Quesion Answering

CS671: Course Project

April 14, 2018

Dataset

SQuAD Dataset

- Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading context.

SQuAD Dataset

- Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading context.
- The training set consists of around 100,000+ question-answer pairs on 500+ wikipedia articles

SQuAD Dataset Example

- **Context :** "The Joan B. Kroc Institute for International Peace Studies at the University of Notre Dame is dedicated to research, education and outreach on the causes of violent conflict and the conditions for sustainable peace. It offers PhD, Master's, and undergraduate degrees in peace studies. It was founded in 1986 through the donations of Joan B. Kroc, the widow of McDonald's owner Ray Kroc. The institute was inspired by the vision of the Rev. Theodore M. Hesburgh CSC, President Emeritus of the University of Notre Dame. The institute has contributed to international policy discussions about peace building practices."

SQuAD Dataset Example

- **Context :** "The Joan B. Kroc Institute for International Peace Studies at the University of Notre Dame is dedicated to research, education and outreach on the causes of violent conflict and the conditions for sustainable peace. It offers PhD, Master's, and undergraduate degrees in peace studies. It was founded in 1986 through the donations of Joan B. Kroc, the widow of McDonald's owner Ray Kroc. The institute was inspired by the vision of the Rev. Theodore M. Hesburgh CSC, President Emeritus of the University of Notre Dame. The institute has contributed to international policy discussions about peace building practices."
- **Question :** "What institute at Notre Dame studies the reasons for violent conflict?"

SQuAD Dataset Example

- **Context :** "The Joan B. Kroc Institute for International Peace Studies at the University of Notre Dame is dedicated to research, education and outreach on the causes of violent conflict and the conditions for sustainable peace. It offers PhD, Master's, and undergraduate degrees in peace studies. It was founded in 1986 through the donations of Joan B. Kroc, the widow of McDonald's owner Ray Kroc. The institute was inspired by the vision of the Rev. Theodore M. Hesburgh CSC, President Emeritus of the University of Notre Dame. The institute has contributed to international policy discussions about peace building practices."
- **Question :** "What institute at Notre Dame studies the reasons for violent conflict?"
- **Answer :** "Joan B. Kroc Institute for International Peace Studies"

SQuAD Dataset Example

- **Context :** "The Joan B. Kroc Institute for International Peace Studies at the University of Notre Dame is dedicated to research, education and outreach on the causes of violent conflict and the conditions for sustainable peace. It offers PhD, Master's, and undergraduate degrees in peace studies. It was founded in 1986 through the donations of Joan B. Kroc, the widow of McDonald's owner Ray Kroc. The institute was inspired by the vision of the Rev. Theodore M. Hesburgh CSC, President Emeritus of the University of Notre Dame. The institute has contributed to international policy discussions about peace building practices."
- **Question :** "What institute at Notre Dame studies the reasons for violent conflict?"
- **Answer :** "Joan B. Kroc Institute for International Peace Studies"
- **Answer Start :** 4

