde{b}$

$$\text{cost} = \sum_i \sum_j |b_i + \tilde{b}_j + \tilde{v}_j^T v_i - \log X_{ij}|$$

- Most word pairs  $(i, j)$  meaningless, especially for rare words
- Discount them with a scaling function

$$f(x) = \min(1, (x/x_{\max})^\alpha)$$

hyper parameter values, e.g.,  $\alpha = \frac{3}{4}$  and  $x_{\max} = 200$

- Bias terms  $b$  and  $\tilde{b}$

$$\text{cost} = \sum_i \sum_j |b_i + \tilde{b}_j + \tilde{v}_j^T v_i - \log X_{ij}|$$

- Most word pairs  $(i, j)$  meaningless, especially for rare words
- Discount them with a scaling function

$$f(x) = \min(1, (x/x_{\max})^\alpha)$$

hyper parameter values, e.g.,  $\alpha = \frac{3}{4}$  and  $x_{\max} = 200$

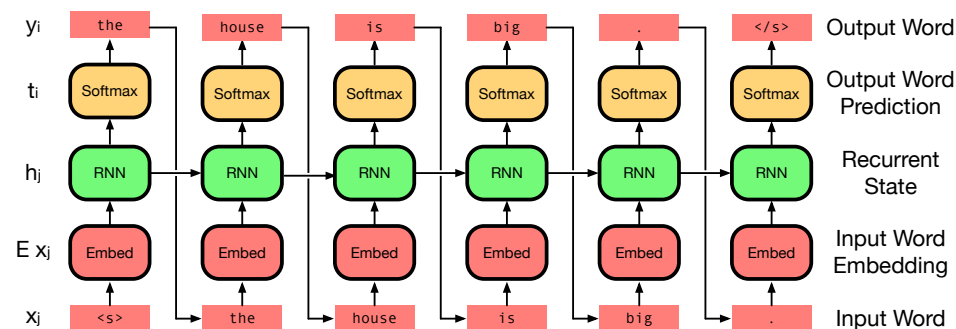
- Complete refined cost function

$$\text{cost} = \sum_i \sum_j f(X_{ij})(b_i + \tilde{b}_j + \tilde{v}_j^T v_i - \log X_{ij})^2$$

- Word embeddings widely used in natural language processing
- But: better refine them in the sentence context

⇒ *Embeddings from language models (ELMo)*

(we have always done this in the encoder of our neural translation models)



- Several layers, use weighted sum of representations at different layers
  - syntactic information is better represented in early layers
  - semantic information is better represented in deeper layers.

- Contextualized word embeddings with Transformer model
- Masked training

The quick brown fox jumps over the lazy dog.



The quick MASK fox MASK over the lazy dog.

- Next sentence prediction

Each unhappy family is unhappy in its own way.



All happy families are alike.

- Essentially BERT, but bigger
- Model: Transformer
  - 175 billion parameters
  - 96 layers
  - 12288 dimensional representations
  - 96 attention heads
- Training
  - trained on about 500 billion word data set, less than 1 epoch
  - 3640 petaflop/s-days on NVIDIA V100 (each can do 0.1 petaflops)
- There currently seems to be not plateau: bigger is better

