

Rediscovering R-NET: An Improvement and In-Depth Analysis on SQuAD 2.0

Stanford CS224N Default Project

Jacob Wagner

Department of Computer Science
Stanford University
jtwagner@stanford.edu

Minju Kim

Department of Computer Science
Stanford University
minjukim@stanford.edu

Abstract

Question-answering is a discipline within the fields of information retrieval (IR) and natural language processing (NLP) that is concerned with building systems that automatically answer questions posed by humans. In this project, we address the question-answering task by attempting to improve the R-NET model [1]. Specifically, our goals are to 1. reproduce R-NET and evaluate its performance on SQuAD v2.0 and 2. change certain features of the R-NET model to further improve its accuracy on SQuAD v2.0. We present an implementation of R-NET using LSTM's instead of GRU's, larger embedding and hidden dimensions, higher dropout, and more layers that achieves an improvement in performance from our baseline R-NET model.

1 Key Information

- Mentor: Mandy Lu

2 Introduction

The proliferation of machine learning (ML) systems has effected a rise in demand for accurate question-answering systems. Question-answering is a supervised ML technique that requires three parts from an example: the question, the passage, and the answer. The system first reads the question and passage and then tries to discern the answer to the question using the information from the passage. The question-answering problem is an interesting challenge as search engines have risen to such a prominent scale in day-to-day life. Today, people turn to their favorite search engines to answer questions that run the gamut from simple math explanations to deep philosophical questions. These question-answering systems must be able to search through enormous amounts of information in order to extract only what is directly relevant to the user's query. The accuracy and relevance of these solutions also impact user satisfaction with search engine tools. ML systems are thus challenged to identify the pertinent information within a passage and effectively resolve the correct answer to the question.

With continued advancements in deep learning, researchers have been able to develop models that can outperform humans in terms of accuracy and speed. These models are fed massive bodies of text and are strapped with pretrained vector representations of words and characters. For our project, we want to investigate Microsoft's R-NET model and examine its effectiveness on the new SQuAD v2.0. Upon its release in 2017, R-NET was the state of the art, as it applied self-matching attention alongside a gated attention-based recurrent network [1]. These cutting edge contributions provided novel results to the Microsoft research team. Our goal in this paper is to re-implement R-NET, assess its performance on SQuAD v2.0 and set this model as our baseline, improve its predictive capabilities, and finally evaluate its successes and failures.

We will attempt to improve R-NET on SQuAD v2.0 in the following ways: switching out the GRU's for LSTM's, changing model size and number of layers, varying the size of hidden and embedding dimensions, and using Adam rather than Adadelta. We plan to then conduct an in-depth analysis on the effects of varying these different hyperparameters, and how the F1 and EM scores are impacted. We will also compare incorrect responses between our implementation of the original R-NET and our best performing model to better evaluate R-NET's behavior.

3 Related Work

The original R-NET model draws its key components from three areas of work: question-aware passage representation, gated networks, and self matching attention. We discuss relevant research in these areas that motivate our choice of improvements for R-NET.

3.1 Question Aware Passage Representations

Prior to the evolution of attention based models, many NLP systems struggled with longer inputs. These models could not maintain long range context and struggled to draw conclusions from the text with increasing time steps. The motivation behind attention thus began as a way to maintain long range dependencies between information and show relevance within instances of the data [2] [3] [4]. These attention mechanisms address the fact that words within the passage have varying levels of importance in relation to the posed question. This attention structure can also be used to create a question-aware passage representation of the textual data such that importance is given not based solely on passage context but question information as well [4]. Our team plans to implement more hidden layers and larger word embedding dimensions to obtain more nuanced representations of these question aware passages.

3.2 Gating Mechanism

The concept of a gated unit was initially introduced as a hidden unit that adaptively remembers and forgets [5]. Specifically, a *reset* gate decides whether the previous hidden state is ignored, while the *update* gate controls how much information from the previous hidden state carries over to the current hidden state. Recent research has shown that gated hidden units can be incorporated in RNN's as the activation function f , replacing the conventional *simple* units such as an element-wise *tanh* [2]. While R-NET uses a GRU to be more computationally efficient, we hope to see improvements by replacing this with an LSTM, as the latter has four gating units that control the information within it [6].