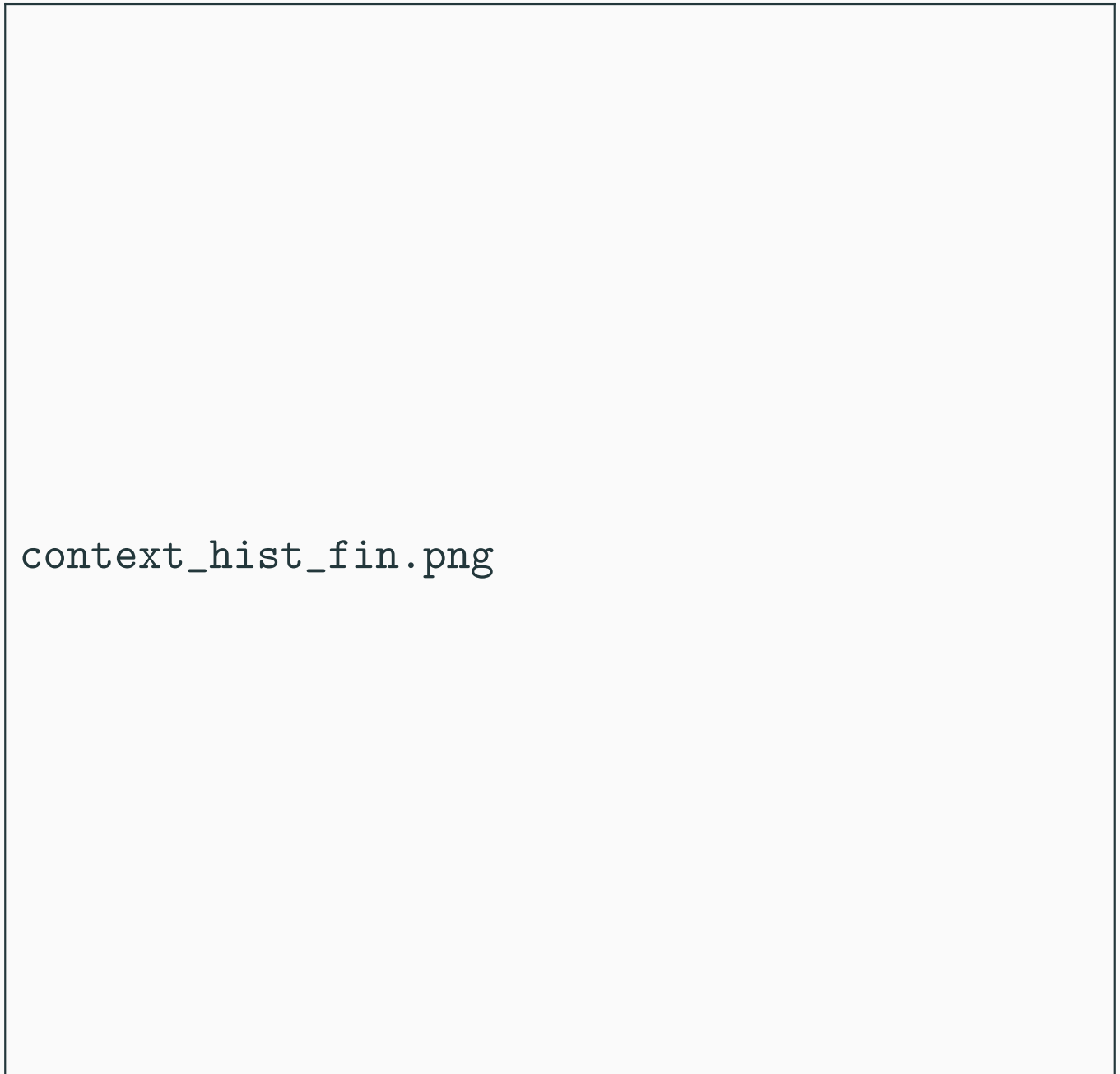
Analysing SQuAD Dataset

- Answer to the SQuAD questions are directly drawn from text, so any model need to predict only correct starting and ending point to answer rather than producing any texts.

