Text Summarization

Samuel Sunarjo
What if…?

- Gmail is trying to implement a new feature to summarize all emails you received recently.
- Generating headlines for news and articles.
- Creating meeting minutes or summaries.
- Outlines, such as notes for students.

Text Summarization!!
What is Text Summarization?

- Shorten long texts/documents.
- Summarize by the main points.
- TL;DR.
- A common problem in ML/NLP.

Jenson Button was denied his 100th race for McLaren after an ERS prevented him from making it to the start-line. It capped a miserable weekend for the Briton; his time in Bahrain plagued by reliability issues. Button spent much of the race on Twitter delivering his verdict as the action unfolded. 'Kimi is the man to watch,' and 'loving the sparks', were among his pearls of wisdom, but the tweet which courted the most attention was a rather mischievous one: 'Ooh is Lewis backing his team mate into Vettel?'

he quizzed after Rosberg accused Hamilton of pulling off such a manoeuvre in China. Jenson Button waves to the crowd ahead of the Bahrain Grand Prix which he failed to start Perhaps a career in the media beckons... Lewis Hamilton has out-qualified and finished ahead of Nico Rosberg at every race this season. Indeed Rosberg has now beaten his Mercedes team-mate only once in the 11 races since the pair infamously collided in Belgium last year. Hamilton secured the 36th win of his career in Bahrain and his 21st from pole position. Only Michael Schumacher (40), Ayrton Senna (29) and Sebastian Vettel (27) have more. (...)

Button denied 100th race start for McLaren after ERS failure. Button then spent much of the Bahrain Grand Prix on Twitter delivering his verdict on the action as it unfolded. Lewis Hamilton has out-qualified and finished ahead of Mercedes team-mate Nico Rosberg at every race this season. Bernie Ecclestone confirms F1 will make its bow in Azerbaijan next season.
Why Text Summarization?

- Big Data. A lot of text data.
- Accelerates the process of information retrieval.
- Discover relevant information and consume faster.
- Increases the amount of information that can fit in an area.
- Expensive and time consuming if done without machines.
Types of Summarization

Extraction-based

- Pulling keyphrases from the source and combining them to make a summary.
- Without changes to the texts.
- Example
  - **Source text:**
    Joseph and Mary rode on a donkey to **attend** the annual **event** in Jerusalem. In the city, Mary gave **birth** to a child named **Jesus**.
  - **Extractive summary:**
    Joseph and Mary attend event Jerusalem. Mary birth Jesus.
  - Can be grammatically strange.
Types of Summarization

Abstraction-based

- Paraphrasing and shortening parts of the source document.
- Create new phrases that relay the most useful information from the source.
- Could overcome grammar irregularities of the extractive approach.
- Generally more difficult to develop.
- Example
  - Source text:
    Joseph and Mary rode on a donkey to attend the annual event in Jerusalem. In the city, Mary gave birth to a child named Jesus.
  - Extractive summary:
    Joseph and Mary came to Jerusalem where Jesus was born.
Extraction-based Summarization

- **Lex Rank**
  - A graphical based text summarizer.

- **Luhn**
  - One of the earliest algorithm, named after famous IBM researcher.
  - Scores sentences based on frequency of the most important words.

- **Latent semantic analysis (LSA)**
  - An unsupervised method of summarization it combines term frequency techniques with singular value decomposition to summarize texts.

- **TextRank**
  - Weighted-graph-based summarization with keyword extractions from source.

*Above are all non-ML or unsupervised methods.*
Evaluation Metric

What would be a good metric for evaluating text summarization performance?
Evaluation Metric

ROUGE-N

● N-gram measure between a model prediction and the reference summary

● A recall value which measures how many N-grams from the reference summary appear in the model summary
  ○ $|\text{reference} \cap \text{model}| / |\text{reference}|$

● N controls length of the N-gram word phrase that needs to be matched completely.