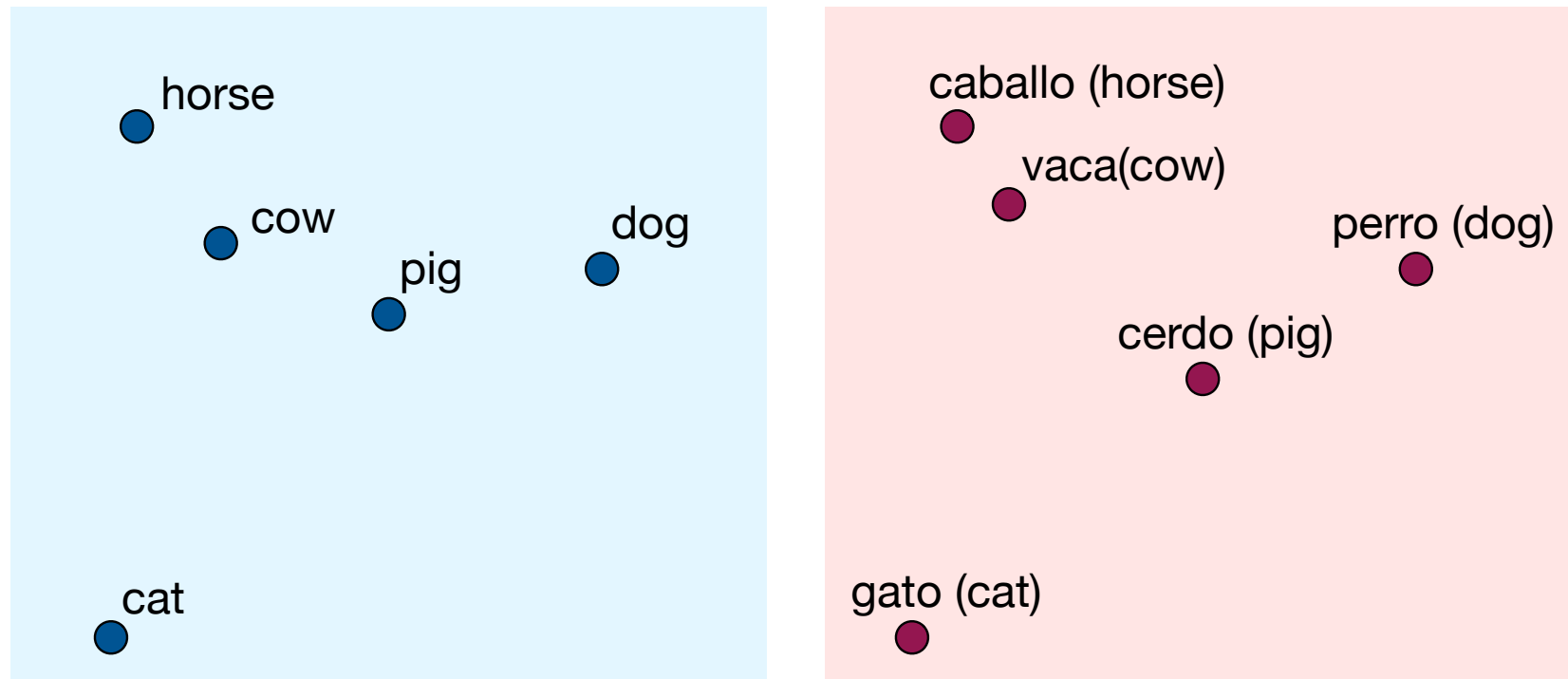


# multi-lingual word embeddings

# Multi-Lingual Word Embeddings

- Word embeddings often viewed as semantic representations of words
- Tempting to view embedding spaces as language-independent  
*cat* (English), *gato* (Spanish) and *Katze* (German) are mapped to same vector
- Common semantic space for words in all languages?

