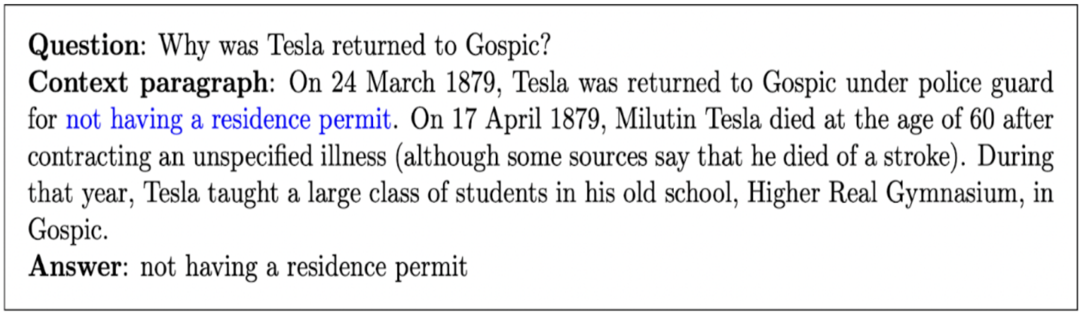
Generate the answer of a question from a given context

1 Introduction

Machine Reading Comprehension/ Question Answering is a task of Identifying the answers for a given set questions, given the context describing them. Dataset used is Stanford Question Answering Dataset(SQuAD) it contains for a context, a list of questions, for which the answer is a span of the context example of which as been given in the Figure 1.



Question: Why was Tesla returned to Gospic?
Context paragraph: On 24 March 1879, Tesla was returned to Gospic under police guard for not having a residence permit. On 17 April 1879, Milutin Tesla died at the age of 60 after contracting an unspecified illness (although some sources say that he died of a stroke). During that year, Tesla taught a large class of students in his old school, Higher Real Gymnasium, in Gospic.
Answer: not having a residence permit

Figure 1: Example from the dataset

2 Methods

In this Project, all the models uses BiDAF (Seo et al., 2016) as the baseline model. It Three Main Layers, Character Embedding Layer, Word Embedding Layer, Contextual Embedding Layer, Attention Flow Layer, Modeling Layer and the Output Layer. A brief overview of the model is shown in the figure 2

2.1 Character Embedding Layer

Used to map the word to a high-dimensional vector space, in this layer we obtain the character level embedding of each word using Convolutional Neural Networks (CNN). Characters are embedded into vectors, which can be considered as 1D inputs to the CNN, and whose size is the input channel size of the CNN. The outputs of the CNN are max-pooled over the entire width to obtain a fixed-size vector for each word. Used 8-D vector to represent each character, and convolution has 100 channels, so the out put of this layer is 300 dimension for each word in the sequence.

2.2 Word Embedding Layer

This layer also maps the word to a high dimensional space. Here pre-trained word vector GloVe(Pennington et al., 2014) is used. This is of 300 dimensional word vector.
The concatenation of the character and word embedding vectors is passed to a two-layer Highway Network (Srivastava et al., 2015). The purpose of this layer is to learn to pass relevant information from the input. A highway network is a series of feed-forward or linear layers with a gating

Preprint. Under review.

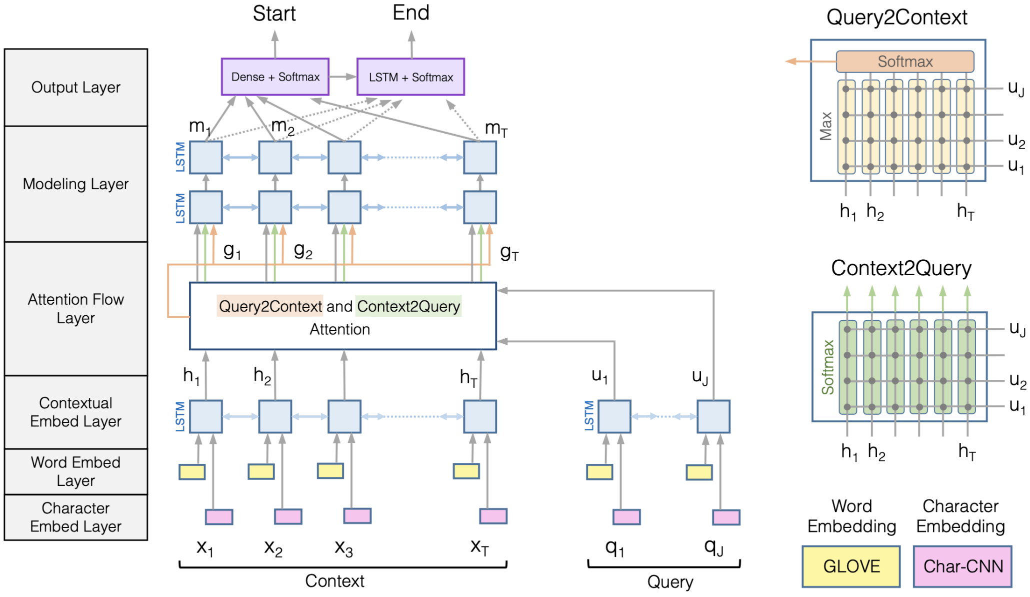


Figure 2: BiDirectional Attention Flow Model.

mechanism. The gating is implemented by using a sigmoid function which decides what amount of information should be transformed and what should be passed as it is.

