# QUESTION ANSWERING

## EXAMINING THE TOP TRENDS AND METHODS FROM THE SQUAD TASK

## INTRODUCTION

- **Extractive question answering** is a machine reading comprehension problem which has exploded in popularity with the release of the Stanford Question Answering Dataset (**SQuAD**)
- **Bi-Directional Attention Flow for Machine Comprehension (BiDAF)** is a well known and good performing architecture for this task
- **Self Attention** is a promising component used in a different popular architecture **R-Net**
- We aim to combine these to build a high performing architecture for the SQuAD task. We investigate the importance of different components and experiment with transfer learning using **ELMo** contextual embeddings.

## PROBLEM

- Given a **context** paragraph C, and a **question** Q, provide the token **span** C[i:j] which answers the question.
- There are two metrics reported for this task:
  - **Exact Match** (**EM**) measures the percentage of questions that received the *exact* correct answer span prediction
  - **F1** gives a averaged measure of overlap between the predicted and ground truth answer spans. To calculate this both are treated as a bag of tokens.

## DATASET

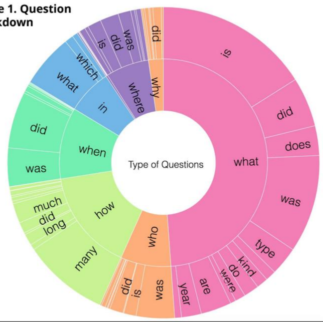
- SQuAD v1.1 consist of 107,785 question-answer pairs on 536 articles from Wikipedia where the answer to every question is a span from the associated article passage

**Example:**
Similarly, it is not known if L (the set of all problems that can be solved in logarithmic space) is strictly contained in P or equal to P. Again, there are many complexity classes between the two, such as NL and NC, and it is not known if they are distinct or equal classes.

**What lies between L and P that prevents a definitive determination of the relationship between L and P?**
*Ground Truth Answers:* complexity classes, many complexity classes, many complexity classes

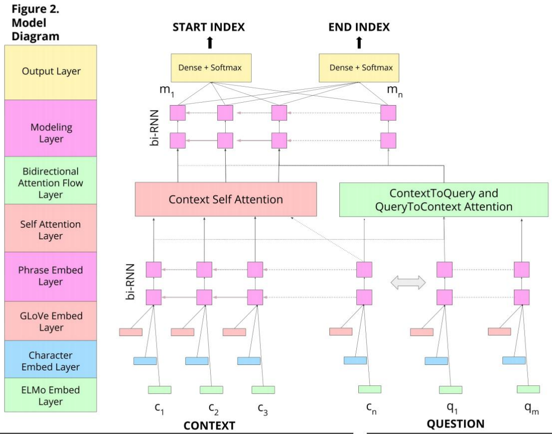
**What variable is associated with all problems solved within logarithmic space?**
*Ground Truth Answers:* L, L, L

**Figure 1. Question Breakdown**




## MODEL ARCHITECTURE

**Figure 2. Model Diagram**




**Self Attention**

$$e^i_j = \mathbf{v}^T \tanh(\boldsymbol{W}_1 \boldsymbol{v}_j + \boldsymbol{W}_2 \boldsymbol{v}_i) \in \mathbb{R}$$

$$\alpha^i = \mathrm{softmax}(\boldsymbol{e}^i) \in \mathbb{R}^N$$

$$\boldsymbol{a}^i = \sum_{j=1}^{N} \alpha^i_j \boldsymbol{v}_j \in \mathbb{R}^\ell$$

$$\{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N\} = \mathrm{biRNN}(\{[\boldsymbol{v}_1; \boldsymbol{a}_1], \ldots, [\boldsymbol{v}_N; \boldsymbol{a}_N]\})$$

**Bidirectional Attention Flow**

$$\boldsymbol{S}_{ij} = \boldsymbol{w}^T_{\mathrm{sim}}[\boldsymbol{c}_i; \boldsymbol{q}_j; \boldsymbol{c}_i \circ \boldsymbol{q}_j] \in \mathbb{R}$$

$$\alpha^i = \mathrm{softmax}(\boldsymbol{S}_{i,:}) \in \mathbb{R}^M \quad \forall i \in \{1, \ldots, N\}$$

$$\boldsymbol{a}_i = \sum_{j=1}^{M} \alpha^i_j \boldsymbol{q}_j \in \mathbb{R}^{2h} \quad \forall i \in \{1, \ldots, N\}$$

$$\boldsymbol{m}_i = \max_j \boldsymbol{S}_{ij} \in \mathbb{R} \quad \forall i \in \{1, \ldots, N\}$$

$$\beta = \mathrm{softmax}(\boldsymbol{m}) \in \mathbb{R}^N$$

$$\boldsymbol{c}' = \sum_{i=1}^{N} \beta_i \boldsymbol{c}_i \in \mathbb{R}^{2h}$$

$$\boldsymbol{b}_i = [\boldsymbol{c}_i; \boldsymbol{a}_i; \boldsymbol{c}_i \circ \boldsymbol{a}_i; \boldsymbol{c}_i \circ \boldsymbol{c}'] \in \mathbb{R}^{8h} \quad \forall i \in \{1, \ldots, N\}$$

