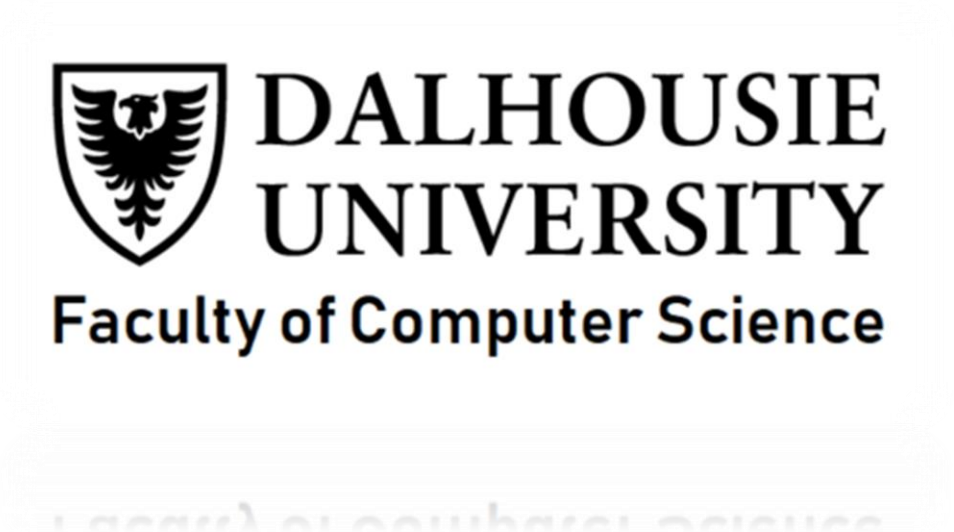

CSCI 6509 - Natural Language Processing

Project Report: Question-Answering on SQuAD Dataset

APRIL. 20th 2020

Disha Malik (B00842569)
Nikitha Pasya (B00830710)
Sneha Kotha (B00839933)

ABSTRACT:

Machine comprehension of text is one of the major applications in natural language processing, which requires modeling complex transactions between the context and query to find the accurate answer. The Stanford Question Answering Dataset (SQuAD) provides large number of questions and their solutions created by humans through crowdsourcing on a set of Wikipedia articles. SQuAD retains a diverse set of answers and requires different forms of logical reasoning, including multi sentence reasoning. In this project, we re-implemented Bidirectional Attention Flow model(BiDAF). BiDAF is a multi-stage hierarchical process which represents context at various levels of granularity. Bidirectional Attention flow mechanism is used to obtain a query-context representation. Through experimental evaluations, our best model has said to be achieved an F1 score of 75 and EM score of 62.5 on the test set.

INTRODUCTION:

QA is an NLP task that requires knowledge of natural languages. In this task, a model should give answers to questions presented in natural language based on given context. The context contains answer to the question. These problems require attention mechanism between context and questions. Machine comprehension and Question Answering have gained immense popularity over past few years. One of the major advancements has been use of neural attention mechanism, which enables system to focus on most relevant context in the paragraph to answer question.

Attention mechanisms in previous works commonly have one or more of the following characteristics. First, the computed attention weights are often used to extract the most relevant records from the context for answering the query through summarizing the context into a fixed-size vector. Second, in the text domain, they are frequently temporally dynamic, whereby the interest weights at the current time step are a characteristic of the attended vector at the preceding time step. Third, they are usually unidirectional, whereby the question attends on the context paragraph. The model is given a paragraph, and a corresponding question related to that paragraph as input. The goal is to provide the answer accurately. SQuAD contains about 107,785 QA pairs based upon 563 articles. The model should select answer from all possible spans in passage[5].

In BiDAF character, word, and contextual embeddings are included, and bi-directional attention flow is applied to obtain a query-to context representation. The attention layer is not used for summarizing the context into fixed-size vectors. The attention is computed at every time step, and the attended vector at each time step, along with the representations from previous layers, which then flow to the subsequent modeling layer. This lessens the information loss caused by early summarization. A memory-less attention mechanism is implemented. That is, while iteratively computing attention through time, the attention for each time step is a function of the query and the context paragraph at the current time step and does not directly depend on the attention of the previous time step[1].

An assumption is created that this simplification results in the division of labor between the eye layer and therefore the modeling layer. The eye layer focuses on learning the eye between query and context. This enables the modeling layer to target learning the interaction within the query-

aware context representation. Attention mechanisms are employed in both directions from query-to-context and context-to-query[4].

