R-Net and Friends

Megan Worrel, Amrita Palaparthi, Ani Vegesana

Department of Computer Science, Stanford University
Research Mentor: Allan Zhou

Problem

Task: Question Answering (Reading Comprehension) is the task of automatically answering a question given a paragraph of relevant context

Importance: Question Answering is critical to determining how well models can **understand** and **draw information from** text

Contribution: R-net explores the effect of additional forms of attention on SQuAD performance; however, it does not combine these with meaningful additional input features. We explore the gains in performance on SQuAD 2.0 that can be achieved through the simultaneous use of **R-net attention mechanisms** and **feature engineering** as proposed by DrQA, coupled with hyperparameter tuning and ensembling techniques.

Background

Problem Setup and Notation: Given the i'th context paragraph c_i and question q_i, predict the start and end indices of the answer within the context if it exists. These indices are represented as the logits p_{start} and p_{end}.

Training Procedure: All models were trained for 30 epochs, using the AdaDelta optimizer with cross-entropy loss.

Evaluation: We use F1 score as the primary metric for evaluation on the SQuAD 2.0 validation and test datasets.

Baseline: We evaluate our results against a baseline Bidirectional Attention Flow (BiDAF) model using word embeddings, which achieves an F1 score of 60.65.

Methods

Additions to the baseline:

1. Character-level embeddings
2. Feature Engineering
3. R-net gated and self attention
4. Hyperparameter tuning
5. Ensembling

Hyperparameter Tuning

Learning Rate Decay: We used a decay rate of 0.5 and patience (number of evaluation steps of decreasing F1 before the change in LR was applied) of 3.

Initial Learning Rate: We tested model performance using an initial LR of 1 (as used in BiDAF) and 0.5 (as used in R-net)

Hidden Layer Size: We experimented with hidden layer sizes of 100 (BiDAF) and 75 (R-net).

Dropout: Using random search, we experimented with dropout values of 0.1, 0.37, and 0.52

DrQA Additional Input Features

$$\boxed{f_{exact\_match}(c_i) = \mathbb{I}(c_i \in q)}$$

1. Exact Match Features
   a. Exact match: *context word found exactly in question*
   b. Lowercase match: *lowercase context word found in lowercase question*
   c. Lemma match: *lemmatized context word found in lemmatized question*
2. Aligned Question Embedding

$$\boxed{a_{i,j} = \frac{\exp(\alpha(\mathbf{E}(c_i)) \cdot \alpha(\mathbf{E}(q_j)))}{\sum_{j'} \exp(\alpha(\mathbf{E}(c_i)) \cdot \alpha(\mathbf{E}(q_{j'})))} = \text{softmax}_{:,j}\left[\alpha(\mathbf{E}(c))\alpha(\mathbf{E}(q))^\top\right]}$$

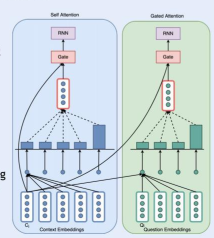
for context embeddings $\mathbf{E}(c_j)$,
question embeddings $\mathbf{E}(q_j)$,
and a dense layer with ReLU nonlinearity α

Ensembling
Models: 9 models that achieved higher performance than the BiDAF baseline with character-level embeddings
Criteria: Maximum confidence score for predicted end index p_{end}

RNet Gated and Self Attention

"Mutually Recursive"

- First we implemented RNet gated and self attention as described in the RNet paper
- RNN output and context-question were computed through mutual recursion, requiring inefficient manual looping in our implementation

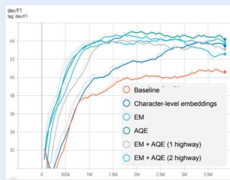


"Non-Recursive"

- For more efficient training, we switched to a non-recursive approach
- Attention computation occurs as an input to the LSTM
- This eliminates the need for expensive manual iteration, leveraging much faster matrix computation and gpu processing