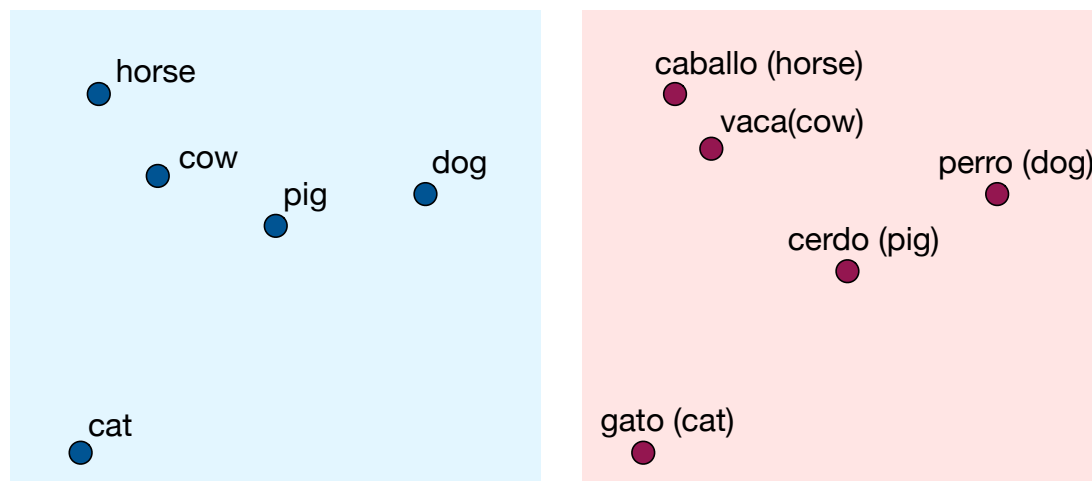
# Language-Specific Word Embeddings



- Train English word embeddings  $C_E$  and Spanish word embeddings  $C_S$



# Mapping Word Embedding Spaces

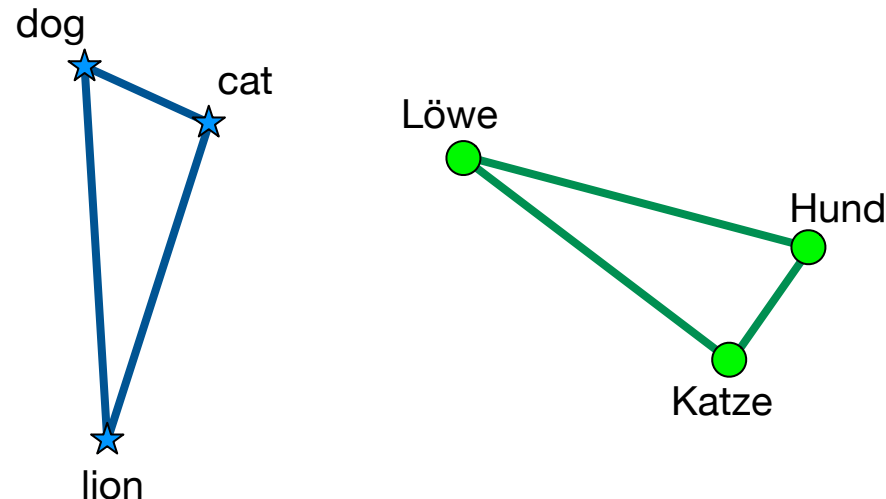


- Learn mapping matrix  $W_{S \rightarrow E}$  to minimize Euclidean distance between each word and its translation

$$\text{cost} = \sum_i ||W_{S \rightarrow E} c_i^S - c_i^E||$$

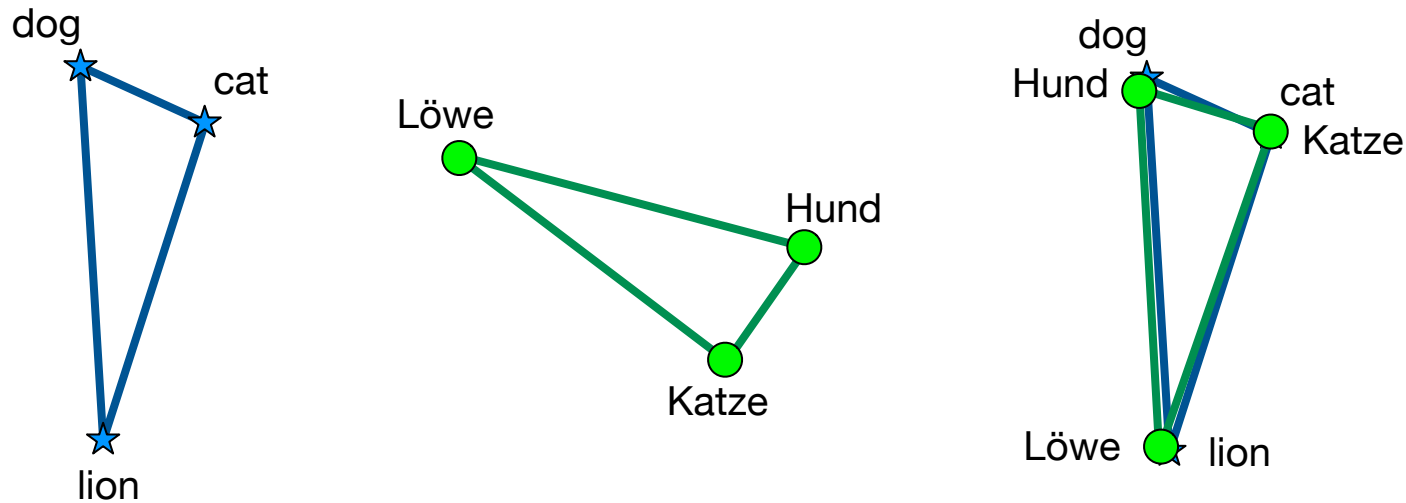
- Needed: Seed lexicon of word translations (may be based on cognates)
- Hubness problem: some words being the nearest neighbor of many words

# Using only Monolingual Data



- Learn transformation matrix  $W_{S \rightarrow E}$  without seed lexicon?
- Intuition: relationship between *dog*, *cat*, and *lion*, independent of language
- How can we rotate the triangle to match up?

# Using only Monolingual Data



- One idea: learn transformation matrix  $W_{\text{German} \rightarrow \text{English}}$  so that words match up

# Adversarial Training

- Another idea: adversarial training
  - points in the German and English space do not match up
  - adversary can classify them as either German and English

- Another idea: adversarial training
  - points in the German and English space do not match up
  - adversary can classify them as either German and English
- Training objective of adversary to learn classifier  $P$

$$\text{cost}_D(P|W) = -\frac{1}{n} \sum_{i=1}^n \log P(\text{German}|W g_i) - \frac{1}{m} \sum_{j=1}^m \log P(\text{English}|e_j)$$

- Training objective of unsupervised learner

$$\text{cost}_D(W|P) = -\frac{1}{n} \sum_{i=1}^n \log P(\text{English}|W g_i) - \frac{1}{m} \sum_{j=1}^m \log P(\text{German}|e_j)$$

# large vocabularies

# Large Vocabularies

40



- 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*

# 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)



- 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?