● For example, “Text Summarization” and “Summarization Text”
  ○ ROUGE-1 would be 1 and ROUGE-2 would be 0
Evaluation Metric

BLEU

- Modified form of precision
- $|\text{reference} \cap \text{model}| / |\text{model}|$
- A weighted average to account for variable length phrases (N-grams).
  - To avoid the problem of repeated/over-generated relevant information.

BLEU with modified N-gram precision

- Clip the total count of each model word by maximum number of times a word occurs in any single reference.
Evaluation Metric

METEOR

- Sentence-level similarity scores. Search all possible alignments.
- Exact
  - Match words if they are identical.
- Stem
  - Match stem words using a language appropriate Snowball Stemmer (Porter, 2001).
- Synonym
  - Match words that share membership in any synonym set based on WordNet database.
- Paraphrase
  - Match phrases if they are listed as paraphrases in a defined paraphrase table.
Get To The Point: Summarization with Pointer-Generator Networks

- **Datasets - CNN/Daily Mail**
  - Contains online news articles paired with multi-sentence summaries.
  - Non-anonymized version. Requires no pre-processing of masking named entity.

- **Shortcomings and challenges of previous approaches (Neural sequence-to-sequence models)**
  - Repetitive.
  - Inaccurate.

- **Main ingredients.**
  - Sequence-to-sequence attentional model
  - Pointer-generator network
  - Coverage mechanism
Sequence-to-sequence attentional model
Sequence-to-sequence attentional model

Attention Distribution \( a^t \):
\[
e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{\text{attn}}) \\
a^t = \text{softmax}(e^t)
\]
\( v, W_h, W_s \text{ and } b_{\text{attn}} \text{ are learnable} \)

Context Vector \( h_t^* \):
\[
h_t^* = \sum_i a_i^t h_i
\]

Vocab Distribution:
\[
P_{\text{vocab}} = \text{softmax}(V'(V[s_t, h_t^*] + b) + b') \\
P(w) = P_{\text{vocab}}(w)
\]
\( V, V', b \text{ and } b' \text{ are learnable} \)

Loss:
\[
\text{loss}_t = - \log P(w_t^*)
\]
\( \text{target word } w_t^* \)
Pointer-generator network
Pointer-generator network

- **Hybrid**
  - Allows both copying words via pointing and generating words from a fixed vocabulary.

- **Generation probability** $p_{\text{gen}}$
  - Probability of generating words from vocabulary vs. copying words from the source text.
  - $p_{\text{gen}} = \sigma(w_h^T h_t^* + w_s^T s_t + w_x^T x_t + b_{\text{ptr}})$
  - Learnable: $w_h^*, w_s, w_x$ and scalar $b_{\text{ptr}}$

- **Extended vocabulary**
  - Union of the pre-set vocabulary and all words appearing in the source document.
  - $P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i:w_i=w} a_i$
    - if $w$ is an out-of-vocabulary (OOV) word, then $P_{\text{vocab}}(w)$ is zero.
    - If $w$ does not appear in the source document, then $\sum_{i:w_i=w} a_i$ is zero.
    - Replace $P(w)$ in previous loss function.
Coverage mechanism

- To address the repetition issue.
- Coverage vector $c^t$
  - $c^t = \sum_{t'=0}^{t-1} d'$ (sum of attention)
- Additional input to attention
  - $e_i^t = v^T \tanh(W_h h_i + W_s s_i + w_c c_i^t + b_{\text{attn}})$
  - Inform attention network a reminder of its previous decisions.
- Coverage loss
  - $\text{covloss}_t = \sum_i \min(d_i^t, c_i^t)$
- Final loss
  - $\text{loss}_t = -\log P(w_t^*) + \lambda \sum_i \min(d_i^t, c_i^t)$
# Results