RELATED WORK:

Question Answering System uses concepts of machine learning, information retrieval and natural language processing, and a significant contribution for the advancement of Machine comprehension has been the availability of large datasets. SQuAD, consists of questions posed by crowd workers on a set of Wikipedia articles, where the answer to each question is a piece of text from corresponding passage. The data was collected in three stages: Curating passages, Crowdsourcing QA on those passages and Obtaining additional answers. Logistic regression model with a range of features, was implemented and its accuracy was compared with three baseline methods. Accuracy on this dataset was measured by F1 score and percent of exact matches through EM score [2].The task of question answering has been gaining a lot of interest in recent times.

Early works on visual question answering (VQA) involved encoding the question using an RNN, encoding the image employing a CNN and mixing them to answer the question [5][6]. Previous works in end-to-end machine comprehension used attention mechanisms. A dynamic attention mechanism, within which the eye weights are updated dynamically given the query and therefore the context further because the previous attention [7]. Simply using bilinear term for computing the eye weights within the same model drastically improves the accuracy [8]. Another mechanism includes reversing the direction of the eye (attending on query words because the context RNN progresses) for SQuAD[9].

Retrospective Reader distinguishes unanswerable questions so as to avoid giving plausible answers. Verification module, called verifier is employed, additionally with encoder, to spot unanswerable questions. There are two stages to spot whether the question is answerable or not, 1) sketchy reading, briefly touches the link of passage and questions, and 2) Intensive Reading, verify answer and provides final prediction. Model employs linear layer with SoftMax operations for final prediction. The implementation is predicated on BERT and ALBERT. The model uses the pre-trained LM weights in encoder module and available PLMs as encoder to make baseline MRC models: BERT and ALBERT. Two official metrics evaluates the model performance: Exact Match (EM) and a softer metric F1 score, which measures the weighted average of the precision and recall rate at a personality level. The model achieved 90% F1 score and 87% EM score on SQuAD dataset, outperforming the ALBERT baseline with simple threshold-based verifier, also achieves new state-of-the-art on the SQuAD [3].

Problem Definition

Existing end-to-end neural architectures for the Question Answering task either rely on the candidate answers or assume that the answer is a single token , which make these methods unsuitable for the SQuAD dataset[10]. To address this machine comprehension problem as defined in the SQuAD dataset, a new end-to-end neural architecture has been proposed in this project.

Methodology

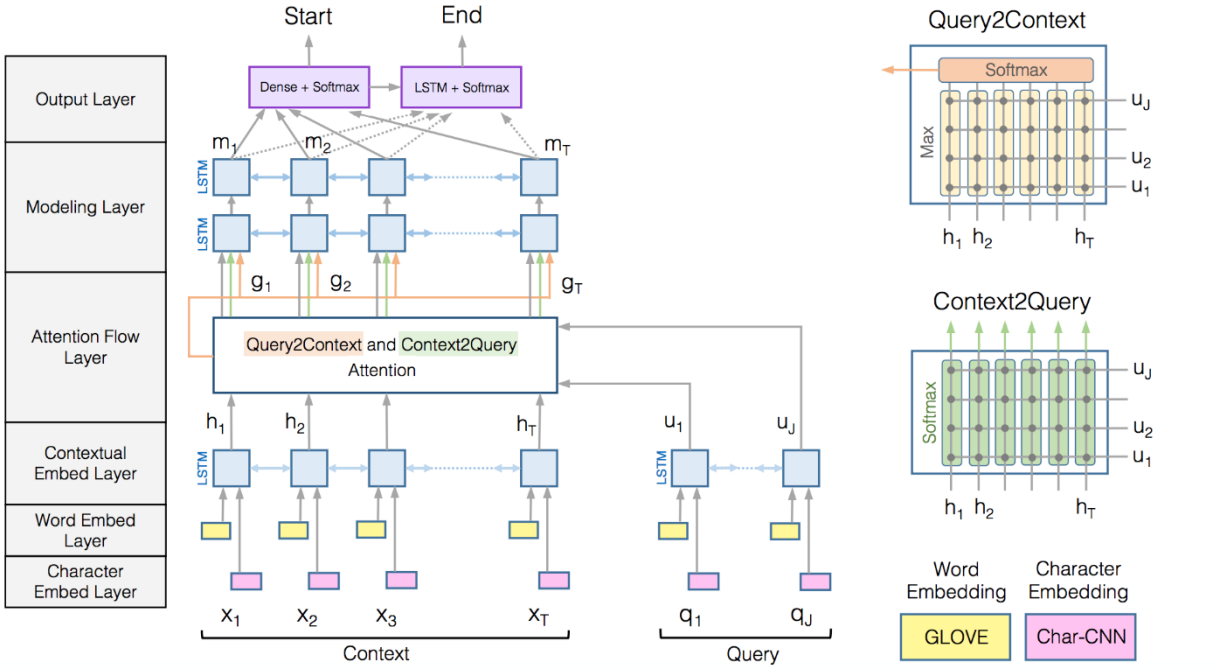

Figure 1:BiDAF with self-attention mechanism[4].

