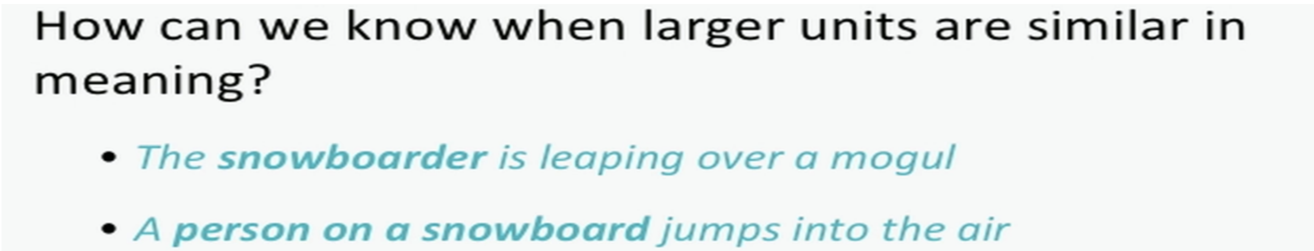
# tree recursive nets and constituency parsing

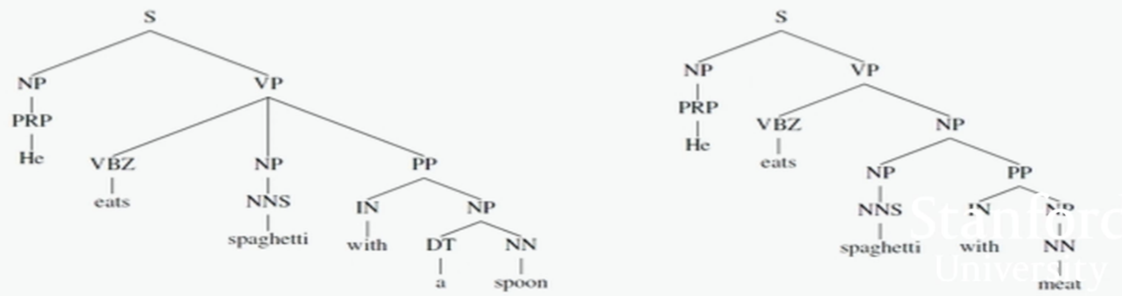
- “I believe languages have a tree structure” - chris manning
- look at these two sentences




- a human knows snowboarder is same as person on a snowboard. because human understands structure and composition of language here. if you just use statistical methods its hard for a computer to figure out this structure
- recursive neural nets offer a prior that can help to model these kinds of **structure and composition** in language → semantic compositionality
- in general, structure and composition are not just in language but also in life in general. images have composition and structure. buildings, stadiums cars are all composition of subparts. people in images too.
- recursive neural nets are also called tree RNNs
- Noam Chomsky thinks languages are recursive.
- some linguistic basics
  - pronouns → it they them he she
  - nouns → things, people, buildings, animals etc
  - verbs → sat, walked, played, other actions
  - preposition → a word governing, and usually preceding, a noun or pronoun and expressing a relation to another word or element in the clause, as in “the man *on* the platform,” “she arrived *after* dinner,” “what did you do it *for* ?”

# Are languages recursive?

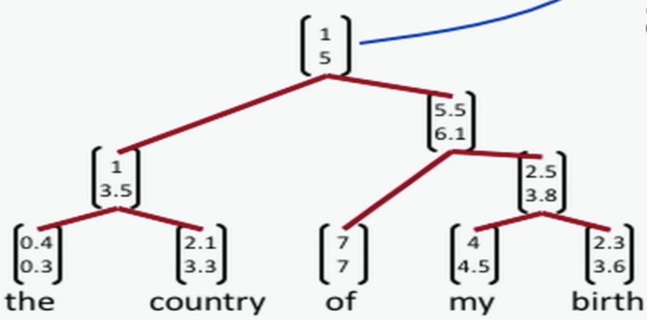
- Cognitively somewhat debatable
- But: recursion is natural for describing language
- *[The man from [the company that you spoke with about [the project] yesterday]]*
- noun phrase containing a noun phrase containing a noun phrase
- Arguments for now: 1) Helpful in disambiguation:



