Lecture 11: Question Answering

1. What is question answering?

    The goal of question answering

    -to build systems that automatically answer questions posed by humans in a nuatural language

    -information source: a text passage, all web documents, knowledge bases, tables, images.

    -question type: factoid vs non-factoid, open-domain vs closed-domain, simple vs compositional

    -answer type: a short segment of text, a paragraph, a list, yes

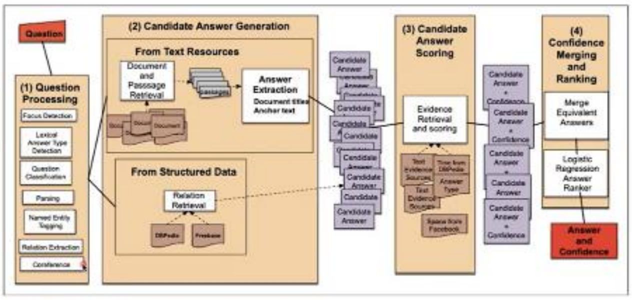


Image credit: J & M, edition 3

(1) Question processing, (2) Candidate answer generation, (3) Candidate answer scoring, and (4) Confidence merging and ranking.

2. Reading comprehension

    Reading comprehension: building systems to comprehend a passage of text and answer questions about its content (P, Q)→A

Stanford question answering datase (SQuAD)

-100k annotated (passage, question, answer) triples

-Passages are selected from English Wikipedia, usually 100~150 words

-Questions are crowd-sourced

-Each answer is a short segment of text (or span) in the passage. → This is a limitation not all the questions can be answered in this way

Evaluation : exact match (0 or 1) and F1 (partial credit)

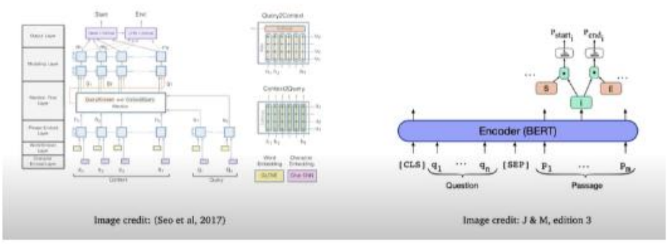
For development and testing sets, 3 gold answers are collected, because there could be multiple plausible answers

We compare the predicted answer to each gold answer and take max scores. Finally, we take the average of all the examples for both exact match an F1.

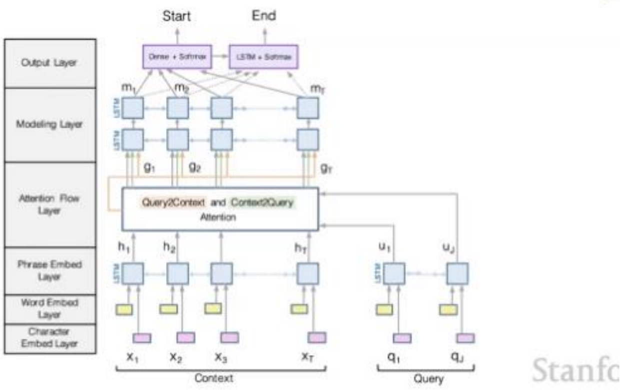
Build a model to solve SQuAD

- Problem formulation
    - input = C=(c1,c2,c3,...,cN)=(q1,q2,...,qM), ci,qi
    - output: 1≤start≤end≤N
    - N~100, M~15

LSTM-based vs BERT models

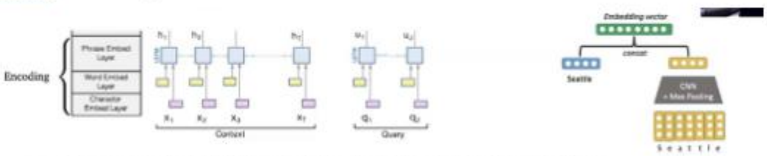


BiDAF: the Bidirectional Attention Flow model



(Seo et al., 2017): Bidirectional Attention Flow for Machine Comprehension

BiDAF: Encoding



- Use a concatenation of word embedding (GloVe) and character embedding (CNNs over character embeddings) for each word in context and query.

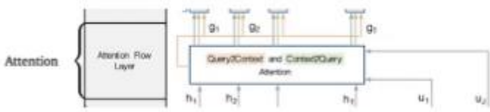
$$e(c_i) = f([\text{GloVe}(c_i); \text{charEmb}(c_i)]) \qquad e(q_i) = f([\text{GloVe}(q_i); \text{charEmb}(q_i)])$$

f: high-way networks omitted here

- Then, use two **bidirectional** LSTMs separately to produce contextual embeddings for both context and query.

