Question answering:

MCTest:it is a corpus of about 600 reading comprehension with questions which can be attempted by the model and humans can evaluate it. Therefore it can only be a test set.

SQuAD: (Stanford Question Answering Dataset)

It is a dataset developed in 2016 that can be used for training. In SQuAD the answer to a question is always a subsequence of words called a span.
SQuAD evaluation:

1. Exact match: 1 for exact match with one of 3 saved gold spans (Gold span is answer given by humans), then calculate the percentage.
2. F1 metric: do word for word match and calculate precision: percent of words in system span which are in gold span, recall:percent of words in gold span which are in system span and take harmonic mean of the two.

Note: both ignores punctuations and articles like a, an, the.

SQuAD 2.0: problem with SQuAD 1.0 is that many of the questions didn't have answers, that problem was solved on SQuAD 2.0 where 1/3 of the training data didn't have answers and 1/2 of test/dev didn't.

Important problem with SQuAD: the questions were made as the people were staring at the paragraph. Therefore the sentence structure is very similar to the one in the paragraph, but in the real world application this might not be the case.

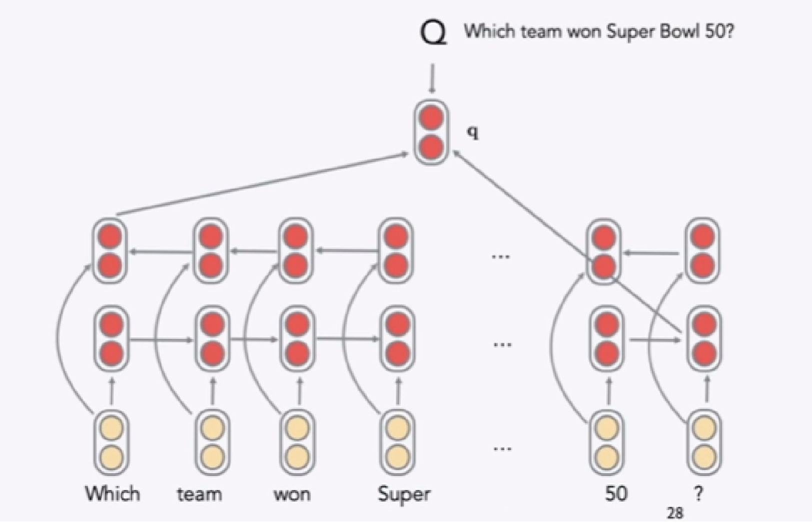
Factoid question answering: this is the type of answering for which the answer is a named entity. Example: who is the 7th pm of Australia.

Stanford Attentive Reader:

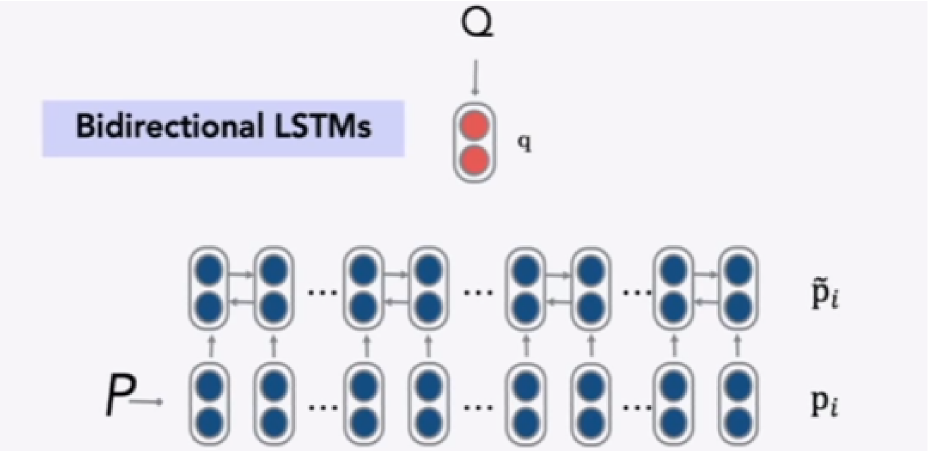
It is a relatively simple model that achieves very good results for question answering.

In it the question is sent through a bidirectional RNN and the concatenated vector of the last step of the forward direction and the reverse direction is concatenated and taken as a vector that represents the question. This vector along with a vector that represents each hidden state is multiplied with a leaned weight matrix and passed through softmax to get an attention distribution and this represents the start of the span. Similarly another weight matrix is learned which gives the end token of the span.

