Optimizing Match-LSTM for SQuAD v2.0

Peter Nsaka, James Dong, Alex Lee

Problem

After the SQuAD v1.0 dataset was created as a standardized dataset of evaluating machine comprehension (MC) / question answering (QA) models, many approaches were explored including the Match-LSTM / Pointer Network model that our project utilizes. However, with the new SQuAD 2.0 dataset that includes non-answerable questions, we attempted to adapt this approach to the v2.0 dataset to perform better than the baseline BIDAF model.

Background

- Match-LSTM [1] w/ Pointer Network - end to end model for MC that takes in context paragraph and question and points to a section of paragraph as answer
- This approach already performs comparably to baseline performance of the BIDAF model on answerable questions

Methods

- On a high level, our model consists of a pre-processing layer, a Match-LSTM layer and an output layer.
- During pre-processing of dataset, a no-answer token is added with "id" value of 1". This id was prepended to each sentence's token, part-of speech, and entity tag list for every context.
- When predicting an answer, if pstart(0) ·pend(0) is greater than any predicted answer span, the model predicts no-answer. Otherwise we predict the highest probability span as usual

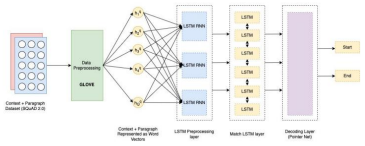

Fig 1: Match LSTM Model with Pointer Net Model

Experiments

model	EM	F1	AvNA
BiDAF	**57.91**	**61.4**	**68.32**
Match-LSTM	48.63	51.6	59.38

Table 2: BiDAF versus Match-LSTM.

Ablation Study

model	EM	F1	AvNA
Match-LSTM	48.63	51.6	59.38
Match-LSTM features off	**50.3**	**53.33**	**60.5**
Match-LSTM ans upsampled	47.01	50.55	58.49
Match-LSTM ans upsampled*	**51.88**	**54.51**	**60.46**
Match-LSTM unans upsampled	52.07	52.11	52.4
Match-LSTM unans upsampled w/o search	52.02	52.1	52.42

Table 5: Match-LSTM ablation study.

model	ans prec	ans rec	unans prec	unans rec
Match-LSTM	56.07	**70.03**	64.31	49.59
Match-LSTM features off	**57.18**	69.7	**65.15**	**52.05**
Match-LSTM ans upsampled	54.3	**84.02**	**70.48**	35.04
Match-LSTM ans upsampled*	**57.68**	65.43	63.77	**55.90**
Match-LSTM unans upsampled	61.04	1.62	52.29	99.05
Match-LSTM unans upsampled w/o search	59.78	1.89	52.31	98.83

Table 6: Match-LSTM ablation detailed study.

model	EM ans	F1 ans	unans pred	unans actual
Match-LSTM	67.96	76.8	40.19	52.12
Match-LSTM features off	69.43	78.52	**41.64**	52.12
Match-LSTM ans upsampled	**71.45**	**80.25**	25.91	52.12
Match-LSTM ans upsampled*	**72.58**	**80.98**	**45.69**	52.12
Match-LSTM unans upsampled	57.45	61.98	98.73	52.12
Match-LSTM unans upsampled w/o search	56.36	64.25	98.49	52.12

Table 7: Match-LSTM ablation detailed study ctnd.

Analysis

	EM	F1
Ans len = 1	**71.44**	**74.12**
Ans len > 1	54.27	64.05
No ans	25.91	25.91

Table 8: Prediction analysis on Match-LSTM model trained on answerable questions upsampled.

- Match-LSTM performs better on answerable questions when it predicts answerable than unanswerable questions, but the precision is low
- Recall for unanswerable is a lot lower than that of the answerable
- Training on answerable upsampled:
  - Pro: boosts answerable recall, accuracy of answer span
  - Con: overall performance worsened due to lower unanswerable recall.
- Training on unanswerable upsampled:
  - Con: model predicts almost everything as unanswerable
- Training on answerable upsampled, double counting loss on unanswerable
  - Boosts AvNA, gets the benefits from both world, a good answer precision and better unanswerable recall
- Match-LSTM performs much better on the short and answerable questions

Conclusion

- Match-LSTM works better on SQuAD v1.0 than SQuAD v2.0
- It lacks the ability to distinguish between answerable and unanswerable
- It is still reliable for predicting precise spans for short answerable questions

References

[1] Shuohang Wang and Jing Jiang, "Machine Comprehension Using Match-LSTM and Answer Pointer". arxiv.org/abs/1608.07905