<table>
<thead>
<tr>
<th></th>
<th>ROUGE 1</th>
<th>ROUGE 2</th>
<th>ROUGE L</th>
<th>METEOR exact match</th>
<th>METEOR + stem/syn/para</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstractive model</td>
<td>35.46</td>
<td>13.30</td>
<td>32.65</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(Nallapati et al., 2016)*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>seq-to-seq + attn baseline (150k vocab)</td>
<td>30.49</td>
<td>11.17</td>
<td>28.08</td>
<td>11.65</td>
<td>12.86</td>
</tr>
<tr>
<td>seq-to-seq + attn baseline (50k vocab)</td>
<td>31.33</td>
<td>11.81</td>
<td>28.83</td>
<td>12.03</td>
<td>13.20</td>
</tr>
<tr>
<td>pointer-generator</td>
<td>36.44</td>
<td>15.66</td>
<td>33.42</td>
<td>15.35</td>
<td>16.65</td>
</tr>
<tr>
<td>pointer-generator + coverage</td>
<td><strong>39.53</strong></td>
<td><strong>17.28</strong></td>
<td><strong>36.38</strong></td>
<td>17.32</td>
<td>18.72</td>
</tr>
<tr>
<td>lead-3 baseline</td>
<td>40.34</td>
<td>17.70</td>
<td>36.57</td>
<td>20.48</td>
<td>22.21</td>
</tr>
<tr>
<td>(ours)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lead-3 baseline</td>
<td>39.2</td>
<td>15.7</td>
<td>35.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(Nallapati et al., 2017)*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>extractive model</td>
<td>39.6</td>
<td>16.2</td>
<td>35.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(Nallapati et al., 2017)*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* were trained and evaluated on the anonymized dataset
lead-3 baseline uses the first three sentences of the article as a summary.
Results - Coverage

Coverage eliminates undesirable repetition.
The baseline mode produces more novel n-grams but mostly incorrect.
Results - Example

Baseline
- Cannot generate OOV words (name).
- Thwart -> destabilize
- Repetition and eventually non-sense.
Conclusion

- Ability to produce OOV words is one of the primary advantages of the hybrid pointer-generator model.
- Drastically reduced repetition through coverage.
- Significantly outperformed the abstractive state-of-the-art result.
- Have abstractive abilities, but attaining higher levels of abstraction remains an open research question.
Acknowledgement

- Abigail See and et al. - Get To The Point: Summarization with Pointer-Generator Networks
- Dr. Michael J. Garbade - A Quick Introduction to Text Summarization in Machine Learning
- Pranay and et al. - Text Summarization in Python: Extractive vs. Abstractive techniques revisited
- Eric Ondenyi - Extractive Text Summarization Techniques With sumy
Thank you!!
Deep Reinforced Model for Abstractive Text Summarization

Srihari Veeraraaghavan
Text Summarization

- What is text summarization?
- What are the types of text summarization? *Abstractive and Extractive*
- Most text summarization models are extractive in nature
- Abstractive summarization models are all based on a regular neural encoder-decoder architecture (usually some variations of these)

**Problem:**
- Can only generate **very short summaries** (<75 characters) and is usually used with **one or two input sentences**
Prior Models

Neural Encoder-Decoder sequence models

● Use LSTMs to encode input sentence into a fixed vector, create a new output sequence from that vector using another RNN
● Inputs: Word embeddings - convert language tokens to vectors
● Attention mechanisms - scalable and performant
● Problem: Use fixed input and output vocabulary - prevents learning new words
Prior Models

RL for Sequence Generation

- RL is used when the metric to optimize is not differentiable
- Apply RL to train various RNN based models
- Critic model - to predict rewards and stabilize objective function gradients
Proposed Solution

- **Intra-attention model**
  - Intra-temporal attention (Encoder): Records previous attention weights at each time step \( t \)
  - Sequential intra-attention model (Decoder): Takes care of words already generated by the decoder

- **Objective function**
  - Maximum-likelihood cross-entropy loss + rewards from policy gradient RL
Model
Intra-temporal attention on input sequence