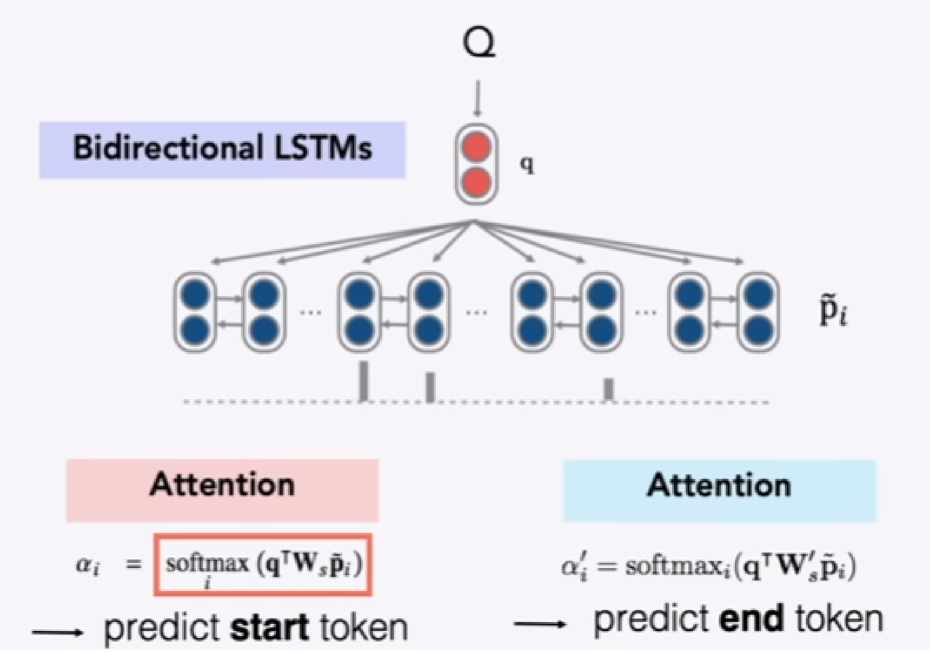
Diagramatic Representation:
Getting the representation of the question:



Similarly run it through a bidirectional LSTM (this time represented shortly)



Now find the attention distribution:



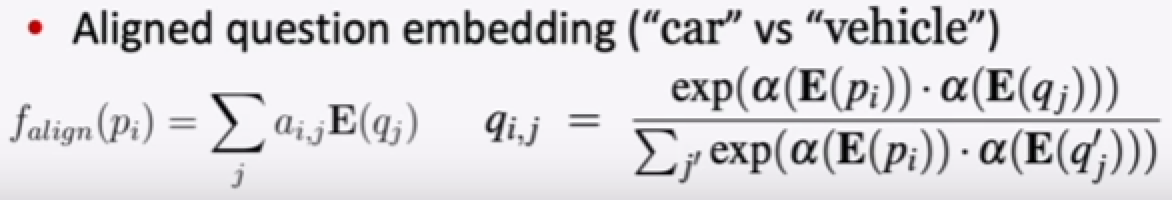
$\alpha_i = \boxed{\operatorname{softmax}_i (\mathbf{q}^\top \mathbf{W}_s \tilde{\mathbf{p}}_i)}$ → predict **start** token

$\alpha'_i = \operatorname{softmax}_i (\mathbf{q}^\top \mathbf{W}'_s \tilde{\mathbf{p}}_i)$ → predict **end** token

Stanford Attentive Reader++

The main change in this are:

- Instead of simply concatenation two end vectors of both the direction, what we do is we take a weighted sum of all the states and the weights are calculated using a “sentinel” matrix and the hidden state.
- A one hot vector indication the character of the word (if it is a name, place etc.,) is calculated and added on top of the word vector along with the word frequency and if there is an exact match of the same word in the question sentence and the Aligned question embedding to represent the word.



- Aligned question embedding (“car” vs “vehicle”)

$$f_{align}(p_i) = \sum_j a_{i,j} \mathbf{E}(q_j) \qquad q_{i,j} = \frac{\exp(\alpha(\mathbf{E}(p_i)) \cdot \alpha(\mathbf{E}(q_j)))}{\sum_{j'} \exp(\alpha(\mathbf{E}(p_i)) \cdot \alpha(\mathbf{E}(q'_j)))}$$

BiDAF (Bi-Directional Attention Flow):

It is an architecture based on attention flowing in both ways, from context (paragraph) to the question and from the question to the context.

BERT: it is used to get contextual word representations using complex methods of attention.