mohit

Week 10 - Tree Recursive Neural Networks

Constituents

Overview

- The structure of a sentence is defined by a grammar (e.g. Context Free Grammars)
- A language can be thought of as a dictionary (words) and a grammar (structure)
- Constituent is a phrase within a sentence which forms a unit of meaning (a phrase)
- Can be thought as a production rule from CFG in Computer Science
- Problem: parse a sentence

Context Free Grammar

- Models the possible structures a constituent can take
- Gives rise to a parse tree which can be used to understand the meaning of the phrase in a structured manner
- Can be hierarchically embedded (to a point)

Representing Constituents (phrases)

- Borrow from word2vec representations
- "Hash" the lexicon (words) and grammar (structure) into d-Space
- A sentence is a tree of constituents

Learning Structure

Tree Recursive Neural Network

- Train to know "what types of words fit together"
- Designed to work with recursive data structures
- Outputs a combined representation of Left and Right element (combined representation) and a score (should these two go together)
- Used by greedily merging repeatedly by highest score for all possible merges at this state

Tree Score

- Score can be generalized as the sum of scores of all current nodes of the tree construction
- Can be compared with the best score (the correct parsing of the tree) to compute a loss score for the tree
- However we don't have access to this oracle giving the best score
- Solution: use Beam Search keeping n best possible partial trees, and always make progress on the best so far
- Similar to Monte Carlo Tree Search (approximate tree search) up the parse tree

Syntactically-Untied RNN

- Previous structure (Syntactically-Untied) treats all words as the same part of speech
- However, partial parses inform us of the "best-guess" of what part of speech each word belongs to
- Why throw away this information?
- Could have a weight matrix for each combination of semantic elements defined by parse rules of the grammar
- However, incredibly data consumptive for all the different matrices

Matrix-vector RNNs

- A combination of things (word, pair of words, phrase, etc.) can be thought of as a thing (word, pair of words, phrase, etc.) and something that modifies it
- A matrix operates on vectors, so vectors can be things, and matrices modifiers
- However, we don't know if a phrase is a thing or a modifier without an oracle
- So we need to represent each phrase as a vector (in case it's a thing) and a matrix (in case it's a modifier)
- Now in combining them we need to form another vector and a matrix
- The vector part is formed by combining vector and a matrix (a thing formed by modifier and thing)
- The matrix part is formed by combining matrices (a modifier formed by combining modifiers)
- Same other processes as Syntactically-Untied RNNs (beam search, score functions, and other general considerations)

Class Discussion

1. What forms a language?
    a. A language can be thought of as a dictionary, and a grammar.
    b. Many words in the dictionary can be multiple parts of speech (noun, verg, advjective, etc.).
    c. The grammar helps decide which parts of speech the word takes on in this instance.
    d. The grammar defines the sets of rules that can be used to determine which phrases are admissible in a given language.
2. Are languages recursive?
    a. Recursion can be thought of as a natural feature of human language (embedding noun-verb phrase within a large phrase and so on.)
    b. However, may lead to overly complicated sentences not heard in vernacular or overly nesting recursive structure.
3. Context Free vs Context Sensitive Grammar
    a. Context Free Grammar is defined as a set of rules where the production rules are always independent (the LHS is always unary).
    b. Context Sensitive Grammar is defined in a similar way, but the LHS can be non-unary (so replacement depends also on neighbours).
4. "Hashing" constituents
    a. A constituent is a bag of words, and a structure.
    b. However, how best to represent the constituent.
    c. The parts of speech and the ordering help.
    d. For example "the red car" is a type of car, so the phrase representation of "car" should be close in d-Space to "the red car."
    e. "Country of my birth" should be close to "Germany" or "France," which are countries.
5. Tree Recursive Neural Network similarities to agglomerative clustering.
    a. Both are trying to build a "tree" of structure to explain something.
    b. Both can be thought of as a form of dynamic programming.
    c. Both are using greedy search as a best effort heuristic.
6. What to do when we don't have an oracle to guide the loss function towards the best parsing?
    a. Play a "game" against the adversary of "worst possible loss" in case of making greedy choices repeatedly.
    b. Use Beam Search (essentialy Monte Carlo Tree Search against this adversary).
7. The state of machine learning.

a. "Structured" neural networks are a harder problem, not getting much attention now.
b. There is too much low hanging fruit at the moment.
c. However, more difficult problems will need to be looked at eventually.

Additional Reading

- Recent paper on Constituency parsing: https://arxiv.org/abs/1805.01052
- Chomsky Hierarchy: https://en.wikipedia.org/wiki/Chomsky_hierarchy
- Garden-path sentence: https://en.wikipedia.org/wiki/Garden-path_sentence
- Universal Grammar: https://en.wikipedia.org/wiki/Universal_grammar
- Recursive Neural Network: https://arxiv.org/abs/0911.3298
- Beam Search: https://en.wikipedia.org/wiki/Beam_search
- Monte Carlo Tree Search: https://en.wikipedia.org/wiki/Monte_Carlo_tree_search
- Matrix Vector Spaces: https://ai.stanford.edu/~ang/papers/emnlp12-SemanticCompositionalityRecursiveMatrixVectorSpaces.pdf