- Attend over specific parts of encoded input sequence at a time step
- Impact:
  - Prevents model from attending over same parts of input on different decoding steps
  - Reduce amount of repetitions over long documents
- Attention score:

\[ e_{ti} = f(h^d_t, h^e_i) \]
Intra-temporal attention on input sequence

- Bilinear function: \[ f(h_t^d, h_i^e) = h_t^d W_{\text{attn}} h_i^e. \]

- Normalize attention weights:
  \[
  e'_{ti} = \begin{cases} 
  \exp(e_{ti}) & \text{if } t = 1 \\
  \frac{\exp(e_{ti})}{\sum_{j=1}^{t-1} \exp(e_{ji})} & \text{otherwise.}
  \end{cases}
  \]

- Normalized attention vector scores and input context vector:
  \[
  \alpha_{ti}^e = \frac{e'_{ti}}{\sum_{j=1}^{n} e'_{tj}} \quad c_t^e = \sum_{i=1}^{n} \alpha_{ti}^e h_i^e.
  \]
Intra-decoder attention

- Incorporates information about the previously decoded sequence into the decoder
- Introduces an intra-attention mechanism
- For each decoding step $t$, our model computes a new decoder context vector $c_t^{(d)}$

\[
e^{d}_{tt'} = h^d_T W^d_{\text{attn}} h^d_{t'}
\]

\[
\alpha^{d}_{tt'} = \frac{\exp(e^{d}_{tt'})}{\sum_{j=1}^{t-1} \exp(e^{d}_{tj})}
\]

\[
c_t^d = \sum_{j=1}^{t-1} \alpha^{d}_{tj} h^d_j
\]
Token Generation and Pointer

- Primarily used to copy rare or unseen from the input sequence
- Switch function is used to decide between the two

<table>
<thead>
<tr>
<th>Token Generation</th>
<th>Pointer Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(y_t</td>
<td>u_t = 0) = \text{softmax}(W_{out}[h_t^d</td>
</tr>
</tbody>
</table>

- Probability distribution function:

$$p(y_t) = p(u_t = 1)p(y_t | u_t = 1) + p(u_t = 0)p(y_t | u_t = 0).$$
Sharing Decoder Weights and Repetition avoidance

- Weight sharing between embedding matrix and $W_{out}$ matrix of token generation
- Allows token generation function to use syntactic and semantic information
- Ground truth summaries never contain the same trigram twice
- Force the decoder to ensure this constraint by setting $p(y_t) = 0$ if the trigram already exists in the previous decoded sequence
Supervised learning with teacher forcing

- Minimizes maximum-likelihood at each decoding step
  \[ L_{ml} = - \sum_{t=1}^{n'} \log p(y_t^* | y_1^*, \ldots, y_{t-1}^*, x) \]

- Problems?
  - Exposure bias
  - Large number of potentially valid summaries
Policy Learning

- Learn a policy that minimizes a specific discrete metric instead
- Uses self-critical policy gradient training algorithm
- Produces two separate output sequences at each time step
  - $y^s$ - sampled from $p(y^s_t | y^s_1, \ldots, y^s_{t-1}, x)$ probability distribution at each decoding time step
  - $\hat{y}$ - baseline output
- Equation:

$$L_{rl} = (r(\hat{y}) - r(y^s)) \sum_{t=1}^{n'} \log p(y^s_t | y^s_1, \ldots, y^s_{t-1}, x)$$
Mixed Training Objective Function

- Optimizing for a specific metric does not guarantee increase in quality and readability
  - ROUGE: n-gram overlap between generated summary and reference sequence
  - Human readability: Language model
- Calculation of \( p(y_t) \) based on previous predicted sequence \( \{y_1 \ldots y_{t-1}\} \) and input sequence \( x \) can assist the algorithm to generate more natural summaries
- Equation:
  \[
  L_{\text{mixed}} = \gamma L_{rl} + (1 - \gamma) L_{ml}
  \]
Experiments

- Maximum Likelihood (ML) training with and without intra-decoder attention
- Initialize model with best ML parameters vs RL with mixed objective learning
- ROUGE Metrics:
  - Full length F-1 score of ROUGE-1, ROUGE-2 and ROUGE-L
  - ROUGE-L as reinforcement reward
Quantitative Analysis

