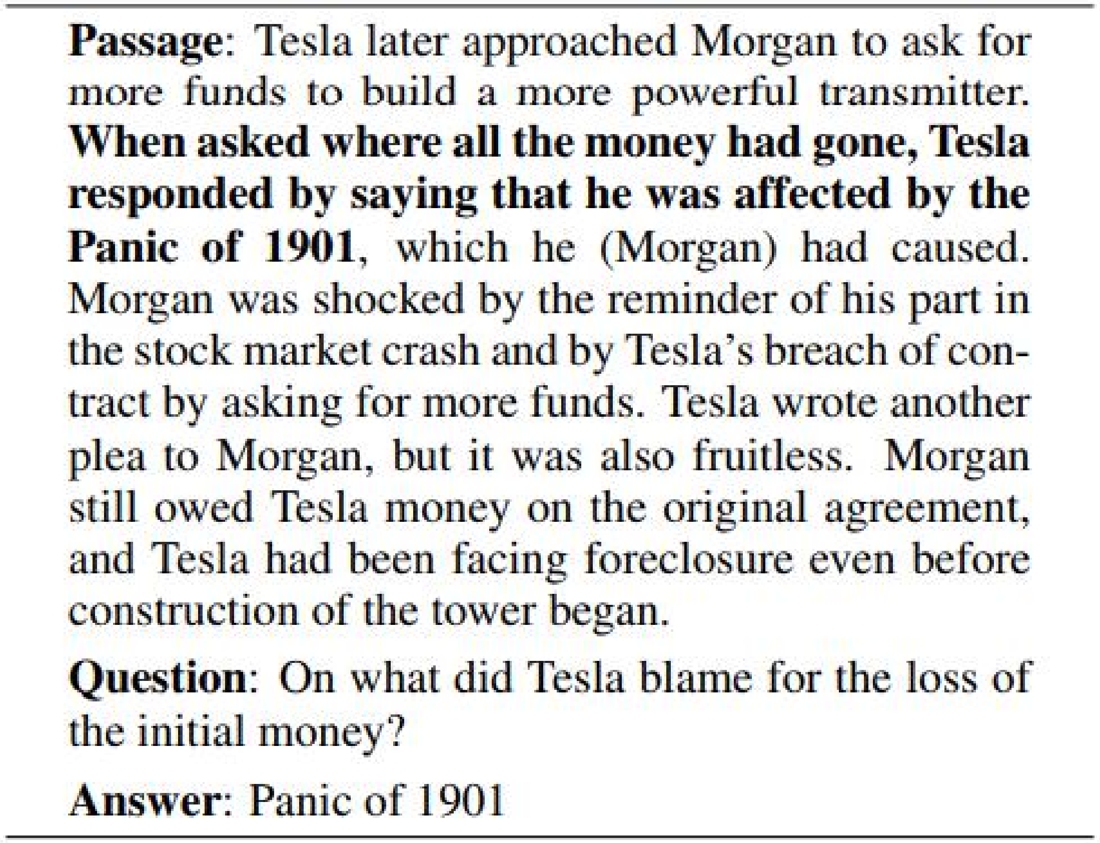
Gated Self-Matching Networks for Reading Comprehension and Question Answering

Stanford Question Answering Dataset (SQuAD)

SQuAD 是最新的阅读理解数据集，该数据集包含 10 万个（问题，原文，答案）三元组，原文来自维基百科，而问题和答案的构建主要是通过众包的方式，提出最多 5 个基于文章内容的问题并提供正确答案，且答案出现在原文中。