# Some Causes for Large Vocabularies

- Morphology

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

→ morphological analysis?

- Compounding

*homework, website, ...*

→ compound splitting?

# Some Causes for Large Vocabularies

- Morphology

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

→ morphological analysis?

- Compounding

*homework, website, ...*

→ compound splitting?

- Names

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

→ transliteration?

# 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    t h e \_ f a t \_ c a t \_ i s \_ i n \_ t h e \_ t h i n \_ b a g  
a t → at    t h e \_ f a t \_ c a t \_ i s \_ i n \_ t h e \_ t h i n \_ b a g  
i n → in    t h e \_ f a t \_ c a t \_ i s \_ i n \_ t h e \_ t h i n \_ b a g  
t h e → the    t h e \_ f a t \_ c a t \_ i s \_ i n \_ t h e \_ t h i n \_ 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 .



- 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

46



\_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

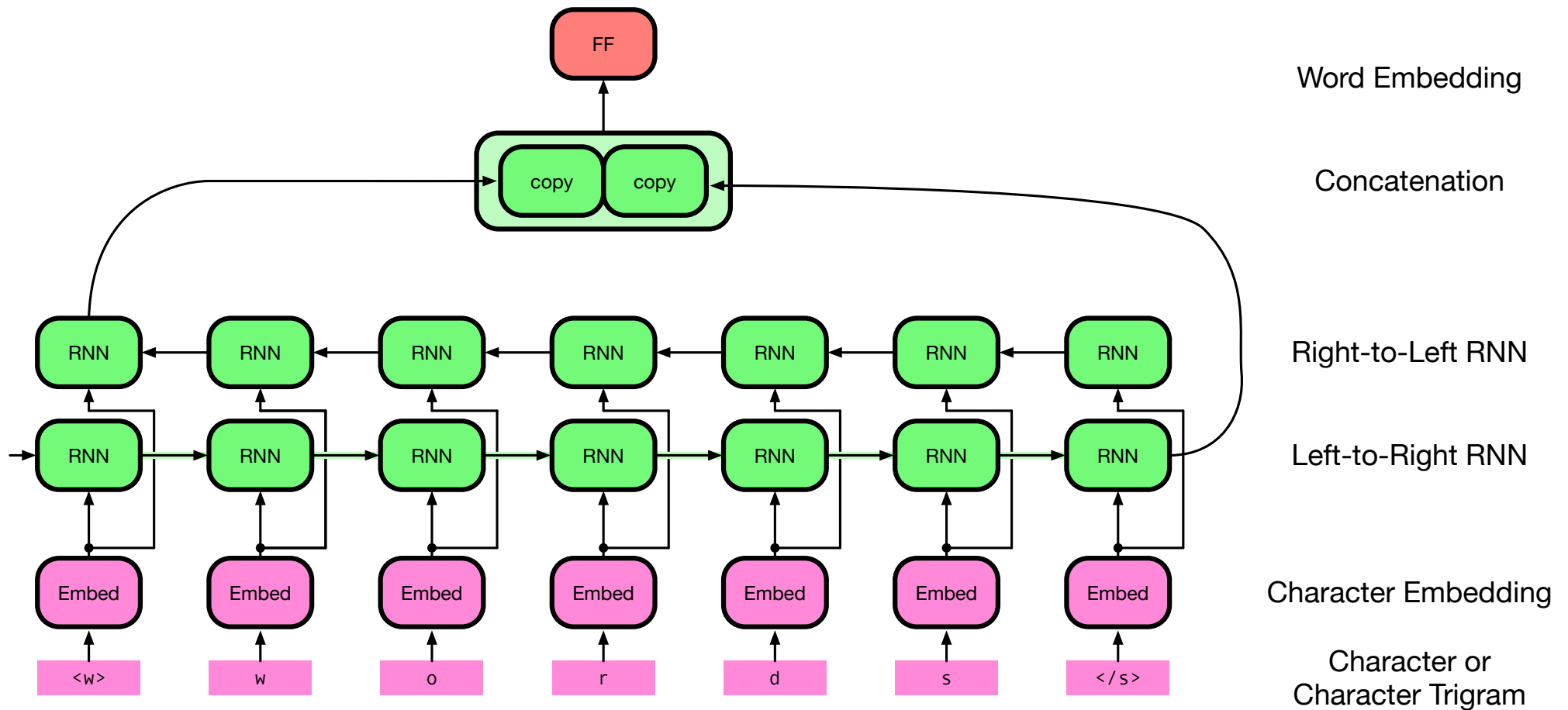
# Character Sequence Models

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

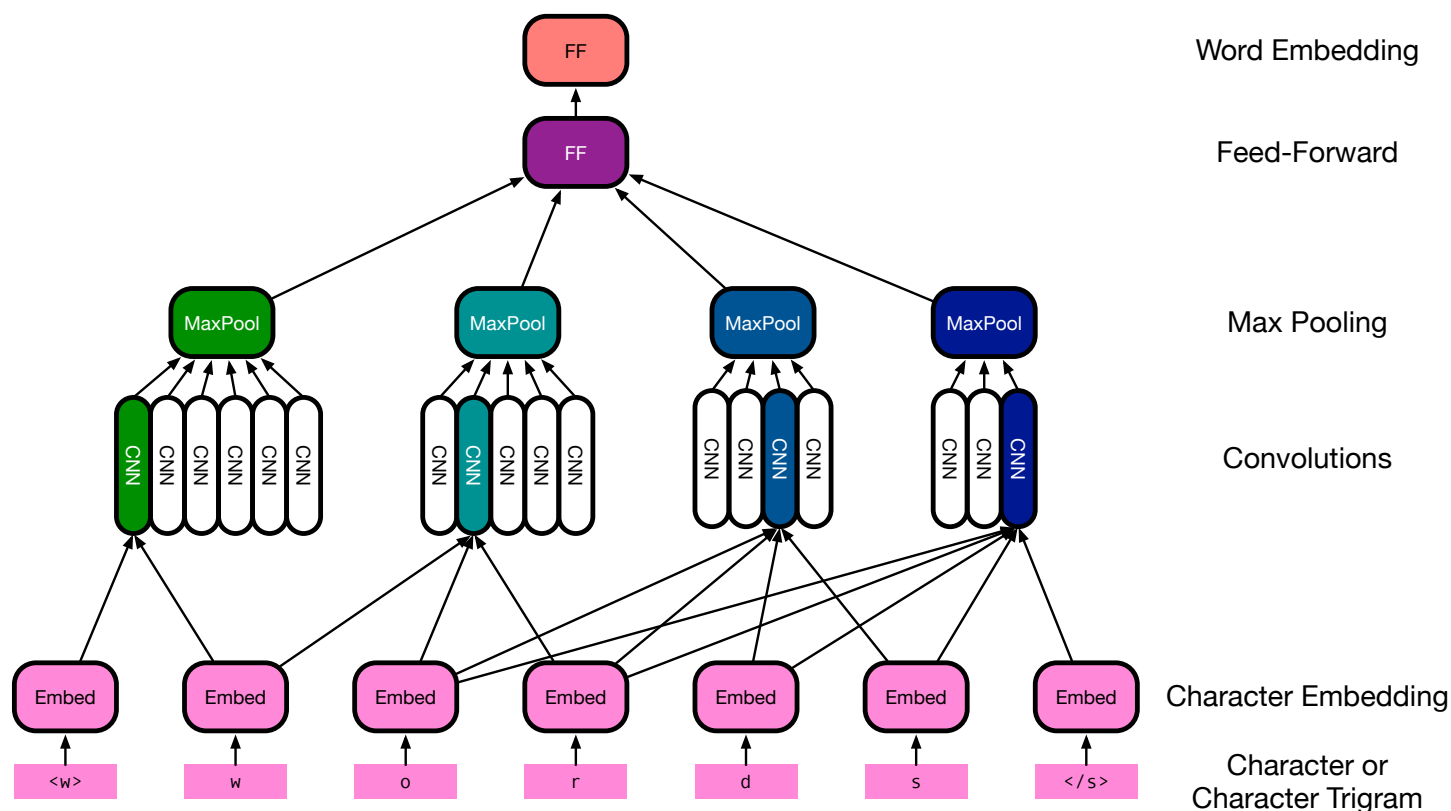
# Character Based Word Models

- Word embeddings as before
- Compute word embeddings based on character sequence
- Typically, interpolated with traditional word embeddings

# Recurrent Neural Networks



# Convolutional Neural Networks



- Convolutions of different size: 2 characters, 3 characters, ..., 7 characters
- May be based on letter n-grams (trigrams shown)