3.3 Self Matching Attention

Although question aware passage representation determines the importance of information within the passage based on the question, it ultimately loses the general context of the passage. To combat this, R-NET compares the question aware passage representation against the original passage, a concept called self-matching attention. Self-attention has become an increasingly valued tool in transformers, where it provides a reflective layer in which the model can regain positioning and context after conducting some level of encoding and abstraction away from the initial data [7] [8]. Our team hopes that increasing the number of layers within the encoders will help R-NET better identify context clues from self-attention.

4 Approach

Our primary approach was to first create our own implementation of the R-NET model and modify aspects of the original model (i.e. number of layers, GRU's v. LSTM's, etc.) so that we can improve its performance on SQuAD 2.0. The gated attention-based recurrent networks and self-matching attention features from the original R-NET paper will be the most critical components of our implementation. See Figure 1 in the Appendix for a high level overview of R-NET's architecture.

4.1 Question and Passage Encoder

We convert the question and passage to their respective word level embeddings by using nn.Embedding that utilizes the pre-trained word vectors that we received in the baseline code. We then use a linear transformation, $y = xA^T + b$, to get the embedding to the hidden layer dimensionality (64). After this linear transformation, we then encode both the question and passage with a bidirectional RNN.

Let e_t represent the embedding for a word at position t and u_t the encoding for a word at position t . Q represents all words in the question, and P all words in the passage.

$$\begin{aligned} u_t^Q &= BiRNN(u_{t-1}^Q, e_t^Q) \\ u_t^P &= BiRNN(u_{t-1}^P, e_t^P) \end{aligned}$$

4.2 Character Embeddings

In addition to word level embeddings, we implement character embeddings to help handle out-of-vocab (OOV) tokens. Given the 4D character embeddings from the starter code, we apply a 1D convolution over the sequence length and max pool over the char limit axis. We then concatenate the 3D character embeddings to the word embeddings, updating our encodings for all words in the question and passage, respectively, as such:

$$\begin{aligned} u_t^Q &= BiRNN(u_{t-1}^Q, [e_t^Q, c_t^Q]) \\ u_t^P &= BiRNN(u_{t-1}^P, [e_t^P, c_t^P]) \end{aligned}$$

4.3 Question Aware Passage Representation

Note: We used the default final project starter code BiDAFAttention layer [9] as a foundational starting point for our implementation of this functional operation in our code. Assume we have passage hidden states $p_1, \dots, p_n \in R^{2H}$ and question hidden states $q_1, \dots, q_n \in R^{2H}$ where H is the hidden vector length. We then use these to compute a similarity matrix $S \in R^{M \times N}$ which contains a similarity score S_{ij} for each possible pairing of question and passage hidden states p_i and q_j . S_{ij} appears as the following:

$$S_{ij} = w_{sim}^T (p_i \odot q_j)$$

Here $w_{sim}^T \in R^{6H}$ represents a weight vector. Note that because this is a similarity matrix we can use the columns in order to attend to the context.

Then we perform the question to context attention in order to create the encoding of question information within every word of the passage.

$$\begin{aligned} T_{:,j} &= \tanh(S_{:,j} \in R^N; j \in \{1, \dots, M\}) \\ U &= ST^T \in R^{N \times N} \\ c_i &= \sum_{j=1}^N U_{i,j} \cdot q_j \in R^{2H}; i \in \{1, \dots, N\} \end{aligned}$$

Here c_i represents the question aware passage representation for a particular word within the passage.

4.4 Gated Attention-Based Recurrent Networks

Following the R-NET structure, we created an attention layer to incorporate question information into the passage representation and then utilize a gate layer to determine the importance of specific portions of the passage. For the gated function, we utilize the encoding from the question aware passage representation which we will call c_t . For the gating layer, we update our encodings by using the following:

$$\begin{aligned} g_t &= \text{sigmoid}(W_g[u_t^P, c_t]) \\ [u_t^P, c_t]' &= g_t * [u_t^P, c_t] \end{aligned}$$

4.5 Self-Matching Attention

Once we have retrieved the question aware passage representation by using the gated attention based layering, we want to re-establish context of the passage while maintaining the importance hierarchy from the question aware passage representation. To do this, we will conduct a similar attention and gating, except it will be conducted with the question aware passage representation and the passage itself. Note that it is very similar to the "Question Aware Passage Representation" section.

$$\begin{aligned} A_{ij} &= w_{sim}^T(c_i \odot p_j) \\ B_{:,j} &= \tanh(A_{:,j} \in R^N); j \in \{1, \dots, M\} \\ F &= AB^T \in R^{N \times N} \\ g_i &= \sum_{j=1}^N F_{i,j} \cdot p_j \in R^{2H}; i \in \{1, \dots, N\} \end{aligned}$$

Here, g_i represents the question aware passage representation that now includes greater context awareness.