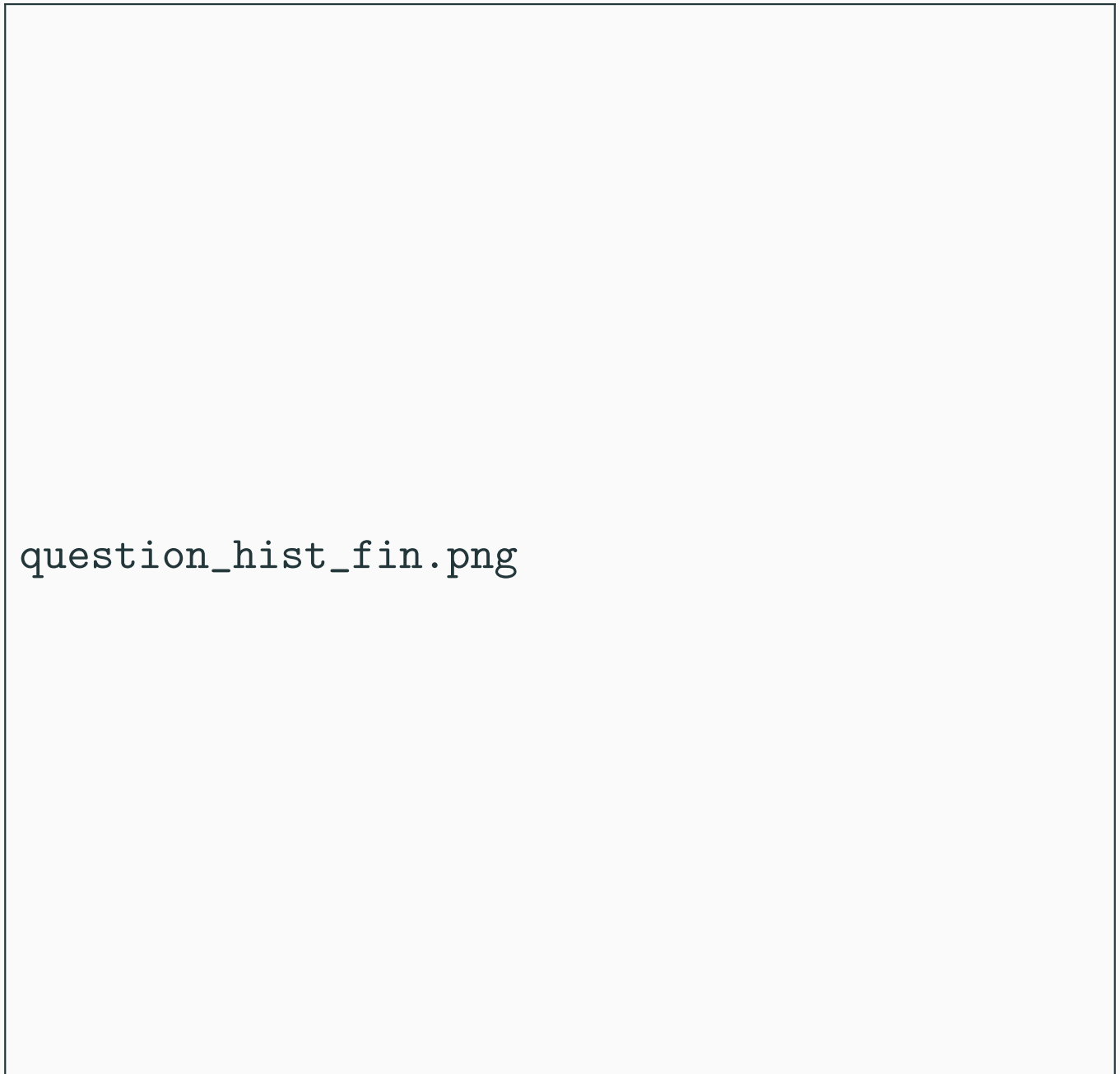
Analysing SQuAD Dataset

- Answer to the SQuAD questions are directly drawn from text, so any model need to predict only correct starting and ending point to answer rather than producing any texts.