- Improves relative score on ROUGE-1 for long ground truth summaries
# Quantitative metrics

<table>
<thead>
<tr>
<th>Model</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead-3 (Nallapati et al., 2017)</td>
<td>39.2</td>
<td>15.7</td>
<td>35.5</td>
</tr>
<tr>
<td>SummaRuNNer (Nallapati et al., 2017)</td>
<td>39.6</td>
<td>16.2</td>
<td>35.3</td>
</tr>
<tr>
<td>words-Ivt2k-temp-att (Nallapati et al., 2016)</td>
<td>35.46</td>
<td>13.30</td>
<td>32.65</td>
</tr>
<tr>
<td>ML, no intra-attention, no trigram avoidance</td>
<td>35.15</td>
<td>13.28</td>
<td>32.13</td>
</tr>
<tr>
<td>ML, no intra-attention</td>
<td>37.86</td>
<td>14.69</td>
<td>34.99</td>
</tr>
<tr>
<td>ML, with intra-attention</td>
<td>38.30</td>
<td>14.81</td>
<td>35.49</td>
</tr>
<tr>
<td>RL, with intra-attention</td>
<td>41.16</td>
<td>15.75</td>
<td>39.08</td>
</tr>
<tr>
<td>ML+RL, with intra-attention</td>
<td>39.87</td>
<td>15.82</td>
<td>36.90</td>
</tr>
</tbody>
</table>

Table 1: Quantitative results for various models on the CNN/Daily Mail test dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML, no intra-attention, no trigram avoidance</td>
<td>42.85</td>
<td>26.22</td>
<td>39.09</td>
</tr>
<tr>
<td>ML, no intra-attention</td>
<td>44.26</td>
<td>27.43</td>
<td>40.41</td>
</tr>
<tr>
<td>ML, with intra-attention</td>
<td>43.86</td>
<td>27.10</td>
<td>40.11</td>
</tr>
<tr>
<td>RL, no intra-attention</td>
<td>47.22</td>
<td>30.51</td>
<td>43.27</td>
</tr>
<tr>
<td>ML+RL, no intra-attention</td>
<td>47.03</td>
<td>30.72</td>
<td>43.10</td>
</tr>
</tbody>
</table>

Table 2: Quantitative results for various models on the New York Times test dataset
Jenson Button was denied his 100th race for McLaren after an ERS prevented him from making it to the start-line. It capped a miserable weekend for the Briton; his time in Bahrain plagued by reliability issues. Button spent much of the race on Twitter delivering his verdict as the action unfolded. 'Kimi is the man to watch,' and 'loving the sparks,' were among his pearls of wisdom, but the tweet which courted the most attention was a rather mischievous one: 'Ooh is Lewis backing his team mate into Vettel?' he quizzed after Rosberg accused Hamilton of pulling off such a manoeuvre in China. Jenson Button waves to the crowd ahead of the Bahrain Grand Prix which he failed to start Perhaps a career in the media beckons... Lewis Hamilton has out-qualified and finished ahead of Nico Rosberg at every race this season. Indeed Rosberg has now beaten his Mercedes team-mate only once in the 11 races since the pair infamously collided in Belgium last year. Hamilton secured the 36th win of his career in Bahrain and his 21st from pole position. Only Michael Schumacher (40), Ayrton Senna (29) and Sebastian Vettel (27) have more. (...)

**Ground truth summary**

Button denied 100th race start for McLaren after ERS failure. Button then spent much of the Bahrain Grand Prix on Twitter delivering his verdict on the action as it unfolded. Lewis Hamilton has out-qualified and finished ahead of Mercedes team-mate Nico Rosberg at every race this season. Bernie Ecclestone confirms F1 will make its bow in Azerbaijan next season.