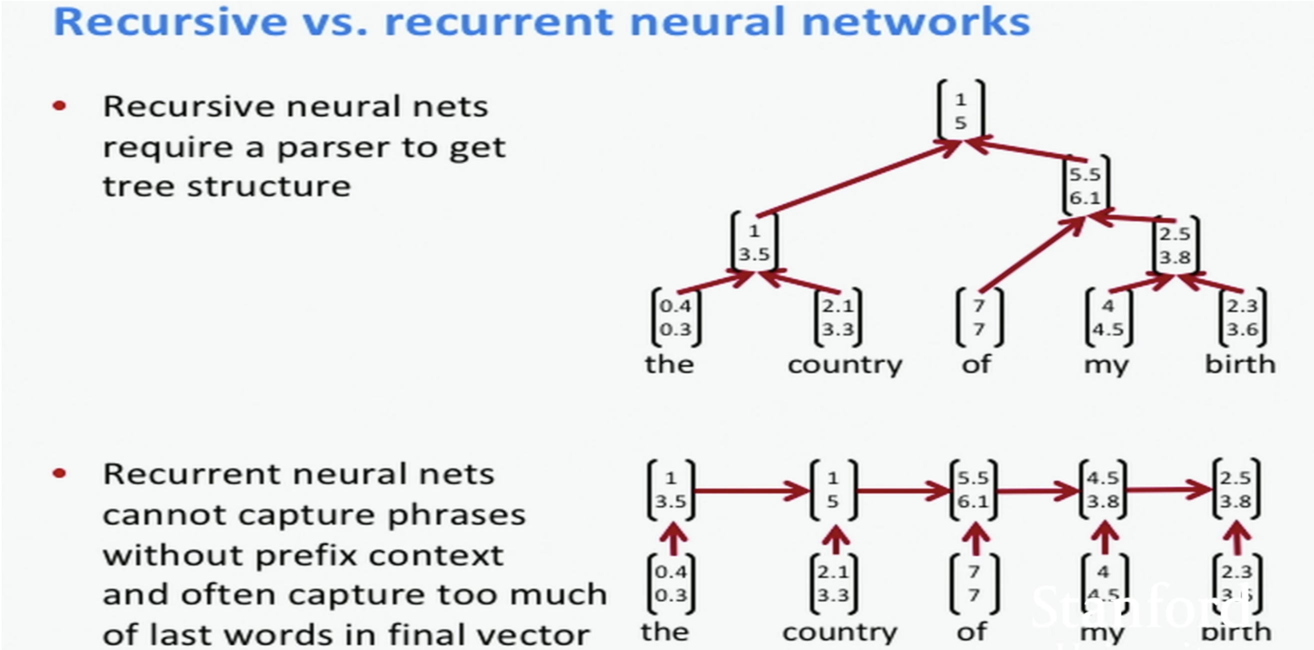



# How should we map phrases into a vector space?

| Use principle of compositionality |
| --- |
| The meaning (vector) of a sentence is determined by<br>(1) the meanings of its words and<br>(2) the rules that combine them. |