4.6 Output Layer

Note: We used the default final project starter code BiDAFOutput layer [9] as a foundational starting point for our implementation of this functional operation in our code. The output layer is tasked with determining a vector of probabilities that correspond to every word within the passage. Our output layer finds the probability p_{start} and $p_{end} \in R^N$. $p_{start}(i)$ and $p_{end}(i)$ represent the probability that the spanning solution starts at position i or ends at position i .

To do this, we begin by taking the question aware passage representation with greater context awareness as described in the previous subsection and apply a bidirectional GRU.

$$\begin{aligned} g'_{i,rev} &= GRU(g'_{i+1}, g_i) \in R^H \\ g'_i &= \exp(g'_i) / \sum_{j=1}^n \exp(g'_j) \\ g_{i,fwd} &= GRU(g'_{i-1}, g_i) \in R^H \\ g_i &= \exp(g_i) / \sum_{j=1}^n \exp(g_j) \end{aligned}$$

After applying this encoding, we use a tanh function to derive probabilities. W_{start} and W_{end} are both learned values here. G and G' represent the output from the encoded attention. M represents the last hidden state from the question aware passage representation with context awareness.

$$\begin{aligned} p_{start} &= \tanh(W_{start}[M; G]) \\ p_{end} &= \tanh(W_{end}[M; G']) \end{aligned}$$

Also note that there was an out-of-vocab token used at the beginning of the passage to include the possibility for no answer present within the passage. This token acted as a catch-all for the case where the probabilities of p_{start} and p_{end} were not high enough to justify a solution prediction.

5 Experiments

5.1 Data

We train our model on approximately 129,941 examples from the SQuAD v2.0 dataset [10]. Our dev set is approximately half (6078 examples) of the official SQuAD v2.0 dev set, and our test set is the rest of the official dev set (5915 examples). The data consists of (context, question, answer) triples. Context is drawn from Wikipedia articles and the correct answer is either "N/A" or a span of text in the context. Questions can be couched as "What", "Who", "How", "When", "Where", "Which", "Why", etc. For each answerable question, the official dev set also provides three possible answers. We use our custom dev set to tune hyperparameters and ascertain a sense of our model's performance.

5.2 Evaluation method

Our team uses both the exact match (EM) and F1 scores to evaluate our model. EM is a binary measure of whether the model's prediction matches the actual answer exactly. Due to EM's strict

standards, we also use F1, a metric that computes the average of the model’s precision and recall. For the examples in our dev and test sets, only the maximum EM and F1 scores across all three provided answers are taken into consideration. The final reported scores are the average of all final EM and F1 scores across the dataset.

5.3 Experimental details

Our team worked through three separate experimental trials. The first was a baseline that most similarly resembles R-NET, especially with regard to the hyperparameters (hidden layer size, hidden vector length, number of layers, optimizer, dropout rate, etc.). We then ran a second trial with a larger neural network with larger **embedding sizes**, **hidden layer** sizes, **hidden vector** length, and **number of layers**. From the second experiment onwards, we switched to **LSTM’s** over GRU’s to evaluate results. For our third trial, we experimented with an even larger model and adjusted the optimizer from AdaDelta to **Adam**. We will detail each of these trials separately within this section and give a further analysis of the results of each in section 5.4.

5.3.1 First Trial: R-NET (Baseline)

For our first experiment, we tried to model the R-NET architecture as closely as possible. The R-NET model specifically used a question aware passage representation that relied on attention [11]. It also had hidden layers of size **75** and hidden vector lengths of size **75**. The model was optimized using AdaDelta [12]. The ρ , ϵ , and learning rates are 0.95, e^{-6} , and 1, respectively. The authors of R-NET used a 3-layer bidirectional GRU to encode questions and passages, which we implemented. The baseline model took approximately 22 hours to run to completion.