Experiments

Feature Engineering Results



Model	F1
Baseline	60.65
+ Character-level Embeddings	63.76
+ EM	64.34
+ AQE	64.52
+ EM & AQE (1 highway)	65.04
+ EM & AQE (2 highway)	**65.16**

RNet Results

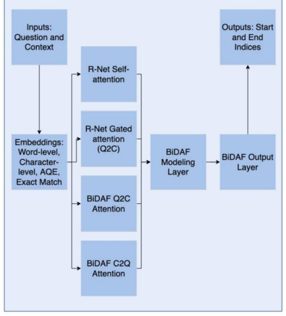
BiDAF Attention	RNet Gated Attention	RNet Self Attention	BiDAF Output Layer	RNet Output Layer	F1
■			■		**65.16**
	■	■		■	64.36ǂ
	■	■		■	61.61*
	■	■	■		62.32*
	■		■		63.68
■		■	■		63.35*
■	■	■	■		65.02†

* attention layers are in series
† attention layers are in parallel
ǂ "Mutually Recursive" RNet implementation
All other experiments utilize "Non-Recursive" implementation

Ensembling Results

Number of Models	F1 Score
5	67.60
8	68.22
9	68.32

Best Model Architecture: Parallel Model



RNN character embeddings: We switched the BiDAF CNN character embedding layer with a bidirectional LSTM. We saw no discernible increase in performance.

Analysis

Summary of Highest-Performing Improvements:

Incremental Improvement	Maximum F1 Score
Baseline	60.65
Feature Engineering	65.16
R-Net Attention (untuned)	65.02
Hyperparameter Tuning	65.39
Ensembling	68.32

- Exact match and AQE each individually boosted F1 score, but combining them did not result in significant improvement. This aligns with DrQA's finding that these two features play complementary roles
- BiDAF's attention mechanism performed more effectively than R-net gated attention alone, as they play a similar role, but BiDAF uses multiplicative rather than additive attention

Out-of-vocabulary Words

- **Question:** Who designed the garden for the University Library?
  Context: Another important library – the University Library, founded in 1816, is home to over two million items. The building was designed by architects Marek Budzyński and Zbigniew Badowski and opened on 15 December 1999. It is surrounded by green. The University Library garden, designed by Irena Bajerska, was opened on 12 June 2002. It is one of the largest and most beautiful roof gardens in Europe with an area of more than 10,000 m2 (107,639.10 sq ft), and plants covering 5,111 m2 (55,014.35 sq ft). As the university garden it is open to the public every day.
  Answer: Irena Bajerska
  Prediction (baseline): Marek Budzyński and Zbigniew Badowski
  Prediction (char embeddings): Irena Bajerska

Anaphora

- **Question:** What is the term for a task that generally lends itself to being solved by a computer?
  Context: Computational complexity theory is a branch of the theory of computation in theoretical computer science that focuses on classifying computational problems according to their inherent difficulty, and relating those classes to each other. A computational problem is understood to be a task that is in principle amenable to being solved by a computer, which is equivalent to stating that the problem may be solved by mechanical application of mathematical steps, such as an algorithm.
  Answer: computational problems
  Prediction (BiDAF): N/A
  Prediction (RNet self-attention): computational problem

Conclusions

- Implementing **R-Net attention mechanisms** in conjunction with **DrQA's additional input features** results in a substantial increase in performance over our baseline model on Question Answering for SQuAD 2.0
- Each of the **four main components** of our approach - *DrQA additional input features, R-Net attention mechanisms, hyperparameter tuning, and ensembling* - **built upon one another** to provide an incremental increase in F1 score
- **Ensembling** the variety of models we trained based on confidence score enabled our approach to perform well on **a wide range of inputs**, further augmenting performance

References

[1] Jason Wetson,] Antoine Bordes, Danqi Chen, Adam Fisch. Reading wikipedia to answer open-domain questions. In Association for Computational Linguistics (ACL), 2017

[2] Microsoft Research Asia Natural Language Computing Group. R-net: Machine reading comprehension with self-matching networks. In Association for Computational Linguistics (ACL), 2017. 5