Character Embedding Layer: Character embedding layer is responsible for mapping each word to a vector. Let $\{ x_1, \ldots x_T \}$ and $\{ q_1, \ldots q_J \}$ represent the words in the input context paragraph and question, respectively. We obtain character level embedding of each word using Convolutional Neural Networks (CNN). The output of the CNN is max-pooled over entire width to obtain a fixed-size vector for each respective word[4].

Word Embedding Layer: Word embedding layer also maps each word to a vector. Pre-trained word vector GloVe is used to obtain the fixed word embedding for each word. The combination of the character and word embedding vectors is passed to a two-layer Highway network. The outputs of the Highway Network are two sequences of dimensional vectors, or more conveniently, two matrices: $X \in R^{dXT}$ for the context and $Q \in R^{dXJ}$ for the query[4].

Contextual Embedding Layer: Long Short-Term Memory Network (LSTM) is implemented on top of the embeddings provided by the previous layers to model the temporal interactions between words. We place an LSTM in both directions i.e., from left to right and right to left, and then combine the outputs of the two LSTMs. Thus, we obtain $H \in R^{2dXT}$ from the context word vectors X, and $U \in R^{2dXJ}$ from query word vectors Q. The first three layers of the model are computing features from the query and context at varying levels of granularity [4].

Attention Flow Layer: The attention vector at each time step, along with the embeddings from previous layers, are allowed through the subsequent modeling layer. The inputs to the layer are contextual vector representations of the query U and context H. The outputs of the layer is contextual embeddings from the previous layer along with the query vector representations of the

context words, G. In this layer, attentions are computed in two directions: from query to context and context to query .The similarity matrix is computed by S_{tj} = α($H_{:t}$, $U_{:j}$) ∈ R. where $H_{:t}$ is t-th column vector of H, and $U_{:j}$ is j-th column vector of U and α is a trainable scalar function that encodes the similarity between its two input vectors, . Similarity matrix S is used to obtain the attentions and the attended vectors in both directions.

Context-to-query Attention(C2Q): This attention specifies which of the query words are mostly relevant with respect to each context word. If $a_t \in R^J$ represents the attention weights on query words by t-th context word, $\sum a_{tj}$ = 1 for all t. Then attention weight is computed by a_t = softmax($S_{t:}$) ∈ R^J , and each attended query vector subsequently is $U_{:j}$= $\sum_j a_{tj}\ U_{:j}$. Hence U˜ is a 2d-by-T matrix that contains the attended query vectors for the entire context [4].

Query-to-context Attention(Q2C): This attention specifies which of the context words have closest similarity to one of the query words and are crucial in answering the query. The attention weights are obtained on the context words by b = softmax(max_{col}(S)) ∈ R^T.The attended context vector is h˜ =$\sum_t b_t\ H_{:t}$ ∈R^{2d}. This context vector indicates the weighted sum of the most important words in the context with respect to the query. h˜ is tiled T times across the column, thus giving H˜ ∈ R^{2dXT}. Finally, the contextual embeddings and the attention vectors are combined together to yield G. G is defined as $G_{:t}$ = β($H_{:t}$, $U\tilde{}_t$, $H\tilde{}_t$) ∈ R^{d_G}.where β is a trainable vector function that fuses its input vectors, $G_{:t}$ is the t-th column vector (corresponding to t-th context word), and d_G is the output dimension of the β function[4]

Modeling layer: The input for this layer is G and is responsible for encoding the query-aware representations of context words. The output layer captures the interaction between context words conditioned on the query. Implementation of two layers of bi-directional LSTM, with the output size of d for each direction. A matrix M ∈ R^{2dXT} is generated, which is passed onto the output layer to predict the answer [4].

Output Layer: BiDAF allows us to swap out the output layer based on the task. The QA task requires the model to find a context in the paragraph to answer the query. The context is derived by predicting the start and the end indices of the phrases in the paragraph. The probability distribution of the start index is given by p^1 = softmax($w_{p^1}^T$ [G;M]), where $w_{p^1}^T$ ∈ $R^{10d}$is a trainable weight vector. For the end index, M is passed to another bidirectional LSTM layer to obtain M^2 ∈ R^{2dXT} . M^2 is used to obtain the probability distribution of the end index in a similar manner: p^2 = softmax($w_{p^2}^T$ [G; M^2])[4].

Training: Training loss (to be minimized) is defined as the sum of negative log probabilities of true start and end indices by the predicted distributions, averaged over all examples:

$$L(\theta) = -\frac{1}{N} X \sum_i^N \log\left(p^1{}_{y_i^1}\right) + \log(p^2{}_{y_i^2})$$

where N is the number of examples in the dataset, θ is set of all trainable weights in the model y_i^1 and $y_i^2$are the true start and end indices of the i-th example, and p_k indicates the k-th value of the vector p [4].