5.3.2 Second Trial: R-NET with LSTM’s and Larger Model Size

After implementing the baseline, our team increased the hidden layer size and hidden vector length to **128** to test if a larger model would enable a more nuanced representation that in turn would allow for better reading comprehension. We also increased the original 3 layers [1] for the encodings to 5 in the hopes of creating a more direct understanding of the context. We hypothesized that achieving this would allow the model to more acutely identify context clues from the self attention to ultimately make better predictions. During this trial phase we also changed our use of GRU’s to LSTM’s. Learning rate was 0.5. These changes lengthened training time to over 30 hours, a near 50 percent increase from the baseline implementation.

5.3.3 Third Trial: R-NET with LSTM’s and Adam

For the final trial, we wanted to test if an even larger model than the previous trial would give us bigger improvements over the baseline. To do this, we increased the hidden vector length and hidden layer size to **256**. We also replaced the AdaDelta optimizer to Adam. Number of epochs was increased to 40, dropout rate was increased from 0.2 to 0.25 to avoid overfitting, and learning rate was decreased from .5 to 0.25. Total training time for our final trial was around 45 hours.

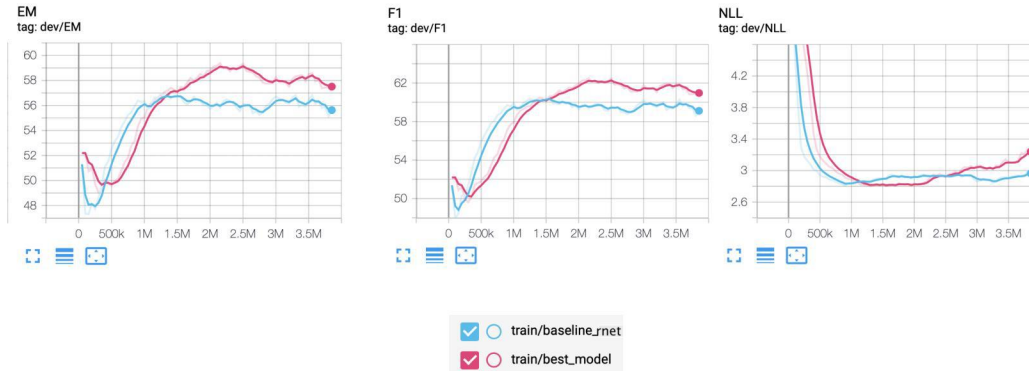
5.4 Results

Our quantitative evaluations on the dev and test sets are as follows:

Model	EM (dev)	F1 (dev)	EM (test)	F1(test)
BiDAF	56.83	60.34	-	-
R-NET (Baseline)	56.56	60.35	-	-
R-NET with LSTM’s and Larger Model Size	59.18	62.34	57.008	60.565
R-NET with LSTM’s and Adam	52.19	52.19	47.963	47.963

We attribute the slight difference in EM/F1 scores between the dev and test datasets to slight differences in data distribution. The R-NET with LSTM’s and Adam performed far worse than our baseline despite retaining similar architecture as our best performing model with larger hidden layers; we thus believe Adam may have been suboptimal for R-NET.

5.4.1 Best Performance Against Baseline



Our best performance was achieved with our second model (5.3.2), the R-NET with LSTM's and larger model size. As shown in the EM and F1 plots for our baseline rnet and best model, the latter consistently outperformed the baseline from around 1.5 million steps onwards. Performance gaps between EM tended to be larger than those between F1 scores.

Our improvements were expected for various reasons. Most notably, we believe that using LSTM's over GRU's yielded better performance and conclude that the former is preferred for larger datasets due to having more gates, despite being more computationally expensive. Additionally, we acknowledge that in general, enlarging the model increases its descriptive power, as more parameters are learned and a more nuanced understanding of question and passage relations is obtained. Overall, our approach of replacing GRU's with LSTM's, increasing hidden layer sizes and embedding dimensions, and increasing encoding layers were shown to improve R-NET's performance on SQuAD 2.0.

