Week 8

Word Embedding

- What is a word embedding? How long are they usually?
  - A parameterized function mapping words in some language to a high-dimensional vector space.
- How does training a network to recognize the validity of 5-grams result in a Word-to-Vector "map"?
  - Because the network learns to associate words with other words, so the result will become a word-to-vector map where the vector is contains the related words.
- Can you think of another training method to achieve the same side-effect?
  - Using RL, we can train the network to associate words with other words by giving it a reward for associating words with other words.
- Pretend you are a word embedder. Give examples (2-3 for each) of words in the same family as:
  - King: queen, prince, princess
  - Button: zipper, snap, hook
  - Pain: hurt, ache, sore
  - Water Bottle: cup, glass, mug
- What is the explanation given for the emergence of the "male-female difference vector"?
  - Sentences often contain gender-specific pronouns, therefore switching male and female terms will maek the sentence nonsensical.
- What is pre training/transfer learning/multi-task learning?
  - Pre-training is training a model on a large dataset to learn a general representation of the data.
  - Transfer learning is using the pre-trained model to learn a specific representation of the data.
  - Multi-task learning is training a model on multiple tasks at the same time.

RNN

- What does a language model, in general, try to predict?
  - Hint: Predicting X given Y. What are X and Y?
  - Predict the next word based on the previous words.
- What happens to the memory vector as we move through time?
  - It is modified so that it contains relevant information from the previous words.
- Describe how a RNN would deal with the sentence "How are you?" in terms of its unrolled computational graph
  - Basically, what happens to these words and the hidden states generated from these words?
    - A hidden state h_0 is fed into the word "how" to generate an

output o_1 and a new hidden state h_1.
        * The new hidden state h_1 is fed into the word "are" to generate an output o_2 and a new hidden state h_2.
        * And so on
    - How long would the unrolled computational graph be in terms of RNN nodes (circles)?
        * There will be 4 (or 3 if not counting ? as a word) RNN nodes.
- What is the memory vector initialized to?
    - 0 vector
- What is `<\s>`? What does it signify according to the article?
    - Beginning of sentence

BPTT

- What are W_x, W_y, and W_s?
    - W_x is the weight matrix for the input layer.
    - W_y is the weight matrix for the output layer.
    - W_s is the weight matrix for the hidden layer.
- Why does BPTT not work with a large number of timesteps?
    - Because the gradient becomes too small when backpropagating through too many layers.
- What is this problem called?
    - Vanishing gradient
- How does Truncated BPTT solve this problem?
    - It uses temporal dependencies to backpropagates through a limited number of layers, so the gradient is not too small.

GPT

- Why does not being able to capture long-term dependencies result in nonsensical generated paragraphs (like clicking autocomplete)?
    - Because then the words will only be related to the words directly before and after them, so the generated text as a whole will be nonsensical.
- What do the probabilities that GPT outputs represent, and what is greedy decoding?
    - The probability vector represents the probability of each word being the next word in the output sentence. Greedy decoding is picking the word with the highest probability.
- What is a token?
    - Bits of language, could be letters, words, byte-pairs, etc.
- What are the big blocks that make up the GPT architecture?
    - Transfers, decords, and token embeddings
- What are the two main blocks inside a transformer-decoder block?
    - Masked self-attention and feed-forward neural network

- Describe masked self-attention in your own words (not including the vector math).
  - A mechanism that allows the model to place "focus" on certain words when predicting the next word.
- How would this help stop generated sequences from being mostly nonsensical?
  - This allows the model to have a vague grasp of the context of the sentence and the expected next word.
- How are Query, Key, and Value vectors generated from each word embedding?
  - Query, key, and value vectors are generated by multiplying the word embedding by their respective weight matrices.
- How is the score for each word calculated using one word's query vector and all the other words' key vectors?
  - The score is calculated using the dot product of query and key to get a scalar score.
- How is the score for each word and its value vector used to create the vector for a single transformed word? (In the case of the article, the word is "He").
  - Multiply each word's value vector by its score then sum them up to get the transformed word vector.
- Briefly describe multi-headed attention
  - It allows focus on multiple words at the same time.
  - It is achieved by splitting the query, key, and value vectors into multiple heads, and then performing the same operations on each head.

Summary Questions:

- Show some text that you finetuned model generated!

  Clown:

  I am clown!

  Clowness:

  Why, no, no; I am not a tapster.

  Shepherd:

  Clubs and such; but I am a tapstool.

  What, is it good to be a tapsters' bird?

  AUTOLYCUS:

  Good to be so, to be good to me, I know not.

  Clog and such, I am too fond:

  But I am no tapster, I have a score of them.

I have a dozen that are as many as you have:

And you have to

I'll be your father, and not your mother;

For I am a widow, and you a wife.

KING RICHARD III:

What, doth your husband live?

DUKE OF AUMERLE:

Ay, if he die to-morrow,

And you are not heir to his daughter.

I'll have you wed again, and that's enough.

What say you, then, to the prince your son?

Tell me, my son, and tell me, your husband:

Tell him, my daughter, and his

Washington:

I will not be so long in saying.

First Senator:

You are a senator, and you must be so.

You have been forsworn to the people's house:

And you have not been so long to demand it.

Your honours both,

SICINIUS:

We have been so forswaken, and so long.

I would be so brief to say the truth.

The people are not so much in their love

As they are in fearing to hear me tell it.

They are not as much in fear

- What would you like to explore further?
  - Other applications of transformers like computer vision.
  - Generating a text generator from my own text history.
  - Persistent memory and outside-of-input context for generating a response.

  - How exactly the attention mechanism works, like how the array is formed etc.

- What is a resource you read and was there anything interesting in it?
  - Carter's HF tutorial jupyter notebook.
  - The Annotated Transformer, still reading but looks fantastic.
  - Hugging face course