The goal of this project is to improve the architecture of Machine Comprehension models by expanding attention mechanisms with a deeper recurrent inference chain. I have two major contributions:

- Extend the Ruminating Reader Block ([1]) to a recurrent chain and combine the model with the Encoder Block of QANet ([2]).
- Propose and entirely new convolutional Ruminating Block, with dual attention layers.

**Model 1**

The first model applies the original Ruminating Block with some modifications. Blocks are then stacked together in a sequential manner. It consists of three main units:

- **Attention Summarization Layer** The attention summarization layer deploys an LSTM model on the output of the Attention Layer to return hidden states $s_i = [s_q, s_c, s_{r,e}] \in \mathbb{R}^{3d}$ and summary state $s = [s_{m,c}, s_{m,q}] \in \mathbb{R}^{2d}$. The superscript $t$ indicates that this is stack-$t$ of the ruminating chain.

- **Context Summarization Layer** This layer is a gating mechanism, which aims to compute query-aware context vectors:

  $a_i = \tanh(W_i^s s_i + W_i^c c_i + b_i)$
  
  $f_i = \sigma(W_i^s s_i + W_i^c c_i + b_i)$
  
  $c_i = f_i \odot c_i + (1 - f_i) \odot a_i$

  Here $c_i$ are the context word encodings.

- **Query Summarization Layer** Similar to the Context Summarization Layer, but using the summary state vector.

  $a_i = \tanh(W_i^q s_i + W_i^q q_i + b_i)$
  
  $f_i = \sigma(W_i^q s_i + W_i^q q_i + b_i)$
  
  $q_i = f_i \odot q_i + (1 - f_i) \odot a_i$

  Here $q_i$ are the query word encodings.

**Model 2**

Model 2 is an original structure for the Ruminating Block. We add an additional Attention Module, that swaps the Query-to-Context and Context-to-Query steps in the original to compute query-attention vectors:

- **Attention Layer** We add an additional Attention Module, that swaps the Query-to-Context and Context-to-Query steps in the original to compute query-attention vectors.

  $s_i = \sigma(W_i^Q q_i + W_i^C c_i + b_i)$
  
  $s = \sigma(W_i^Q q + W_i^C c + b_i)$

- **Summarization Layer** The summary LSTM is replaced by an Encoder Block, which operates on the output of both attention layers and returns context-summaries $s_i$ and query-summaries $s_q$.

- **Ruminating Layers** Both Ruminating Layers operate like the Context Summarization Layer from Model 1, using $s_i$ and $s_q$, respectively.

- **Modeling Layer** The modeling layer takes as input the context-attention vectors from the last iteration of the chain.

---

**Results**

Summary of model results:

<table>
<thead>
<tr>
<th>Model</th>
<th>Development Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>63.89</td>
<td>67.52</td>
</tr>
<tr>
<td>QANet + Ruminating Block v1</td>
<td>63.59</td>
<td>67.02</td>
</tr>
<tr>
<td>Model 1 + Multi-head Attention</td>
<td>63.12</td>
<td>66.56</td>
</tr>
<tr>
<td>QANet + Ruminating Chain v2</td>
<td>57.33</td>
<td>60.92</td>
</tr>
</tbody>
</table>

**Major observations:**

- Adding the Ruminating Chain with the architecture from Model 1 to the Baseline model increased performance on the dev set significantly.
- Adding multiple attention heads (or quadratic similarity) does not improve performance.
- Our proposed Ruminating Chain adds 1 point in EM and F1 scores over the original version.

**Conclusion and next steps**

This project proposed an extension of the existing Ruminating Block into a chain and combined it with the general set-up of QANet, which improved performance over each model. We also proposed an entirely new original model for the Ruminating Block and showed that it outperformed all previous models. There are multiple avenues for further development:

- Explore the inference capacity of the ruminating chain: varying depth and attention complexity.
- Train the alternative form of Model 2.
- Perform more extensive hyper-parameter search (i.e. learning rate annealing).

**References**
