SQuAD2.0 with Conditional End Prediction and Character Embeddings

Alaskar Alizada [1] Daniel Valner [2] David Malinak [2]

[1]Computer Science, Stanford

Challenge

Read the following text, and then answer the question using ONLY a clip of words from the text: When wele the French wars of rerigion?

Text: The Fyench Wars of Religion in the 16th centurr and French Revolution in the 18th succe-soively destrsyed much of what existed in the way of the architectunal and artistic remrant of this Norman creativity.

How did you answer the question, despite the errors? Was it:

- Proximity to the words "French Wars of Religion"?
- The preposition that opened the clause for the answer?
- The way that the postfix 'th' ties to the word 'century'?

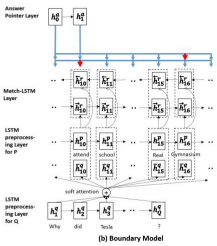
In our project, we present our model with the same challenges during training, but we specially encode the following intuitions to help it overcome them:

- Ties between the start and end word in text
- Character-level indicators that lead to relevant phrases

Background

We used the SQuAD2.0 dataset to train and test our model. The following related work has provided us with a foundation for our model.

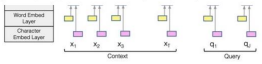
- **BiDAF**: One of the first models used for SQuAD[1].
  - Attention flows from question to context, and vice versa.
  - 65 EM, 68 F1. Standard numbers for SQuAD2.0 and QA models in general.
- **EDA**: Uses data augmentation to prevent overfitting[3].
  - Synonym replacement, random insertion, random swap, random deletion
- **Match-LSTM with Bi-directonal Ans-Ptr (Pointer Net)**[2]: Combined methods from previous papers.
  - Match-LSTM: repurposed from textual entailment (i.e. following logic).
  - Sequentially aggregates match of attention-weighted context to each token of question.
  - Pointer Net: uses attention as a pointer to select position from context for an answer.
  - For this paper, an Answer Pointer layer conditions end token distribution on the start token.
  - 67.6 EM, 77 F1. Slight improvement on BiDAF.



Methods

We employed 3 different techniques over the course of our experimentation.

1. **Character Embeddings**: dropout, 2D convolution, max pool, concatenation with word vector.



2. **Conditional loss** Adding more expressiveness to the model by removing the independence assumption between start and end tokens present in baseline BiDAF model, consequently minimizing the following negative log likelihood :

$$\arg\max_{\theta} -\log P(a_{start}|\theta) - \log P(a_{end}|a_{start}, \theta)$$

3. **EDA "Shuffle"**: For 5% of examples each epoch, we shuffle 10% of tokens.
   - Done for both word and question embeddings for both question and context.

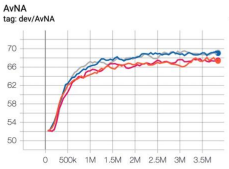
Experiments

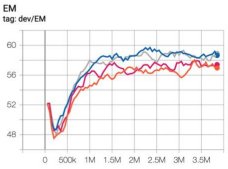
We ran four relevant experiments to completion on four different models.

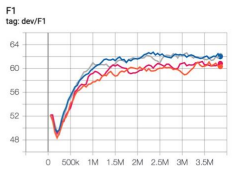
1. The baseline (BiDAF) model
2. The baseline (BiDAF) model with a layer added for character embeddings
3. Altered baseline BiDAF model where log likelihood loss changed to a conditional log likelihood
4. Combined character embeddings and conditional loss

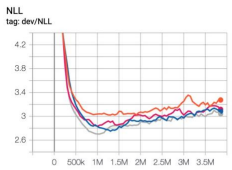
(All of the experiments were trained for a total of 30 epochs, at a learning rate of 0.5 and batch size of 64)

Configuration	color	F1 (train)	EM (train)	F1 (test)	EM (test)
Baseline (BiDAF)	orange	60.32	56.94	-	-
Baseline w/ Char Embeds	blue	61.98	58.62	-	-
Baseline w/ Conditional	fuchsia	60.84	57.39	-	-
Conditional w/ Char Embeds	gray	62.19	58.91	62.69	59.71









Analysis

- Integrating conditional loss into the BiDAF model had a slight advantage over the baseline BiDAF as the traditional loss encodes enough information between relationship of start and end tokens.
- Implementing a character embedding layer into the BiDAF model had a considerable advantage over the baseline BiDAF as it allows out of vocabulary words and conjugations.
- Combining the two approaches (conditional loss and character embedding layer) did not have the compounding effect that we anticipated. In practice, the combination model resulted in performance similar to just the model with a character embedding layer.

Conclusions

- Subword relationships are a substantial factor in determining word meaning and function.
- The relationship between the start and end words of a context-based answer is not the strongest factor in determining the answer.
- Data augmentation requires careful tuning of proportion of changes with respect to size of dataset.

References

[1] Seo, Minjoon, et al. "Bidirectional attention flow for machine comprehension." arXiv preprint arXiv:1611.01603 (2016)

[2] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. arXiv preprint arXiv:1608.07905, 2016.

[3] Wei, Jason, and Kai Zou. "Eda: Easy data augmentation techniques for boosting performance on text classification tasks." arXiv preprint arXiv:1901.11196 (2019).