SQuAD 和之前的完形填空类阅读理解数据集最大的区别在于：SQuAD 中的答案不在是单个实体或单词，而可能是一段短语，这使得其答案更难预测。

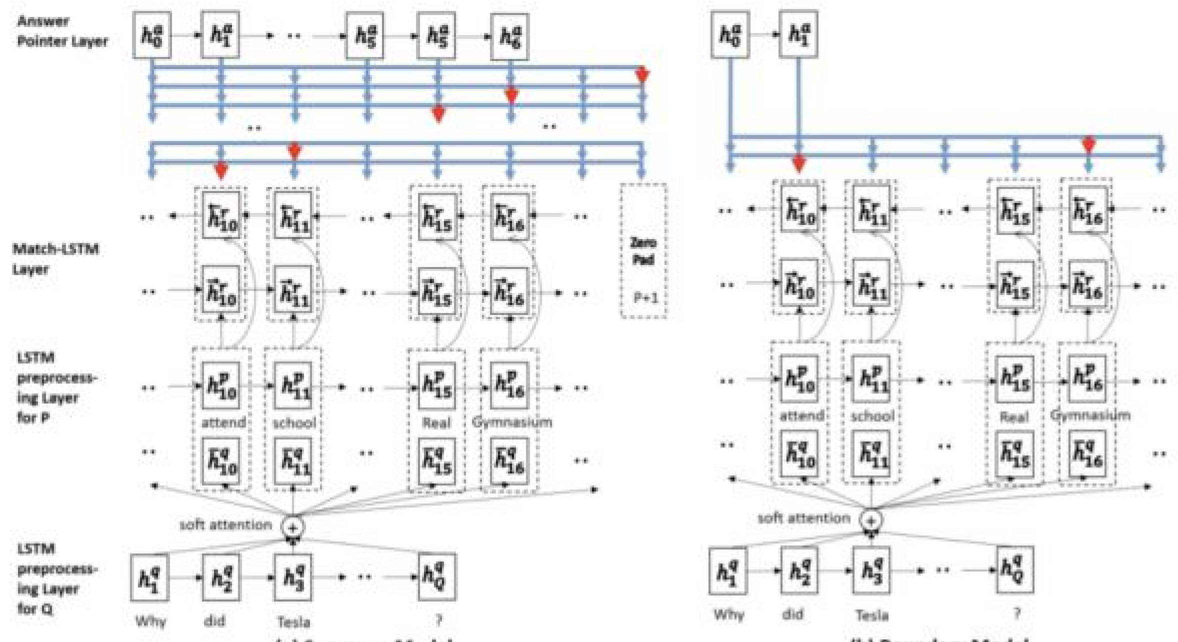


Passage: Tesla later approached Morgan to ask for more funds to build a more powerful transmitter. **When asked where all the money had gone, Tesla responded by saying that he was affected by the Panic of 1901**, which he (Morgan) had caused. Morgan was shocked by the reminder of his part in the stock market crash and by Tesla's breach of contract by asking for more funds. Tesla wrote another plea to Morgan, but it was also fruitless. Morgan still owed Tesla money on the original agreement, and Tesla had been facing foreclosure even before construction of the tower began.

Question: On what did Tesla blame for the loss of the initial money?

Answer: Panic of 1901

Match-LSTM



(a) Sequence Model

(b) Boundary Model

Match-LSTM 的主要步骤：

1) Embed 层使用词向量表示原文和问题；

2) Encode 层使用单向 LSTM 编码原文和问题 embedding

3) Interaction 层对原文中每个词，计算其关于问题的注意力分布，并使用该注意力分布汇总问题表示，将原文该词表示和对应问题表示输入另一个 LSTM 编码，得到该词的 query-aware 表示；

4) 在反方向重复步骤 2，获得双向 query-aware 表示；

5) Answer 层基于双向 query-aware 表示使用 Sequence Model 或 Boundary Model 预测答案范围。

端到端的多层神经网络模型

源于match-LSTM的边界模型

1. The recurrent network encoder to build representation for questions and passages separately
2. The gated matching layer to match the question and passage
3. The self-matching layer to aggregate information from the whole passage,
4. The pointer-network based answer boundary prediction layer.