- We perform some basic analysis on the dataset in the form of histograms of the context paragraph lengths, question lengths, and answer lengths in the training set.

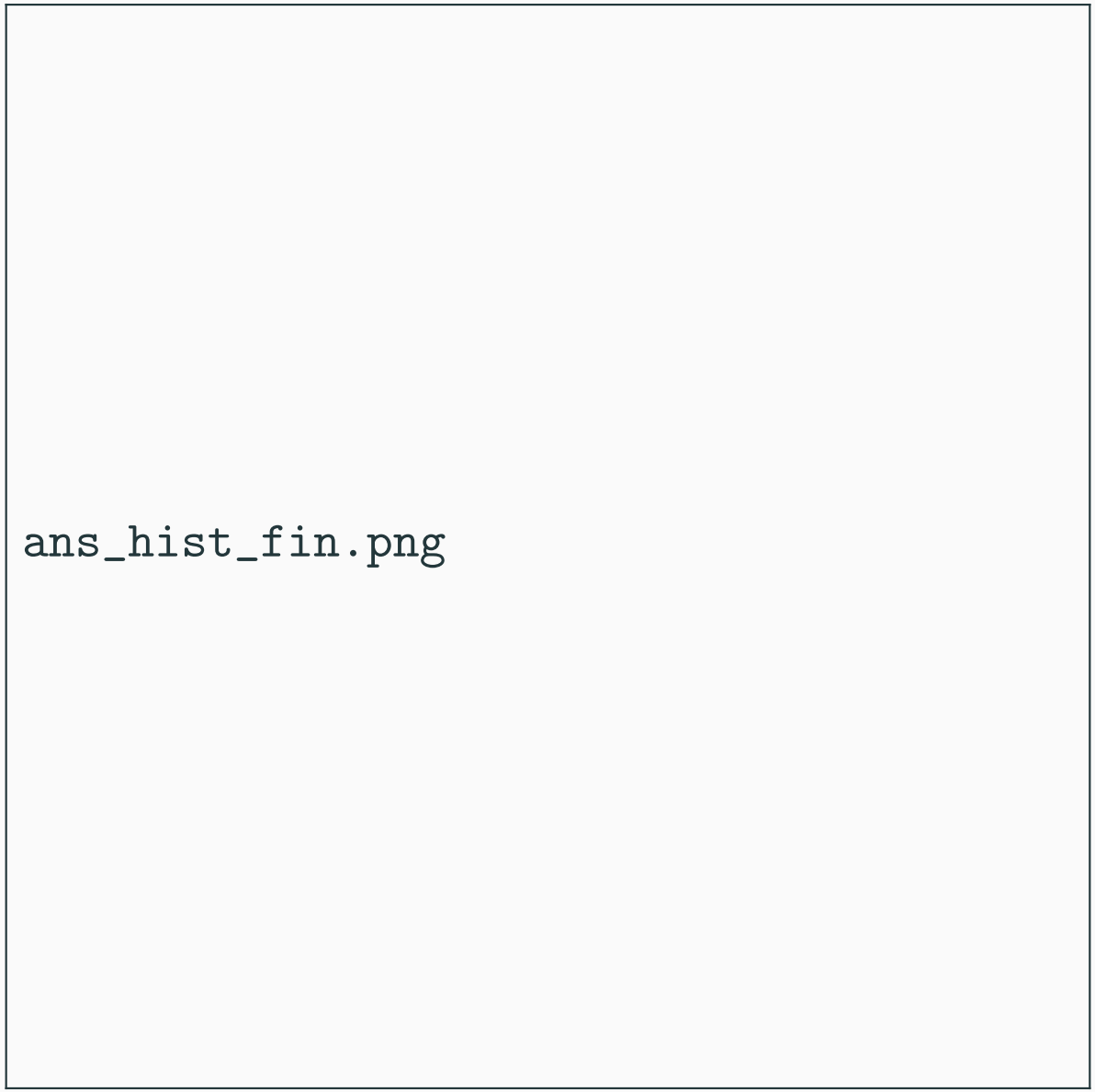
Analysing SQuAD Dataset - Graph(1/4)