**ML, with intra-attention (ROUGE-1 41.58)**

Button was denied his 100th race for McLaren. ERS prevented him from making it to the start-line. The Briton. He quizzed after Nico Rosberg accused Lewis Hamilton of pulling off such a manoeuvre in China. Button has been in Azerbaijan for the first time since 2013.

**RL, with intra-attention (ROUGE-1 50.00)**

Button was denied his 100th race for McLaren after an ERS prevented him from making it to the start-line. It capped a miserable weekend for the Briton. Button has out-qualified. Finished ahead of Nico Rosberg at Bahrain. Lewis Hamilton has. In 11 races. The race. To lead 2,000 laps. In. And.

**ML+RL, with intra-attention (ROUGE-1 44.00)**

Button was denied his 100th race for McLaren. The ERS prevented him from making it to the start-line. Button was his team mate in the 11 races in Bahrain. He quizzed after Nico Rosberg accused Lewis Hamilton of pulling off such a manoeuvre in China.

Table 3: Example from the CNN/Daily Mail test dataset showing the outputs of our three best models after de-tokenization, re-capitalization, replacing anonymized entities, and replacing numbers. The ROUGE score corresponds to the specific example.
Qualitative Analysis

- Human Evaluation Criteria:
  - Relevance
  - Readability
- RL Model: short and truncated sentences towards the end of sequences
- RL + ML model: Highest readability and relevance scores
Qualitative Analysis

<table>
<thead>
<tr>
<th>Model</th>
<th>Readability</th>
<th>Relevance</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML</td>
<td>6.76</td>
<td>7.14</td>
<td>84.46</td>
</tr>
<tr>
<td>RL</td>
<td>4.18</td>
<td>6.32</td>
<td>16417.68</td>
</tr>
<tr>
<td>ML+RL</td>
<td>7.04</td>
<td>7.45</td>
<td>121.07</td>
</tr>
</tbody>
</table>

Table 5: Comparison of human readability scores on a random subset of the CNN/Daily Mail test dataset. All models are with intra-decoder attention.
Conclusion

Intra-attention decoder and combined training objective can be applied to other sequence-to-sequence tasks
Discussion

What other sequence-to-sequence tasks can you think of where this model will be useful?
What are we going to talk about?

1. Explain a current popular Q&A challenge (Stanford Question Answering Dataset)
2. Design our own Q&A challenge
3. What you need to know to get started on a state of the art Q&A model
What are going to talk about?

1. Explain a current popular Q&A challenge (Stanford Question Answering Dataset)
2. Design our own Q&A challenge
3. What you need to know to get started on a state of the art Q&A model
Stanford Question and Answering Dataset (SQuAD)

Challenge: Given a paragraph and a question, highlight the parts of the paragraph that contain the answer to the question.

Example:

How did some suspect that Polo learned about China instead of by actually visiting it?

**Answer:** through contact with Persian traders
Some questions have no answer

Passage: Mergesort is a sorting algorithm that uses the divide and conquer techniques to reduce the time complexity of sorting to \( O(n\log n) \). It is often taught in introductory programming or algorithm classes.

Question: What is the time complexity of Bubblesort?

Answer: Answer not found
Given the following passage, what would be the answer to the question?

Passage: I went on a walk to the park today. The park was sunny and the grass was green. There were dogs playing by the river.

Question: What color was the grass?

Answer: ???
Given the following passage, what would be the answer to the question?

Passage: I went on a walk to the park today. The park was sunny and the grass was green. There were dogs playing by the river.

Question: What color was the grass?

Answer: The grass was green or green
Given the following passage, what would be the answer to the question?

Passage: The pecan is a species of hickory native to northern Mexico and the southern United States in the region of the Mississippi River. The tree is cultivated for its seed in the southern United States, primarily in Georgia, and in Mexico which produces nearly half of the world total.

Question: What area of Georgia do pecans grow?

Answer: ???
Given the following passage, what would be the answer to the question?