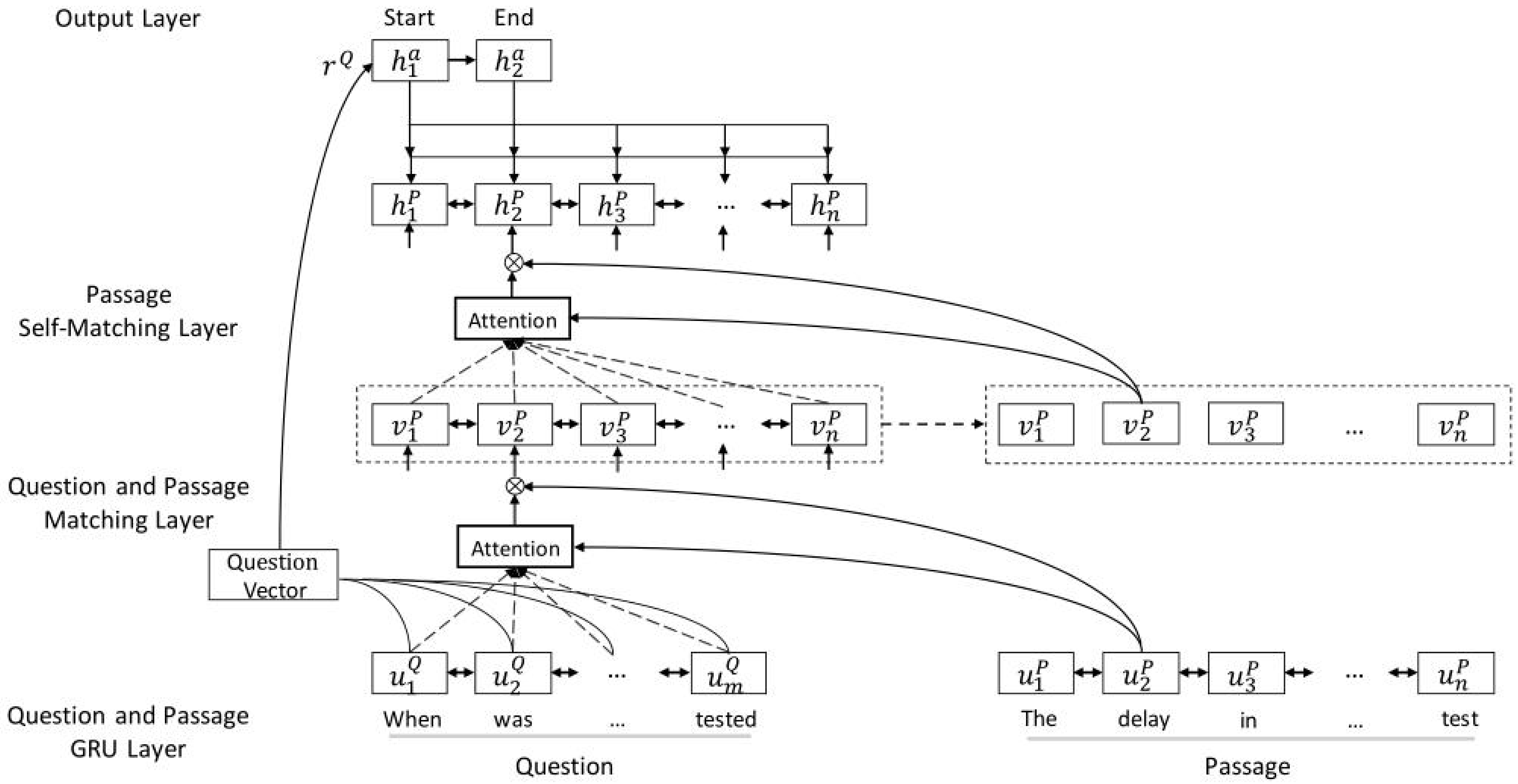
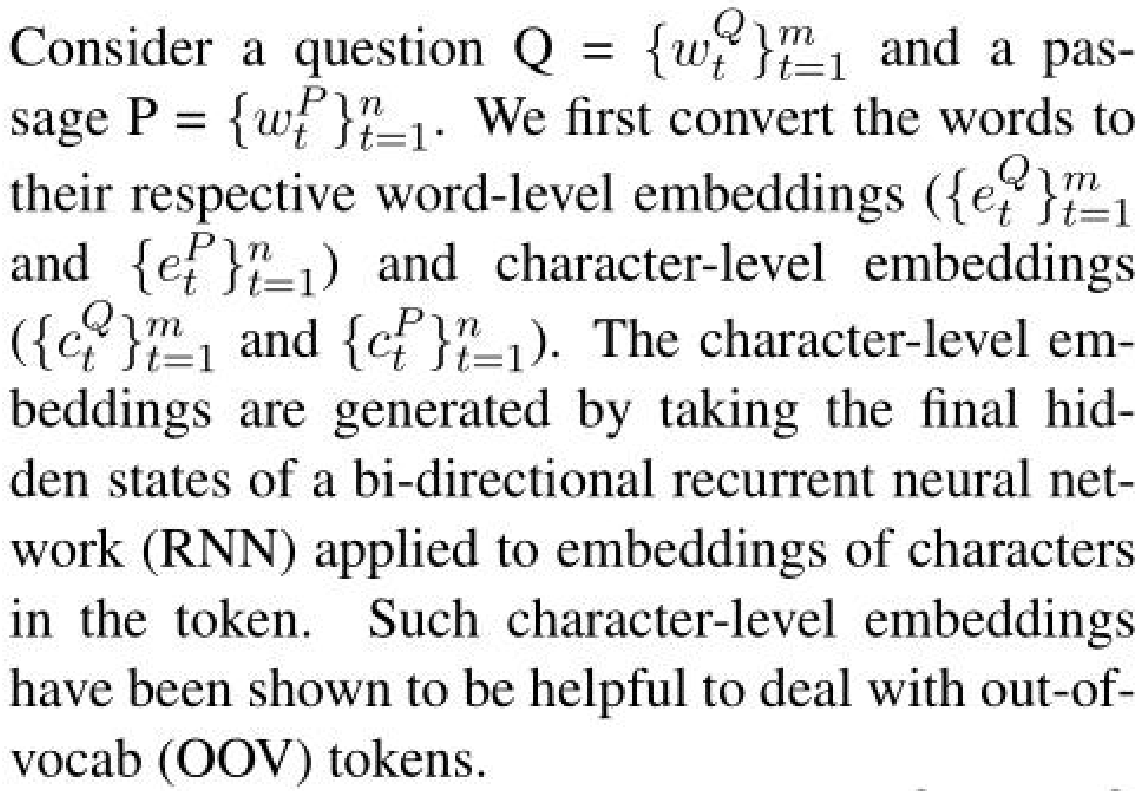