Analysing SQuAD Dataset - Graph(2/4)

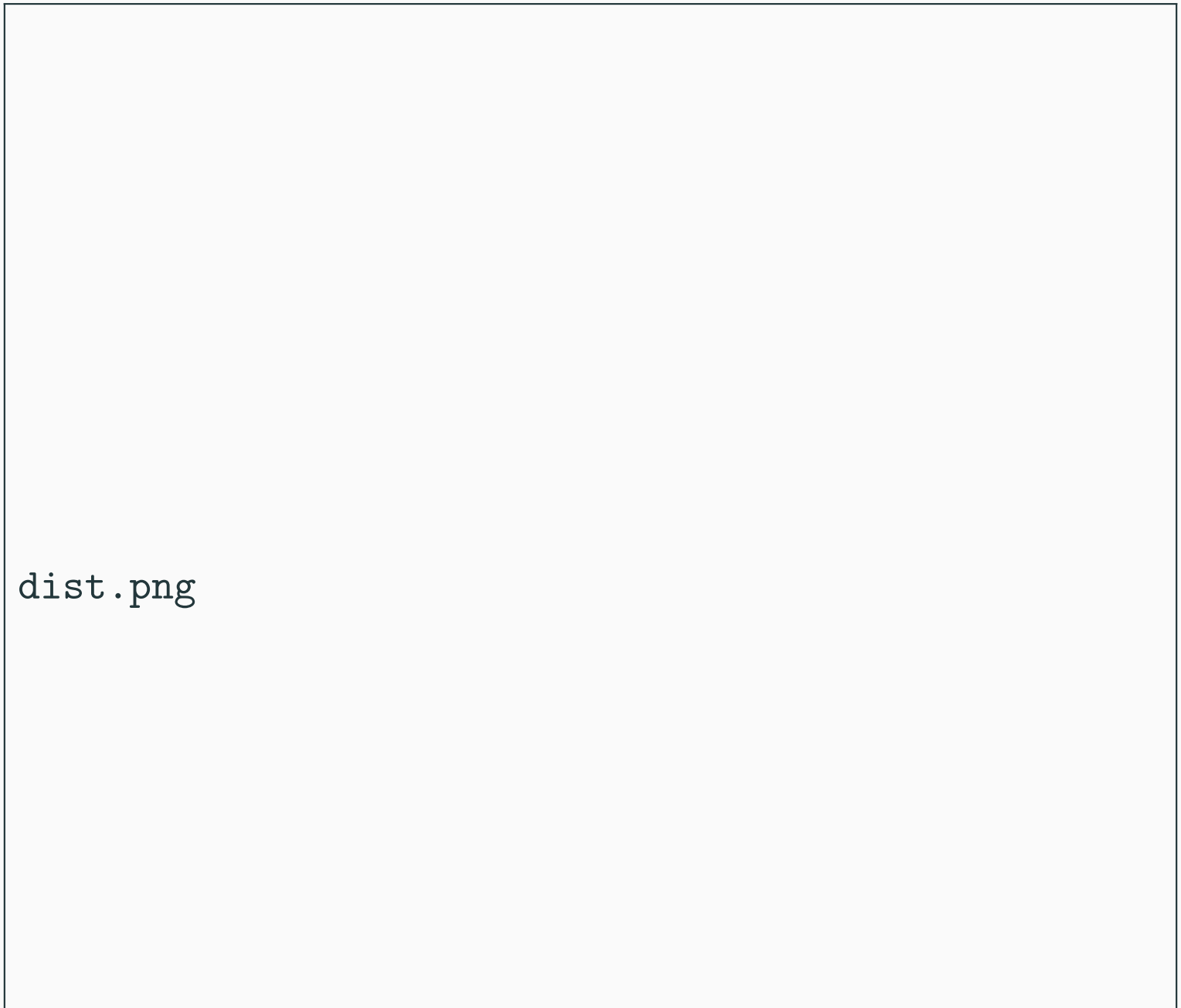


Analysing SQuAD Dataset - Graph(3/4)



- Most answers are shorter than 10 words

Analysing SQuAD Dataset - Graph(4/4)



Analysing SQuAD Dataset

- Based on the above analysis, we set an optimal choice for hyper parameters such as maxContextLength, maxQuestionLength, maxAnswerLength and use these to clip / pad the contexts, questions and answers

Models

We explore two models for the machine comprehension task,

Models

We explore two models for the machine comprehension task,

- Match-LSTM and Answer Pointer

Models

We explore two models for the machine comprehension task,

- Match-LSTM and Answer Pointer
- Bidaf

Match-LSTM and Answer Pointer

LSTM Preprocessing Layer

- The purpose of the LSTM Preprocessing Layer is to incorporate contextual information into the representation of each token in the context and the question.

LSTM Preprocessing Layer

- The purpose of the LSTM Preprocessing Layer is to incorporate contextual information into the representation of each token in the context and the question.
-

$$\boldsymbol{H}^p = \overrightarrow{\text{LSTM}}(\boldsymbol{P}), \qquad \boldsymbol{H}^q = \overrightarrow{\text{LSTM}}(\boldsymbol{Q})$$

LSTM Preprocessing Layer

- The purpose of the LSTM Preprocessing Layer is to incorporate contextual information into the representation of each token in the context and the question.
-

$$\boldsymbol{H}^p = \overrightarrow{\text{LSTM}}(\boldsymbol{P}), \qquad \boldsymbol{H}^q = \overrightarrow{\text{LSTM}}(\boldsymbol{Q})$$

- The matrices $\boldsymbol{H}^p$ and $\boldsymbol{H}^q$ represent the hidden representations of the context and the question matrices, respectively.

Match-LSTM Layer

- Question is treated as a premise and answer as a hypothesis

Match-LSTM Layer

- Question is treated as a premise and answer as a hypothesis
- The match-LSTM sequentially goes through the context.

Match-LSTM Layer

- Question is treated as a premise and answer as a hypothesis
- The match-LSTM sequentially goes through the context.
- At position i the word-by-word attention is calculated as follows:

Match-LSTM Layer

- Question is treated as a premise and answer as a hypothesis
- The match-LSTM sequentially goes through the context.
- At position i the word-by-word attention is calculated as follows:
-

$$\boldsymbol{G_i} = tanh(\mathbf{W}^q\mathbf{H}^q + (\mathbf{W}^p\mathbf{h}_i^p + \mathbf{W}^r\mathbf{h}_{i-1}^r + \mathbf{b}^p) \otimes e_{\mathbf{Q}})$$

$$\overrightarrow{\alpha_i} = softmax(\mathbf{w}^T\overrightarrow{\mathbf{G}_i} + b \otimes \mathbf{e}_Q)$$

Match-LSTM Layer

- Question is treated as a premise and answer as a hypothesis
- The match-LSTM sequentially goes through the context.
- At position i the word-by-word attention is calculated as follows:
-

$$\boldsymbol{G_i} = tanh(\mathbf{W}^q\mathbf{H}^q + (\mathbf{W}^p\mathbf{h}_i^p + \mathbf{W}^r\mathbf{h}_{i-1}^r + \mathbf{b}^p) \otimes e_{\mathbf{Q}})$$
$$\overrightarrow{\alpha_i} = softmax(\mathbf{w}^T\overrightarrow{\mathbf{G}_i} + b \otimes \mathbf{e}_Q)$$

- where $\mathbf{W}^q$, $\mathbf{W}^p$, $\mathbf{W}^r$, $\in \mathbb{R}^{l \times l}$, $\mathbf{b}, \mathbf{w} \in \mathbb{R}^l$ and $\mathbf{b} \in \mathbb{R}$ are the parameters to be learned.