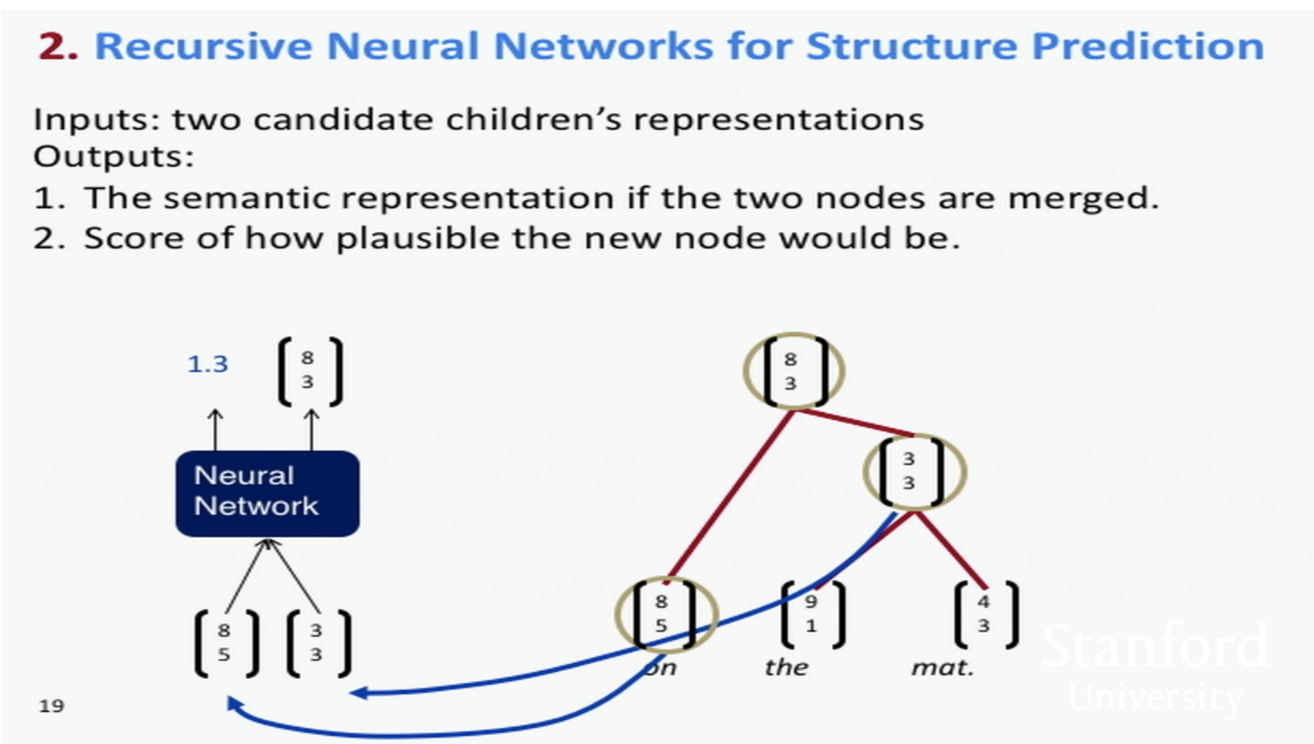

Models in this section can jointly learn parse trees and compositional vector representations



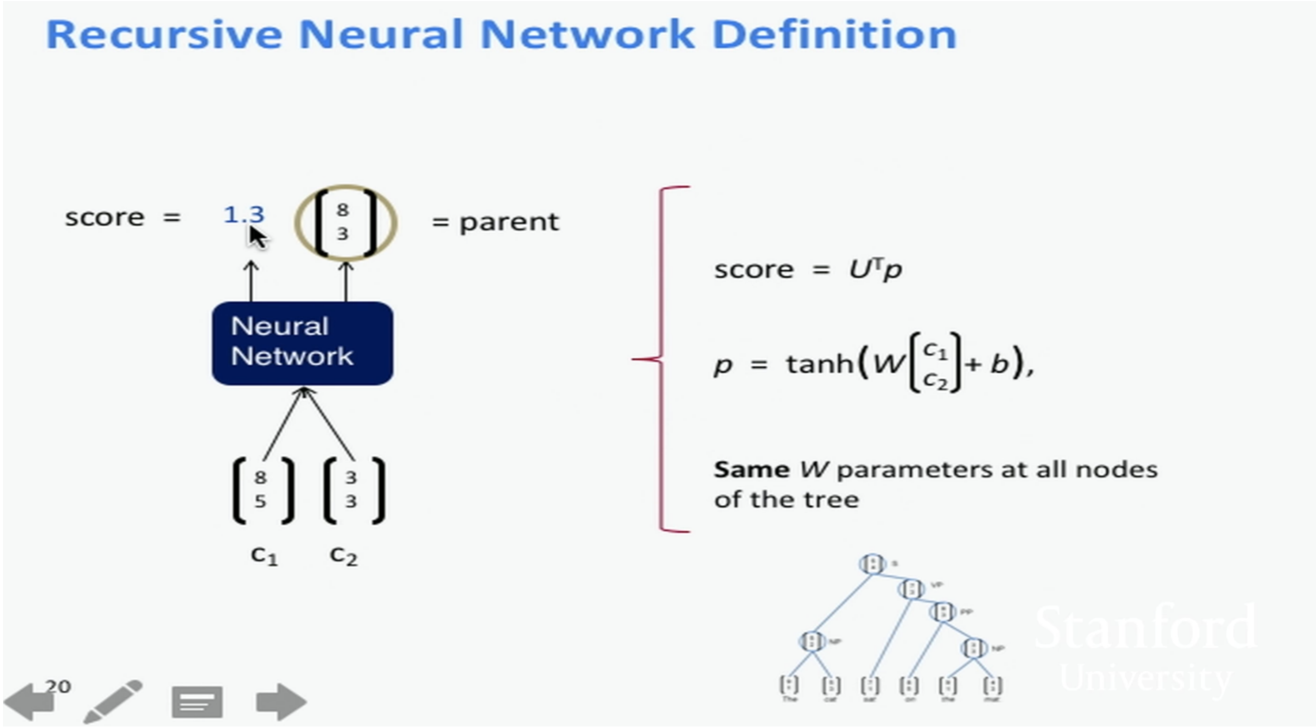

- do recursive nets offer too strong a prior from which you can't escape out if its not the right prior? maybe
- while conv nets run convolutions over all possible bigrams, trigrams etc recursive nets use a parser and only build high level compositions from semantically meaningful phrases that make structural sense, not all possible n-grams

Using a tree RNN to build a sentence embedding in a principled manner using semantic composition of parts of a sentence:

  - train a network to output phrase embedding and whether that makes sense as a phrase semantically.




  - details:




- note **same W params at all nodes** of the tree
- now greedily extract phrases that make sense according to this score and build a parent node whose embedding is the combined embedding output by the above neural net.
- watch this 20 sec video
- we could also use a beam search to look a bit ahead to see what compositions get the best scores