Passage: The pecan is a species of hickory native to northern Mexico and the southern United States in the region of the Mississippi River. The tree is cultivated for its seed in the southern United States, primarily in Georgia, and in Mexico which produces nearly half of the world total.

Question: What area of Georgia do pecans grow?

Answer: No appropriate answer
How do we score proposed solutions?

**Exact Match (EM)** – Does the proposed answer match one of the correct solutions exactly.

**F1 Score** – Average overlap of prediction and solution
How to Calculate the F1 Score

\[
F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

Precision = \# of proposed words that are in the actual solution / \# of proposed words

Recall = \# of proposed words that are in the actual solution / \# of words in solution

The ordering of words doesn’t matter
Calculate the F1 Score

\[
F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

Precision = \# of proposed words that are in the actual solution / \# of proposed words

Recall = \# of proposed words that are in the actual solution / \# of words in solution

The ordering of words doesn’t matter

Actual Solution:
- the dog jumped on the table

Proposed Solution:
- we saw the dog jumped

F1 Score: ???
Calculate the F1 Score

\[ F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \]

Precision = # of proposed words that are in the actual solution / # of proposed words

Recall = # of proposed words that are in the actual solution / # of words in solution

The ordering of words doesn’t matter

Actual Solution:
- the dog jumped on the table

Proposed Solution:
- we saw the dog jumped

F1 Score:
Precision = 3 / 5
Recall = 3 / 6

F1 Score = \( 2 \times \frac{9/30}{33/30} \)
How was the SQuAD made?

Spend around 4 minutes on the following paragraph to ask 5 questions! If you can’t ask 5 questions, ask 4 or 3 (worse), but do your best to ask 5. Select the answer from the paragraph by clicking on ‘Select Answer’, and then highlight the smallest segment of the paragraph that answers the question.

Oxygen is a chemical element with symbol O and atomic number 8. It is a member of the chalcogen group on the periodic table and is a highly reactive nonmetal and oxidizing agent that readily forms compounds (notably oxides) with most elements. By mass, oxygen is the third-most abundant element in the universe, after hydrogen and helium. At standard temperature and pressure, two atoms of the element bind to form dioxygen, a colorless and odorless diatomic gas with the formula O₂.

1. Diatomic oxygen gas constitutes 20.8% of the Earth's atmosphere. However, monitoring of atmospheric oxygen levels show a global downward trend, because of fossil-fuel burning. Oxygen is the most abundant element by mass in the Earth's crust as part of oxide compounds such as silicon dioxide, making up almost half of the crust's mass.

When asking questions, avoid using the same words/phrases as in the paragraph. Also, you are encouraged to pose hard questions.

- Passages are scrapped from wikipedia
- Questions and answers manually created using Amazon Mechanical Turk
Cracking the SQuAD Challenge

- You need a good language model to identify synonyms
- Then pattern match the sentence with the question

...Students thronged to Wittenberg to hear Luther speak....

Who went to Wittenberg to hear Luther speak?
Cracking the SQuAD Challenge

- Determine whether the question is asking for a noun, a verb, or an adjective
- If the question expects a noun find the sentence that has the most words in common with the question and find the noun not in the question

**Passage Segment**

...The European Parliament and the Council of the European Union have powers of amendment and veto during the legislative process...

**Question**

Which *governing bodies* have veto power?
Biases: When in doubt go with the average
What are going to talk about?

1. Explain a current popular Q&A challenge (Stanford Question Answering Dataset)
2. Design our own Q&A challenge
3. What you need to know to get started on a state of the art Q&A model
Making your own Question Answering Dataset Challenge

A PhD student made the SQuAD challenge. Imagine you are a second year PhD student assigned with making the next industry standard Q&A challenge.

How would you make it?

How could you simplify the problem?

How could you make the problem more challenging?

How could you avoid using mechanical turk?

How could you get rid of biases in answers?
How do we get to the top of the SQuAD leaderboard?

If your project for this class was to get to the top of the SQuAD leaderboard, what kind of model would you build?
What are going to talk about?