6 Analysis

For our team's analysis, we describe common threads behind our models' successes and failures. We often found that R-NET was better at answering questions asking "What" and "When" than those asking "Why" and "How." We also provide examples of our best model's capability to better maintain long range dependencies over the baseline model.

6.1 What and When Questions

Question:	What is the Norman architecture idiom?
Passage:	Norman architecture typically stands out as a new stage in the architectural history of the regions they subdued. They spread a unique Romanesque idiom to England and Italy, and the encastellation of these regions with keeps in their north French style fundamentally altered the military landscape. Their style was characterised by rounded arches, particularly over windows and doorways, and massive proportions.
Answer:	Romanesque
Prediction:	Romanesque

In the above example we found that both our improved model and the baseline were able to correctly identify the idiom for Norman architecture (Romanesque). We observed that these forms of identification questions were more commonly answered correctly by both our improved model and the baseline. We believe this is due to R-NET's tendency to look for semantically similar phrases. In this example, we see that the word "idiom" comes directly after the answer. Our team posits that these identification types of questions ("What" and "When") were easier for both our best model and the baseline because they require the least amount of interpretation and can rely heavily on context clues to determine the correct answer.

6.2 How and Why Questions

Question:	How unsuccessful was initial effort by Braddock?
Passage:	In 1755, six colonial governors in North America met with General Edward Braddock, the newly arrived British Army commander, and planned a four-way attack on the French. None succeeded and the main effort by Braddock was a disaster; he was defeated in the Battle of the Monongahela on July 9, 1755 and died a few days later. British operations in 1755, 1756 and 1757 in the frontier areas of Pennsylvania and New York all failed, due to a combination of poor management, internal divisions, and effective Canadian scouts, French regular forces, and Indian warrior allies. In 1755, the British captured Fort Beauséjour on the border separating Nova Scotia from Acadia; soon afterward they ordered the expulsion of the Acadians. Orders for the deportation were given by William Shirley, Commander-in-Chief, North America, without direction from Great Britain. The Acadians, both those captured in arms and those who had sworn the loyalty oath to His Britannic Majesty, were expelled. Native Americans were likewise driven off their land to make way for settlers from New England.
Answer:	N/A
Baseline Prediction:	a few days later
Best Model Prediction	a disaster

In this example, both the baseline and our model failed to realize that the question had no solution within the passage. Both models instead tried to use context clues to identify potential solutions, without actually understanding the question and passage. Here we see that the baseline predicted "a few days later," which comes directly after the mention of Braddock's unsuccessful operation and following death. The baseline was unable to understand that the "How" portion of the question is searching for an explanation of this unsuccessful effort. Similarly, our model predicted "a disaster" which simply falls within the same sentence as "succeeded", "effort", and "Braddock" even though this predicted answer would not be an effective response. Both of these models are quite capable in straightforward identification of solutions, but when prompted with more interpretive "Why" and "How" questions, the models struggle to find implications of statements. They instead search for phrase and word similarity to that of the prompt.

6.3 Long Range Dependencies

Question:	Where are America's oldest collection of maps, gazettes, and atlases housed?
Passage:	The Harvard University Library System is centered in Widener Library in Harvard Yard and comprises nearly 80 individual libraries holding over 18 million volumes. According to the American Library Association, this makes it the largest academic library in the United States, and one of the largest in the world. Cabot Science Library, Lamont Library, and Widener Library are three of the most popular libraries for undergraduates to use, with easy access and central locations. There are rare books, manuscripts and other special collections throughout Harvard's libraries; Houghton Library, the Arthur and Elizabeth Schlesinger Library on the History of Women in America, and the Harvard University Archives consist principally of rare and unique materials. America's oldest collection of maps, gazetteers, and atlases both old and new is stored in Pusey Library and open to the public. The largest collection of East-Asian language material outside of East Asia is held in the Harvard-Yenching Library.
Answer:	Pusey Library
Baseline Prediction:	N/A
Improved Model Prediction	Pusey Library

With this example, we can see how the baseline prediction did not identify a viable solution. This shows that the baseline wasn't able to derive Pusey Library even when given the similar phrasing of the question being placed in the passage. Our team believes the use of "old and new" just before "Pusey Library" deterred the model from predicting correctly because the query asks only about the

oldest collections. We also believe that the words "housed" (from the question) and "stored" (from the passage) detracted from the baseline's predictive capabilities. Meanwhile, our larger, improved model was able to identify the similar span of text. Despite the lengths of the question and sentence, our model was successful in determining this reference and properly predicting "Pusey Library".