## RESULTS

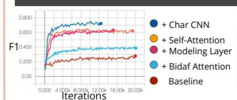



**Figure 3.** Plot showing our progression for dev set after adding different components to the baseline.

*Similarly, it is not known if L (the set of all problems that can be solved in logarithmic space) is strictly contained in P or equal to P. Again, there are many complexity classes between the two, such as NL and NC, and it is not known if they are distinct or equal classes.*
**What lies between L and P that prevents a definitive determination of the relationship between L and P?**
True Answer: *complexity classes* Predicted Answer: *complexity classes*

*In 2007 , the kenyan government unveiled vision 2030 , an economic development programme it hopes will put the country in the same league as the asian economic tigers by the year 2030 ...*
**What did kenya reveil in 2030?**
True answer: *vision 2030* Predicted answer: *asian economic tigers*

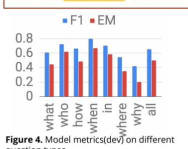



**Figure 4.** Model metrics(dev) on different question types.

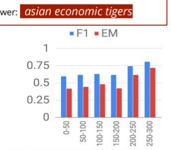



**Figure 5.** Model metrics(dev) by context length groups.

## CHAR CNN

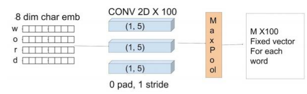



**Figure 6.** Char CNN to learn high dimensional char level word embeddings

## ABLATION

| | F1 | EM |
|---|---|---|
| Baseline | .26 | .19 |
| Baseline ++ | .31 | .21 |
| **Full(batch=100)** | **.78** | **.63** |

**Figure 7.** Performance on dev (batch size 100)

| | | |
|---|---|---|
| Full (batch=64) | 0.73 | 0.59 |
| Full + ELMo | .71 | .57 |
| Self Attention | .71 | 0.55 |
| Modeling Layer | .44 | .32 |
| Character CNN | .65 | .52 |
| BiDAF | .53 | .38 |

**Figure 8.**
Ablation Study with smaller batch size (64) than our final model on dev set to understand how removing certain components affected the final model.

- The modeling layer provides a significant performance boost
- We need to fix our ELMo integration since it is negatively affecting performance. It may be better to pass directly to the modeling layer
- Self attention adds only marginal benefit. Most likely due to the fact that the context only attends to itself.
- Character embeddings were more beneficial than we expected
- BiDAF was more beneficial for this task

## ANALYSIS

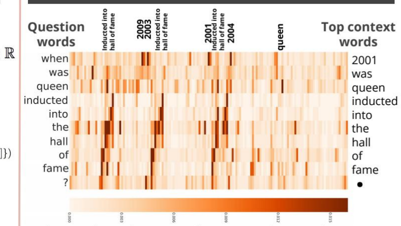



**Figure 9.** Bidirectional attention visualization using similarity matrix between question and context.

**Key takeaways of hyperparameter tuning experiments**

- We found that the dropout rate of 0.2 gives performance boost but overfits as seen in figure 7 and 8
- Model learns slower with learning rate of 0.0001 as seen in Figure 9
- Batch size of 100 gives better performance than 32 and 64

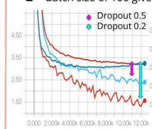



**Figure 10.** Loss vs iterations for different

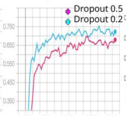



**Figure 11.** F1 vs iterations score for different dropouts (dev)

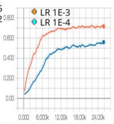



**Figure 12.** F1 vs iterations for different learning rates (dev)

## CONCLUSION

- The modeling layer is very important to the performance of this model
- The model needs more sophisticated language understanding to handle SQuAD 2.0 and abstractive answering.
- Character embeddings helped more than expected and make the model robust for out of vocabulary tokens.
- Attention is very powerful and efficient. RNNs are slow to train.

## FUTURE WORK

- Output layer that conditions the end prediction on the start prediction
- Highway Layers that are used in the BiDAF model
- Experimenting more in how to integrate ELMo and new transfer learning methods such as BERT
- Inputting more features (POS tags, NE tags, EM tags, TFIDF, etc.)
- Test different optimization methods
- Use non-RNN approach (CNNs, Transformer)

## REFERENCES


- Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603, 2016.
- Natural Language Computing Group, Microsoft Research Asia, R-NET: MACHINE READING COMPREHENSION WITH SELF-MATCHING NETWORKS
- CS224N Final Project Worksheet



ANUPRIT KALE anuprit@stanford.edu EDGAR VELASCO edgarv1@stanford.edu