Match-LSTM Layer

- Question is treated as a premise and answer as a hypothesis
- The match-LSTM sequentially goes through the context.
- At position i the word-by-word attention is calculated as follows:
-

$$\boldsymbol{G_i} = tanh(\mathbf{W}^q\mathbf{H}^q + (\mathbf{W}^p\mathbf{h}_i^p + \mathbf{W}^r\mathbf{h}_{i-1}^r + \mathbf{b}^p) \otimes e_{\mathbf{Q}})$$
$$\overrightarrow{\alpha_i} = softmax(\mathbf{w}^T\overrightarrow{\mathbf{G}_i} + b \otimes \mathbf{e}_Q)$$

- where $\mathbf{W}^q$, $\mathbf{W}^p$, $\mathbf{W}^r$, $\in \mathbb{R}^{l \times l}$, $\mathbf{b}, \mathbf{w} \in \mathbb{R}^l$ and $\mathbf{b} \in \mathbb{R}$ are the parameters to be learned.
- $\overrightarrow{\mathbf{h}_{i-1}^r} \in \mathbb{R}^l$ is the hidden vector at position i-1

Match-LSTM Layer

- Take the weighted version of the question and combine it with the current token of the context to form a vector $\overrightarrow{\mathbf{z}_i}$:

$$\overrightarrow{z_i} = \begin{bmatrix} \mathbf{h}_i^p \\ \mathbf{H}^q \overrightarrow{\alpha_i}^T \end{bmatrix}$$

Match-LSTM Layer

- Take the weighted version of the question and combine it with the current token of the context to form a vector $\overrightarrow{\mathbf{z}_i}$:

$$\overrightarrow{z_i} = \begin{bmatrix} \mathbf{h}_i^p \\ \mathbf{H}^q \overrightarrow{\alpha_i}^T \end{bmatrix}$$

- This vector $\overrightarrow{\mathbf{z}_i}$ is fed into a standard one-directional LSTM to form our so-called match-LSTM:

$$\overrightarrow{\mathbf{h}_i^r} = \overrightarrow{\text{LSTM}}(\overrightarrow{\mathbf{z}_i}, \overrightarrow{\mathbf{h}_{i-1}^r}),$$

where $\overrightarrow{\mathbf{h}_i^r} \in \mathbb{R}^l$

Match-LSTM Layer

- Similar Match-LSTM layer is build in the reverse direction.

Match-LSTM Layer

- Similar Match-LSTM layer is build in the reverse direction.
- Obtain a representation that encodes the contexts from both directions for each token in the context

Match-LSTM Layer

- Similar Match-LSTM layer is build in the reverse direction.
- Obtain a representation that encodes the contexts from both directions for each token in the context
-

$$\overleftarrow{\mathbf{G}_i} = tanh(\mathbf{W}^q\mathbf{H}^q + (\mathbf{W}^p\mathbf{h}_i^p + \mathbf{W}^r \overleftarrow{h}^r_{i+1}\mathbf{b}^p) \otimes \mathbf{e}_Q),$$
$$\overleftarrow{\alpha_i} = softmax(\mathbf{w}^T\overleftarrow{\mathbf{G}_i} + b \otimes \mathbf{e}_Q),$$

where the parameters are $(\mathbf{W}^q, \mathbf{W}^p, \mathbf{W}^r, \mathbf{b}^p, \mathbf{w}\ and\ b)$

Match-LSTM Layer

- Similar Match-LSTM layer is build in the reverse direction.
- Obtain a representation that encodes the contexts from both directions for each token in the context
-

$$\overleftarrow{\mathbf{G}_i} = tanh(\mathbf{W}^q\mathbf{H}^q + (\mathbf{W}^p\mathbf{h}_i^p + \mathbf{W}^r \overleftarrow{h}_{i+1}^r \mathbf{b}^p) \otimes \mathbf{e}_Q),$$
$$\overleftarrow{\alpha_i} = softmax(\mathbf{w}^T \overleftarrow{\mathbf{G}_i} + b \otimes \mathbf{e}_Q),$$

  where the parameters are $(\mathbf{W}^q, \mathbf{W}^p, \mathbf{W}^r, \mathbf{b}^p, \mathbf{w}\ and\ b)$
- $\mathbf{z}_i$ and $\mathbf{h}_i$ are defined in similar manner.