$$\overrightarrow{\mathbf{c}}_i = \text{LSTM}(\overrightarrow{\mathbf{c}}_{i-1}, e(c_i)) \in \mathbb{R}^H \qquad \overrightarrow{\mathbf{q}}_i = \text{LSTM}(\overrightarrow{\mathbf{q}}_{i-1}, e(q_i)) \in \mathbb{R}^H$$
$$\overleftarrow{\mathbf{c}}_i = \text{LSTM}(\overleftarrow{\mathbf{c}}_{i+1}, e(c_i)) \in \mathbb{R}^H \qquad \overleftarrow{\mathbf{q}}_i = \text{LSTM}(\overleftarrow{\mathbf{q}}_{i+1}, e(q_i)) \in \mathbb{R}^H$$
$$\mathbf{c}_i = [\overrightarrow{\mathbf{c}}_i; \overleftarrow{\mathbf{c}}_i] \in \mathbb{R}^{2H} \qquad \mathbf{q}_i = [\overrightarrow{\mathbf{q}}_i; \overleftarrow{\mathbf{q}}_i] \in \mathbb{R}^{2H}$$

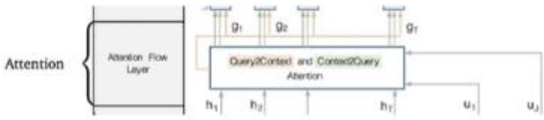
BiDAF: Attention



- Query-to-context attention: choose the context words that are most relevant to one of query words.

While Seattle's weather is very nice in summer, its weather is very rainy in winter, making it one of the most gloomy cities in the U.S. LA is ...

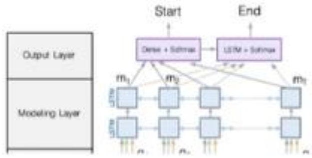
Q: Which city is gloomy in winter?



- First, compute a similarity score for every pair of $(\mathbf{c}_i, \mathbf{q}_j)$:

$$S_{i,j} = \mathbf{w}_{\text{sim}}^{\intercal}[\mathbf{c}_i; \mathbf{q}_j; \mathbf{c}_i \odot \mathbf{q}_j] \in \mathbb{R} \qquad \mathbf{w}_{\text{sim}} \in \mathbb{R}^{6H}$$

BiDAF: Modeling and output layer

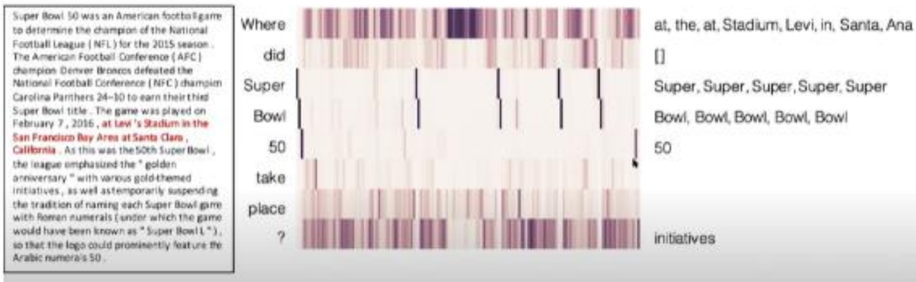


Modeling layer: pass $\mathbf{g}_i$ to another two layers of **bi-directional** LSTMs.

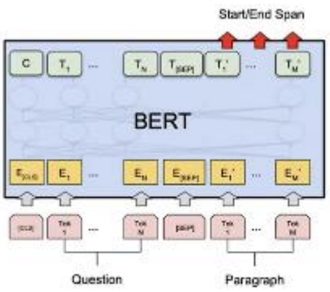
- Attention layer is modeling interactions between query and context
- Modeling layer is modeling interactions within context words

$$\mathbf{m}_i = \text{BiLSTM}(\mathbf{g}_i) \in \mathbb{R}^{2H}$$

attention visualization



BERT for reading comprehension



$$\mathcal{L} = -\log p_{\text{start}}(s^*) - \log p_{\text{end}}(e^*)$$

$$p_{\text{start}}(i) = \text{softmax}_i(\mathbf{w}_{\text{start}}^{\intercal}\mathbf{h}_i)$$

$$p_{\text{end}}(i) = \text{softmax}_i(\mathbf{w}_{\text{end}}^{\intercal}\mathbf{h}_i)$$

where $\mathbf{h}_i$ is the hidden vector of c_i, returned by BERT

Question = Segment A

Passage = Segment B

Answer = predicting two endpoints in segment B

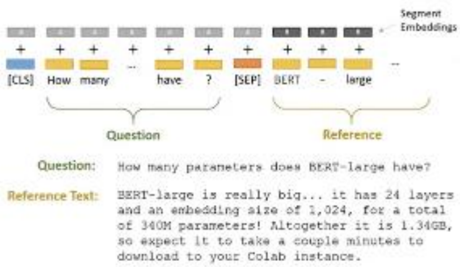


Image credit: https://mccormickml.com/

Stanford

3. Open-domain (textual) question answering
    - Different from reading comprehension, we don't assume a given passage.
    - Instead, we only have access to a large collection of documents. We don't know where the answer is located, and the goal is to return the answer for any open-domin questions.
    - Much more challenging but a more practical problem

Retriver-reader framework