7 Conclusion

Our team confirmed that the original R-NET was quite capable of correctly identifying answers within the passage to the posed prompt. This held true even with the possibility of having no answer present in the passage. Through updates and improvements to the R-NET framework, we were able to increase the length of long range dependencies but identified that our model performed poorly when tested with more complex, interpretive questions phrased as "Why" and "How." Future work should focus on examining how different neural structures perform on these more inference-based questions and how R-NET's architecture may be modified to improve its own performance on them as well.

References

- [1] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, July 2017.
- [2] Dzmitry Bahdanau and KyungHyun Cho. Neural machine translation by jointly learning to align and translate. In *ICLR*, July 2015.
- [3] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. Reasoning about entailment with neural attention, 2016.
- [4] Shuohang Wang and Jing Jiang. Learning natural language inference with lstm, 2016.
- [5] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.
- [6] Alex Graves. Supervised sequence labelling with recurrent neural networks, 2012.
- [7] Ju He, Jieneng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, Changhu Wang, and Alan Yuille. Transfg: A transformer architecture for fine-grained recognition, 2021.
- [8] Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Pretrained transformers as universal computation engines, 2021.
- [9] Chris Manning. Cs 224n default final project: Building a qa system (iid squad track). 2021.
- [10] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.
- [11] Natural Language Computing Group. R-net: Machine reading comprehension with self-matching networks. May 2017.
- [12] Matthew D. Zeiler. Adadelta: An adaptive learning rate method, 2012.

A Appendix (optional)

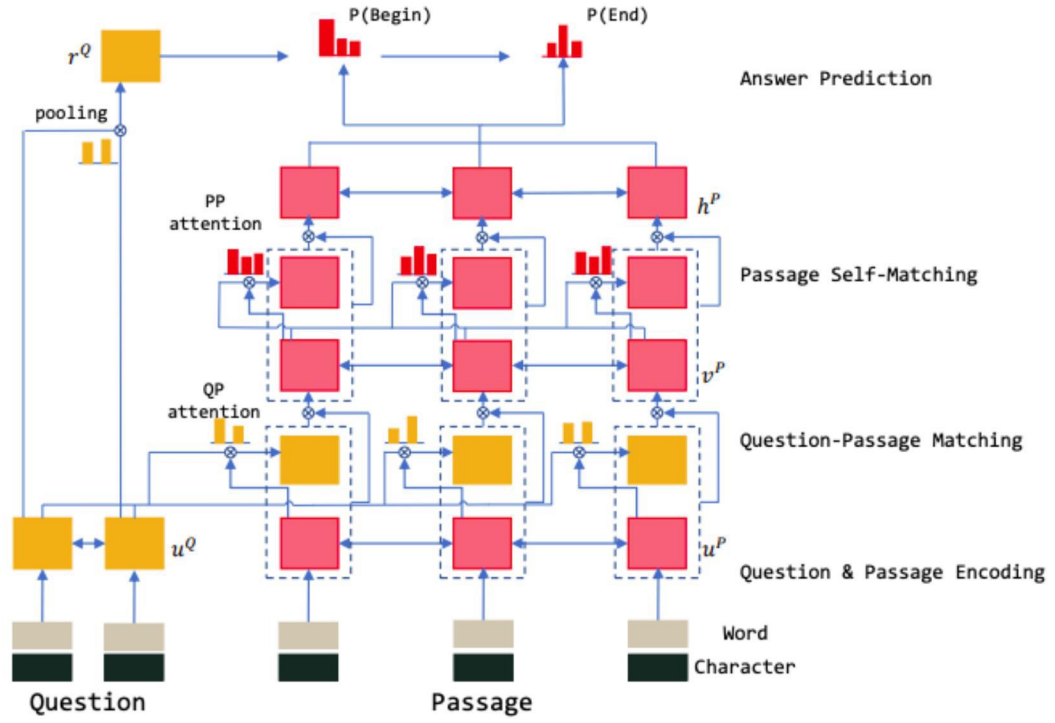


Figure 1: An overview of the architecture of R-NET [11]