Match-LSTM Layer

- $\overrightarrow{\mathbf{H}^r} \in \mathbb{R}^{l \times P}$ represents the hidden states $[\overrightarrow{\mathbf{h}_1^r}, \ldots, \overrightarrow{\mathbf{h}_P^r}]$

Match-LSTM Layer

- $\overrightarrow{\mathbf{H}^r} \in \mathbb{R}^{l \times P}$ represents the hidden states $[\overrightarrow{\mathbf{h}_1^r}, \ldots, \overrightarrow{\mathbf{h}_P^r}]$
- Similarly $\overleftarrow{\mathbf{H}^r} \in \mathbb{R}^{l \times P}$ represents $[\overleftarrow{\mathbf{h}_1^r}, \ldots, \overleftarrow{\mathbf{h}_P^r}]$

Match-LSTM Layer

- $\overrightarrow{\mathbf{H}^r} \in \mathbb{R}^{l \times P}$ represents the hidden states $[\overrightarrow{\mathbf{h}_1^r}, \ldots, \overrightarrow{\mathbf{h}_P^r}]$
- Similarly $\overleftarrow{\mathbf{H}^r} \in \mathbb{R}^{l \times P}$ represents $[\overleftarrow{\mathbf{h}_1^r}, \ldots, \overleftarrow{\mathbf{h}_P^r}]$
- $\mathbf{H}^r \in \mathbb{R}^{2l \times P}$ as

$$\mathbf{H}^r = \begin{bmatrix} \overrightarrow{\mathbf{H}^r} \\ \overleftarrow{H^r} \end{bmatrix}$$

Answer Pointer Layerd

- The model takes $\mathbf{H}^r$ as input and tries to predict the answer.

Answer Pointer Layerd

- The model takes $\mathbf{H}^r$ as input and tries to predict the answer.
- **Boundary Model**: Concerned with the starting and ending tokens of the answer.

Answer Pointer Layerd

- The model takes $\mathbf{H}^r$ as input and tries to predict the answer.
- **Boundary Model**: Concerned with the starting and ending tokens of the answer.
- The probability of answer is modelled as

$$\mathbb{P}[\mathbf{a}|\mathbf{H}^r] = \mathbb{P}[a_s|\mathbf{H}^r]\mathbb{P}[a_e|a_s, \mathbf{H}^r]$$

Answer Pointer Layerd

- The model takes $\mathbf{H}^r$ as input and tries to predict the answer.
- **Boundary Model**: Concerned with the starting and ending tokens of the answer.
- The probability of answer is modelled as

$$\mathbb{P}[\mathbf{a}|\mathbf{H}^r] = \mathbb{P}[a_s|\mathbf{H}^r]\mathbb{P}[a_e|a_s, \mathbf{H}^r]$$

- This is used to compute a **attention vector** $\beta_k \in \mathbb{R}^{P+1}$, where $\beta_{j,k}$ is the probability of selecting the $j$context token as the $k$token in the answer.

$$\mathbb{P}[a_k = j|a_1, a_2 \dots a_{k-1}, \mathbf{H}^r] = \beta_{j,k}$$

Answer Pointer Layerd

- The model takes $\mathbf{H}^r$ as input and tries to predict the answer.
- **Boundary Model**: Concerned with the starting and ending tokens of the answer.
- The probability of answer is modelled as

$$\mathbb{P}[\mathbf{a}|\mathbf{H}^r] = \mathbb{P}[a_s|\mathbf{H}^r]\mathbb{P}[a_e|a_s, \mathbf{H}^r]$$

- This is used to compute a **attention vector** $\beta_k \in \mathbb{R}^{P+1}$, where $\beta_{j,k}$ is the probability of selecting the $j$context token as the $k$token in the answer.

$$\mathbb{P}[a_k = j|a_1, a_2 \dots a_{k-1}, \mathbf{H}^r] = \beta_{j,k}$$

- Objective is to maximize $\mathbb{P}[\mathbf{a}|\mathbf{H}^r]$, and the answer corresponding to the maximum answer is reported as the answer to the query

Answer Pointer Layer

- The answer pointer does not differentiate the question types, and therefore all questions - what? where? why? how? etc. will be answered using the same attention mechanism

Answer Pointer Layer

- The answer pointer does not differentiate the question types, and therefore all questions - what? where? why? how? etc. will be answered using the same attention mechanism
- In order to differentiate, we setup different Answer Pointer layers for different question types