1. Explain a current popular Q&A challenge (Stanford Question Answering Dataset)
2. Design our own Q&A challenge
3. What you need to know to get started on a state of the art Q&A model
If we are trying to get to the top of the SQuAD leaderboard we should probably know about BERT.
What is BERT?

BERT stands for **Bidirectional Encoder Representations from Transformers**.

BERT is a language model, which means it takes your words or phrases and encodes them into a more meaningful representation.
Review question:

Which statements are true about transformers?

A. Transformers use attention
B. Transformers use recurrent units
C. Transformers ignore the position of the inputs (bag of words)
Review question:

Which statements are true about transformers?

A. Transformers use attention True
B. Transformers use recurrent units False
C. Transformers ignore the position of the inputs (bag of words) False, encodes the position using a position vector
Model Architecture of BERT

Same as Attention is All You Need

Paper
Key Contributions of BERT: Self Supervised Training

BERT is pre trained using two self supervised task.

Task 1: Masked LM

Task 2: Next Sentence Prediction
Masked LM

Given an input sentence:

“[Start] The dog jumped over the log [End]”

Randomly mask out 15% of the words:

“[Start] The dog jumped [Mask] the log [End]”

Predict the value of the masked out words:

Prediction = “over”
Next Sentence Prediction

In this task the network has two predict whether the input is two sequential sentences or two random sentences.

Example of two sequential sentences:
“[Start]My laptop ran out of battery.[Sep] I guess it is time to go home [End]”

Example of two random sentences:
“[Start] My laptop ran out of battery.[Sep] Add one cup of sugar to the batter [End]”
Pre-training on a Massive Corpus of Text

You can train a BERT model using these self supervised tasks on a massive corpus of text. In the paper they train on all of wikipedia.

You can then fine tune your model to your specific task.
BrainStorm Other Self Supervised Tasks

Can you think of other self supervised tasks for training a language model?

What would be an analogous self supervised task for vision, audio, or graph data?
BERT is Truly Bidirectional

The traditional self supervised learning task is to predict the next token. Because of this, the model shouldn’t take future tokens in as input. However with masked ML and next sentence prediction, this is not a problem so BERT looks at both the previous tokens and future tokens.
Using BERT for Q&A

BERT is an embedding layer similar to word2vec or elmo. It is not a Question Answering Model in and of itself.

How do we use BERT for Q&A?
If you can identify sentences in your passage that have similar semantics to your question then you are half have there. Since BERT gives us good embeddings we can compute a correlation matrix between our passage and our question.
**Affinity Matrix (Measures Similarity of Tokens)**

Example of computing an affinity matrix: Words embeddings have 3 dimensions

Question has q tokens:

```
[[0, 0, 1],
 [1, 0, 0],
 [1, 1, 1]], q x 3
```

Passage has p tokens:

```
[[2, 0, 0],
 [0, 2, 0],
 [1, 1, 0],
 [0, 1, 1]], p x 3
```

Affinity matrix is q by p:

```
[[0, 0, 0, 1],
 [2, 0, 1, 0],
 [2, 2, 2, 2]]
```

Use the affinity matrix to determine which word of the passage is most similar to the first word of the question: ???
Affinity Matrix (Measures Similarity of Tokens)

Example of computing an affinity matrix: Words embeddings have 3 dimensions

Question has q tokens:
[[0, 0, 1],
 [1, 0, 0],
 [1, 1, 1]], q x 3

Passage has p tokens:
[[2, 0, 0],
 [0, 2, 0],
 [1, 1, 0],
 [0, 1, 1]], p x 3

Affinity matrix is q by p:
[[0, 0, 0, 1],
 [2, 0, 1, 0],
 [2, 2, 2, 2]]

Use the affinity matrix to determine which word of the passage is most similar to the first word of the question: The last word of the passage
Using the Affinity Matrix for Attention

By using the affinity matrix to calculate attention, we can focus on the parts of the passage that are most similar to the question. This gets us on our way to a state-of-the-art Q&A model.