Figure 1: Gated Self-Matching Networks structure overview.

Question and Passage Encoder



Consider a question Q = $\{w_t^Q\}_{t=1}^m$ and a passage P = $\{w_t^P\}_{t=1}^n$. We first convert the words to their respective word-level embeddings ($\{e_t^Q\}_{t=1}^m$ and $\{e_t^P\}_{t=1}^n$) and character-level embeddings ($\{c_t^Q\}_{t=1}^m$ and $\{c_t^P\}_{t=1}^n$). The character-level embeddings are generated by taking the final hidden states of a bi-directional recurrent neural network (RNN) applied to embeddings of characters in the token. Such character-level embeddings have been shown to be helpful to deal with out-of-vocab (OOV) tokens.

分别建立单词级和字符集的embbeding，用Bi-GRU生成问题和篇章的表示

$$u_t^Q = \mathrm{BiRNN}_Q(u_{t-1}^Q, [e_t^Q, c_t^Q]) \quad (1)$$

$$u_t^P = \mathrm{BiRNN}_P(u_{t-1}^P, [e_t^P, c_t^P]) \quad (2)$$

Gated attention-based RNN

$$v_t^P = \mathrm{RNN}(v_{t-1}^P, c_t) \tag{3}$$

where $c_t = att(u^Q, [u_t^P, v_{t-1}^P])$ is an attention-pooling vector of the whole question (u^Q):

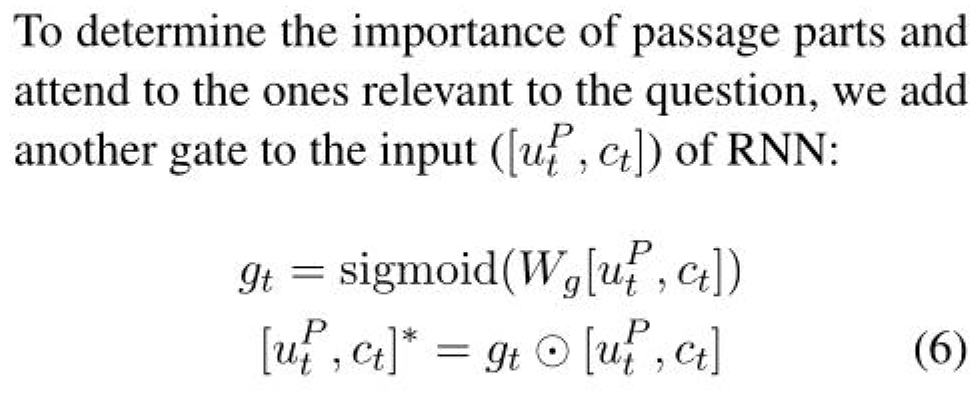
$$s_j^t = \mathrm{v}^{\mathrm{T}}\tanh(W_u^Q u_j^Q + W_u^P u_t^P + W_v^P v_{t-1}^P)$$
$$a_i^t = \exp(s_i^t)/\Sigma_{j=1}^m \exp(s_j^t)$$
$$c_t = \Sigma_{i=1}^m a_i^t u_i^Q \tag{4}$$