Answer Pointer Layer

- The answer pointer does not differentiate the question types, and therefore all questions - what? where? why? how? etc. will be answered using the same attention mechanism
- In order to differentiate, we setup different Answer Pointer layers for different question types
- This allows us to model each question type differently

Loss Function

- The loss function for the Answer Pointer Layer is the Sigmoidal Cross-Entropy Loss

Loss Function

- The loss function for the Answer Pointer Layer is the Sigmoidal Cross-Entropy Loss
- For each answer token, the cross entropy is computed and added to the total loss

Loss Function

- The loss function for the Answer Pointer Layer is the Sigmoidal Cross-Entropy Loss
- For each answer token, the cross entropy is computed and added to the total loss
- For the boundary model, there are only two answer tokens, the start and the end

Loss Function

- The loss function for the Answer Pointer Layer is the Sigmoidal Cross-Entropy Loss
- For each answer token, the cross entropy is computed and added to the total loss
- For the boundary model, there are only two answer tokens, the start and the end
- The loss is given as

$$-\log p(a_s \mid \boldsymbol{P}_n, \boldsymbol{Q}_n) - \log p(a_e \mid \boldsymbol{P}_n, \boldsymbol{Q}_n)$$

Weighted Loss

Attention

Attention

- Model learns what to attend based on the input query.
- Mapping a query and a set of key-value pairs to an output.
- Output is a weighted sum of the values, weights are computed by a compatibility function of the query with the corresponding key.

Scaled Dot-Product Attention

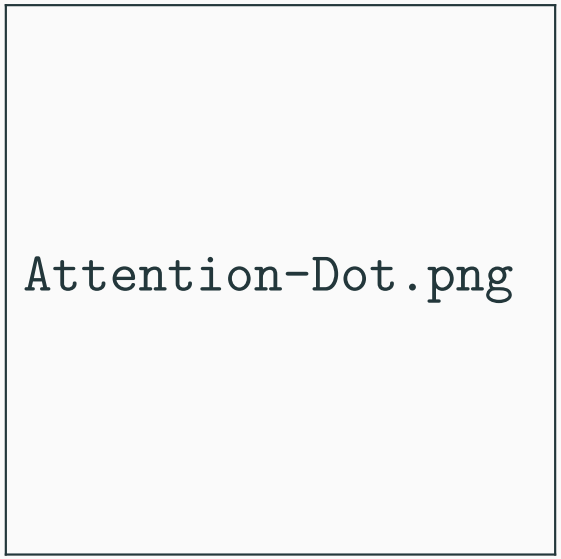


Figure 1: Scaled Dot-Product Attention [Source:][]–Attention is all you need

- We compute the dot products of the query(d_k) with all keys(d_k).
- Attention function is calculated on a set of queries simultaneously, packed together into a matrix Q.
- The keys and values are packed together into matrices K and V.

Scaled Dot-Product Attention

$$\text{Attention}(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

- For large values of d_k , the dot products grow large in magnitude. pushing the softmax function into regions where it has extremely small gradients. To counteract this effect, dot product is scaled by $\frac{1}{\sqrt{d}}$.
- Dot-Product Attention is much faster and more space-efficient in practice as compared to additive attention.

Multi-Head Attention

Figure 2: Multi-Head Attention [Source:][]–Attention is all you need

- Linearly projecting the queries, keys and values h times with different, learned linear projections to d_k , d_k and d_v dimensions, respectively.
- Now on each of these projected version perform the attention function in parallel.

Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Projection are parameter matrices
$W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_k}, W_i^Q \in \mathbb{R}^{d_{model} \times d_v}, W^O \in R^{hd^v \times d_m od}$
Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality.

Pointer-Net

Pointer Net

- Used to predict the start(s) and end(e)
- Given the context representation $\{h_t^P\}_{t=1}^n$, the attention mechanism is utilized as a pointer to select the start position (p^1) and end position (p^2) from the context.
-

Dynamic Programming

Dynamic Programming

- After getting the logits for start and end positions, Dynamic Programming strategies are used.
- We select (s, e) where $s \leq e$ with the maximum value of $p_s^1 p_e^2$, which can be calculated in linear time using dynamic programming.
- Legal answer is selected with highest join probability of start and end. The F1 score was improved by 2 points.
- We then used a sentence-level DP to find the maximum $p_s^1 p_e^2$ where s and e are in same sentence. This improvement boosted the F1 score by another 0.5 point.