Test: Answer span (k, l), where k ≤ l with the maximum value of $p_k^1 \; p_l^2$ is chosen, which can be computed in linear time with dynamic programming [4].

EXPERIMENTS

All the passages, questions and answers are tokenized. The resulting vocabulary contains 117K unique words. Word embeddings from GloVe are used to initialize the model. If words are not found in GloVe, the words are initialized as zero vectors. Word embeddings are not updated during the training of the model. The performance is measured using two metrics: percentage of the exact match with the ground truth answers, and word-level F1 score when comparing the tokens in the predicted answers with the tokens in the ground truth answers. In development set and the test set each question has around three ground-truth answers. F1 scores with the best matching answers are used in computing average F1 score.

Results:

Augmentation	F1 Score (one example)
Baseline	0.4
Bidirectional Attention	0.45
Modeling Layer	0.65
Basic Conditional Output Layer	0.67
BiDAF Output Layer	0.68
Activation and Hyperparameter Tuning	0.7
F1 (three examples)	0.75

This model was able to achieve an F1 score of 75 and EM of 62.5 on the three answer dev set, as well as an F1 of 0.7 on the one -answer dev set, and an F1 of 75.75 and an EM of 65.9 on test set. The two models have clearly outperformed the logistic regression model which relies on carefully designed features. Furthermore, this model has outperformed the sequence model.

Further Analysis:

To clearly understand the pros and cons of the models, a performance analysis is done and found the below results. Firstly, longer answers are harder to predict. To verify this assumption, the performance with respect to both exact match and F1 score with the answer length is analyzed. The performance of the model on different groups of questions is checked. By splitting the questions into different groups based on a set of question words that are defined, including “what,” “how,” “who,” “when,” “which,” “where,” and “why" infers to questions with different types of answers. If we consider the above words, “when” questions look for temporal expressions as the answer, whereas “where” questions look for locations as answers. According to the performance on the dev data set, these models work the best for “when” questions. This may be due to that

dataset's temporal expressions are relatively easier to identify. Other groups of questions whose answers are noun phrases, such as “what” questions, “which” questions and “where” questions, also gave relatively better results. On the other hand, “why” questions are observed to be hardest. This is obviously because the answers to “why” questions can be diverse, and they are not restricted to any certain type of phrases.

Conclusion

In this project, we re-implemented BIDAF, a multi-stage hierarchical process that represents the context at varying levels of granularity and uses a bi-directional attention flow mechanism to achieve a query aware context representation without early summarization. The experimental results show that the model achieves better results in Stanford Question Answering Dataset (SQuAD). The visualizations and discussions show that this model is learning a suitable representation for Machine Comprehension and can answer complex questions by attending to correct locations in the given paragraph.

References

[1] Chang, Allen. “BI-DIRECTIONAL ATTENTION FLOW FOR MACHINE COMPREHENSION.” *Medium*, Medium,2019.medium.com/@allenfrank14/paper-reading-bi-directional-attention-flow-for-machine-comprehension-f885e1284396.

[2] Rajpurkar, Pranav, Zhang, Jian, Konstantin, Liang, & Percy. 2016. SQuAD: 100,000 Questions for Machine Comprehension of Text. Retrieved from https://arxiv.org/abs/1606.05250

[3]Zhuosheng Z., Junjie Y.,Hai Z.2020.Retrospective Reader for Machine Reading Comprehension. Retrieved from https://arxiv.org/abs/2001.09694.

[4] Minjoon.S., Aniruddha K., Ali F., and Hananneh,H. 2018. Bi-directional attention flow for machine comprehension. Retrieved from https://arxiv.org/abs/1611.01603.

[5] Mateusz,M., Marcus,R., and Mario,F, 2015, Ask your neurons: A neural-based approach to answering questions about images.

[6] Stanislaw,A., Aishwarya,A., Jiasen,Lu., Margaret,M., Dhruv,B., C Lawrence,Z., and Devi, P., ICCV, 2015, Vqa: Visual question answering.

[7] Dzmitry Bahdanau, Kyunghyun,C., and Yoshua,B, ICLR, 2015, Neural machine translation by jointly learning to align and translate.

[8]Danqi,C., Jason,B. and Christopher,D. M.A, ACL, 2016, thorough examination of the cnn/daily mail reading comprehension task.

[9]Shuohang,W. and Jing,J, 2016, Machine comprehension using match-lstm and answer pointer. arXiv preprint arXiv:1608.07905.

[10] Yin, W., Ebert, S., & Schütze, H. (2016). Attention-Based Convolutional Neural Network for Machine Comprehension. Proceedings of the Workshop on Human-Computer Question Answering.