通过attention为篇章的不同段落匹配问题相关性，以此划分重点。

将篇章的attention池化作为RNN输入，并新增条件门加以控制。

match-LSTM: $v_t^P = \mathrm{RNN}(v_{t-1}^P, [u_t^P, c_t])$

Gated attention-based RNN



To determine the importance of passage parts and attend to the ones relevant to the question, we add another gate to the input ($[u_t^P, c_t]$) of RNN:

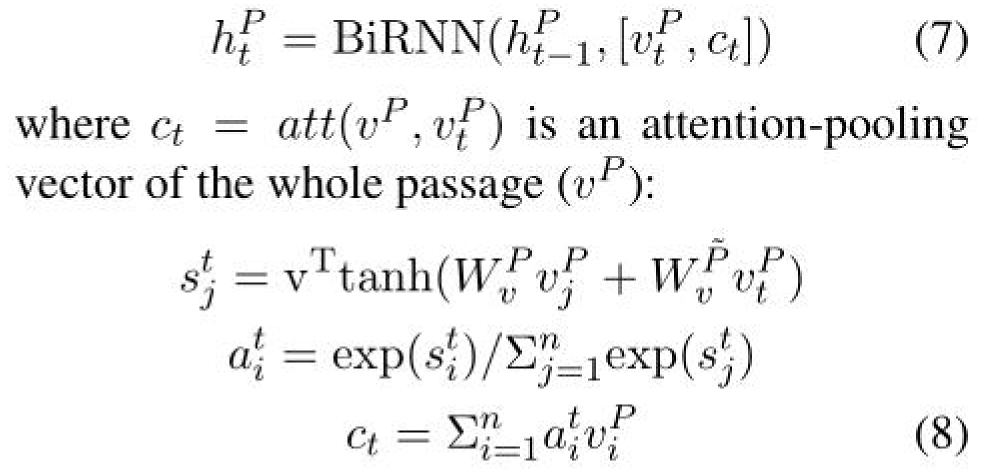
$$g_t = \text{sigmoid}(W_g[u_t^P, c_t])$$
$$[u_t^P, c_t]^* = g_t \odot [u_t^P, c_t] \qquad (6)$$

通过attention为篇章的不同段落匹配问题相关性，以此划分重点。

将篇章的attention池化作为RNN输入，并新增条件门加以控制。

条件门是基于篇章当前处理的词和与问题相关的attention池化向量，这样就可以将问题与当前词联系起来。实验表明文章的某部分与问题相关时条件门才会对输出结果产生影响。

Self-Matching Attention



$$h_t^P = \text{BiRNN}(h_{t-1}^P, [v_t^P, c_t]) \tag{7}$$

where $c_t = att(v^P, v_t^P)$ is an attention-pooling vector of the whole passage (v^P):

$$s_j^t = \mathrm{v}^{\mathrm{T}}\tanh(W_v^P v_j^P + W_v^{\tilde{P}} v_t^P)$$
$$a_i^t = \exp(s_i^t)/\Sigma_{j=1}^n \exp(s_j^t)$$
$$c_t = \Sigma_{i=1}^n a_i^t v_i^P \tag{8}$$

根据当前单词和问题从全文提取信息判断当前词与问题的匹配程度

Output Layer

$$s_j^t = \mathrm{v}^{\mathrm{T}}\tanh(W_h^P h_j^P + W_{\mathrm{h}}^a h_{t-1}^a)$$
$$a_i^t = \exp(s_i^t)/\Sigma_{j=1}^n \exp(s_j^t)$$
$$p^t = \arg\max(a_1^t, \ldots, a_n^t) \qquad (9)$$

Here h_{t-1}^a represents the last hidden state of the answer recurrent network (pointer network). The input of the answer recurrent network is the attention-pooling vector based on current predicted probability a^t:

$$c_t = \Sigma_{i=1}^n a_i^t h_i^P$$
$$h_t^a = \mathrm{RNN}(h_{t-1}^a, c_t) \qquad (10)$$

pointer-network指出答案的起始与结尾位置，用第一层生成的question vector对pointer-network进行初始化

$$s_j = \mathrm{v}^{\mathrm{T}}\tanh(W_u^Q u_j^Q + W_{\mathrm{v}}^Q V_r^Q)$$
$$a_i = \exp(s_i)/\Sigma_{j=1}^m \exp(s_j)$$
$$r^Q = \Sigma_{i=1}^m a_i u_i^Q \qquad (11)$$