2.3 Contextual Embedding Layer

Here a sequential model (LSTMs, RNNs, Transformers respectively) is used on top of the embeddings provided by the previous layers to model the temporal interactions between words. RNNs/LSTMs are placed in both directions, and concatenate the outputs of the two RNNs/LSTMs. Output ==> C

2.4 Attention Flow Layer

This layer is responsible for linking and fusing information from the context and the query words. Here the attention is calculated in both direction Context to Query as well as Query to Context.

Context-to-query Attention Context-to-query (C2Q) attention signifies which query words are most relevant to each context word

Query-to-context Attention Query-to-context (Q2C) attention signifies which context words have the closest similarity to one of the query words and are hence critical for answering the query. Here the layer output is $G = [C; C2Q; C \odot C2Q; C \odot Q2C]$ which called as query aware representation in the paper

2.5 Modeling Layer

The input to the modeling layer is G, which encodes the query aware representations of context words. The output of the modeling layer captures the interaction among the context words conditioned on the query. This is different from the contextual embedding layer, which captures the interaction among context words independent of the query. Here two layers of bi-directional Sequential Model is used. This Layer Output the matrix M. Each column vector of M is expected to contain contextual information about the word with respect to the entire context paragraph and the query.

Table 1: Performance Of BiDAF(LSTM) and QAFCNN for 3 epochs

Model	F1	EM
BiDAF(LSTM)	51.98	40.3
QAFCNN	17.84	13.23

2.6 Output Layer

Here p^1 an p^2 are produced by passing $[G; M]$ into a Fully connected layer and softmaxed. Where p^1 represents start index and p^2 represents end index of the answer span. Cross entropy loss of p^1 and p^2 are added to produce training loss.

2.7 Feed Forward NN Model (QAFCNN)

For Feed Forward model only Glove embedding is used and in modelling layer a three layered NN is used instead of fee forward layer where the input is G. Rest of the part is same as metioned above.

3 Evaluation Criteria

We used tne Metrics that's traditionally used in all types of Question Answering, Namely Exact Match and F1 score.

3.1 Exact Match

In Exact Match If two Strings exactly match than the Score of 1 is given to it else a score of 0 is given. This value is averaged for all the examples in the dataset to get the final Value.

3.2 F1

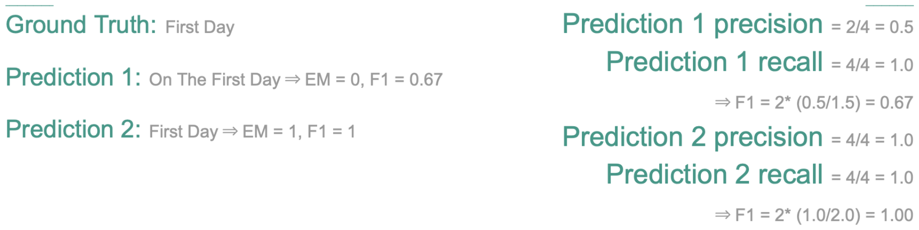
This is a more forgiving metric as it gives the score based on amount of overlap between the predicted string and the Golden truth string. Equation concerning this metric is mentioned below.

$$Precision = \frac{|(GoldenString) \cap (PredictedSting)|}{|(PredictedSting)|},$$

$$Recall = \frac{|(GoldenString) \cap (PredictedSting)|}{|(GoldenSting)|},$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

And the metric calculation for an example string is shown in figure 3.



Ground Truth: First Day

Prediction 1: On The First Day ⇒ EM = 0, F1 = 0.67

Prediction 2: First Day ⇒ EM = 1, F1 = 1

Prediction 1 precision = 2/4 = 0.5

Prediction 1 recall = 4/4 = 1.0

⇒ F1 = 2* (0.5/1.5) = 0.67

Prediction 2 precision = 4/4 = 1.0

Prediction 2 recall = 4/4 = 1.0

⇒ F1 = 2* (1.0/2.0) = 1.00

Figure 3: EM and F1 Calculation Example

4 Analysis of Results

Performance reported in the actual paper is significantly higher then the ones report here, but they ran it for 12 epochs supposed to only 3 epochs in this case. This result significantly better then once reported in original SQuAD Paper (Rajpurkar et al., 2016)

5 Discussions and Conclusion

Though to model performance is better than baseline models, Modern state of the art models perform significantly better than reported here. Future work could be of using pretrained contextual embeddings and use of more sophisticated and well designed Transformer models.

References

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.