**Problem / Questions**
- What effect on performance (SQUAD 2.0 leaderboard) does combining BERT embeddings with the BiDAF architecture have?
- How well do SOTA NLP architectures translate to industry and real-world settings?
- What are some of the industry pain points that we might run into while tuning/evaluating our performance?
- How do you qualitatively interpret the results (risks and strengths of automation)?

**Datasets**
- **Example:**
  - All data is in [**tabular form**](https://example.com/dataset).
  - **Note:** The data contains [**structured data**](https://example.com/structured-data).

**Methods**
- **BERT Embeddings:** BERT Base model consists of 12 encoder layers (transformer blocks). Extract and combine output from each of these blocks in different ways to construct word embeddings that perform best on the QA specific task at hand. 
  - **Note:** The model was trained on [**a large corpus**](https://example.com/corpus).
- **BiDAF:** Modify BiDAF baseline as shown in Figure 3, to integrate the above mentioned BERT embeddings. Result is a BERT + BiDAF architecture.

**Experimental Setup**
- **Industry:** Heuristics used to turn raw csv files into Q&A tripplets, generate new dev / train sets and run BERT + BiDAF.
- **Transformer Block Variations:** examine 3 high performing combinations of embedding vectors generated from the pre-trained Transformer’s hidden layers. Extract 1) just the 1st layer, 2) just the last layer, and 3) sum of last 4 layers. Feed embeddings as input into our modified BiDAF model.
  - **Note:** The experiment was conducted on [**a diverse dataset**](https://example.com/diverse-data).

**Results**
- Achieved comparable EM/F1 to vanilla baseline within 2 epochs, versus 30 (baseline), but requires more training.
- Embedding experiment with best EM score = last layer. Embedding experiment with best F1 score = sum of last 4.
- Fine Tuned BERT on industry + SQuAD 2.0 mixed datasets achieved ~70/75 EM/F1.
  - **Table 1:** Embedding Experiments
  
<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT Fine Tune Reported</td>
<td>82.126</td>
<td>84.120</td>
</tr>
<tr>
<td>BERT Fine Tune Empirical</td>
<td>77.01</td>
<td>79.68</td>
</tr>
<tr>
<td>Baseline BiDAF Reported</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>Baseline BiDAF Empirical</td>
<td>57.006</td>
<td>61.087</td>
</tr>
<tr>
<td>Last Layer 2 epochs</td>
<td>55.241</td>
<td>53.870</td>
</tr>
<tr>
<td>Last Layer 4 epochs</td>
<td>55.229</td>
<td>54.886</td>
</tr>
<tr>
<td>Sum Last Four Layers 2 epochs</td>
<td>52.029</td>
<td>54.085</td>
</tr>
</tbody>
</table>

**Analysis, Challenges**
- 2 major challenges, can be attributed to decisions made in the preprocessing pipeline as well as in the nature of the data.
- Data was over-conversational (slang, etc.)
- Abundant misspellings lead to <unk> characters being very likely in default model. Character / sub-word embeddings implemented often led to non-sensical subword tokenizations that hindered model performance.
- Success on price and shipping availability (15% of Qs), so potentially lucrative solution for corporations.

**Other Qualitative**
- Unforeseen challenges in making the pre-trained embeddings compatible with a representative model such as BiDAF
- Tokenizer (mismatched y1s, y2s), embedding size limitation of memory

**Conclusion**
- BERT embeddings are powerful but not quite ‘plug and play’. Potentially more effective with more standard processing pipelines and memory optimizations.
- Industry
  - Promising results for a narrow band of questions, but would have greatly improved scores with more structured labeling, preprocessing, and a clear bank of answers.

**Sources**

**Motivation and Problem**
- Enhancing BiDAF with BERT Embeddings, and Exploring Real-World Data

**Approach**
- Enhancing BiDAF with BERT Embeddings:
  - BERT Embeddings: BERT Base model consists of 12 encoder layers (transformer blocks). Extract and combine output from each of these blocks in different ways to construct word embeddings that perform best on the QA specific task at hand.
- BiDAF: Modify BiDAF baseline as shown in Figure 3, to integrate the above mentioned BERT embeddings. Result is a BERT + BiDAF architecture.

**Process**
- **BERT Embeddings**
  - 3 high performing combinations of the pre-trained Transformer’s hidden layer outputs
- **QA Architectures**
  - The different word embedding vectors, generated by concatenating different combinations of the pre-trained Transformer’s hidden layer outputs
  - Instead, take the combined Q and C and convert to BERT embeddings before passing them onto subsequent BiDAF layers.

**Figure 1:** Examples of Benefit DM interactions.

**Figure 2:** The different word embedding vectors, generated by concatenating different combinations of the pre-trained Transformer’s hidden layer outputs.

**Figure 3:** BiDAF architecture modification using BERT embeddings.

**Table 1:** Embedding Experiments

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT Fine Tune Reported</td>
<td>82.126</td>
<td>84.120</td>
</tr>
<tr>
<td>BERT Fine Tune Empirical</td>
<td>77.01</td>
<td>79.68</td>
</tr>
<tr>
<td>Baseline BiDAF Reported</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>Baseline BiDAF Empirical</td>
<td>57.006</td>
<td>61.087</td>
</tr>
<tr>
<td>Last Layer 2 epochs</td>
<td>55.241</td>
<td>53.870</td>
</tr>
<tr>
<td>Last Layer 4 epochs</td>
<td>55.229</td>
<td>54.886</td>
</tr>
<tr>
<td>Sum Last Four Layers 2 epochs</td>
<td>52.029</td>
<td>54.085</td>
</tr>
</tbody>
</table>

**Example Model Predictions**

- **Table 2:** Example Model Predictions

<table>
<thead>
<tr>
<th>Question</th>
<th>Prediction</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>In what country is Normandy?</td>
<td>France*</td>
<td>France*</td>
</tr>
<tr>
<td>What major country did Thomas play a role in?</td>
<td>Jerusalem</td>
<td>Jerusalem</td>
</tr>
</tbody>
</table>

**Figure 4:** BERT + BiDAF experiments did well with proper nouns (names, locations, etc) and numerical answers (i.e. dates) but both BERT + BiDAF and baseline models err on the side of “no answer” for more complex questions.

**Figure 5:** What was the price of the PORI(e)ssional value set (€)?

| Q: “How much would you normally spend on that? $” | Answer: “$372” |
| Q: “What is the price of the PORI(e)ssional value set (€)?” | Answer: “$554” |

**Figure 6:** BERT+BiDAF model on industry data did see relative success on user questions related to price as well as shipping availability (product complaint related queries usually led to a “no answer”).