A new model for Machine Comprehension via multi-perspective context matching and bidirectional attention flow

Amirata Ghorbani
Department of Electrical Engineering
Stanford University
amiratag@stanford.edu

Nima Hamidi
Department of Statistics
Stanford University
hamidi@stanford.edu

Abstract

To answer a question about a context paragraph, there needs to be a complex model for interactions between these two. Previous Machine Comprehension (MC) where either not large enough to train end-to-end deep neural networks, or not hard to learn. Recently, after the release of SQuAD dataset dataset, several adept models have been proposed for the task of MC. In this work we try to combine the ideas of two state-of-the-art models (BiDAF and MPCM) with our new ideas to obtain a new model for question answering task. Promising experimental results on the test set of SQuAD encourages us to continue working on the proposed model.

1 Introduction

The task of machine comprehension is to enable a machine understand a given block of text and then answer a question related to that text.

To address the comprehension problem, several large data sets have been developed by researchers. RCTest [1] has been developed for the task of answer selection from a small set of alternatives defined by annotators. It consists of 500 fictional stories and four multiple choice questions per story and has 2000 questions in total. Despite the proposed MC methods based on this dataset, the limited size of the dataset prevents building deep neural network models. To alleviate the scarcity of supervised data, Hermann et al. [2] developed a dataset of millions of Cloze style MC examples through semi-automated techniques from news articles of CNN and Daily Mail. It uses the bullet point summarization of each news article and constructs a triple of paragraph, question, and answer by replacing one entity in the bullet points with placeholders. The machine's task then would become filling the placeholder in the question with an entity in the paragraph. Performing hand analysis in [3], however, concluded that the dataset is not challenging enough to assess the present-day MC techniques.

Recently, Rajpurkar et al. [4] developed Stanford Question Answering Dataset (SQuAD) that first of all has no constraints on the set of allowed answers, other than they should be in the paragraph, and, Secondly, it is almost two orders of magnitude larger than previous datasets.

In this work, we focus on the SQuAD dataset and propose introduce our model, an end-to-end deep neural network model. In this work we have the assumption that the answer of the question is part of the paragraph that has the most similarity with the question. The model introduced in this work, matches the context of the paragraph with the question in several stages of the model. We have also assumed the answer to have a beginning and ending in the paragraph context, therefore, our model predicts the beginning and ending points of the

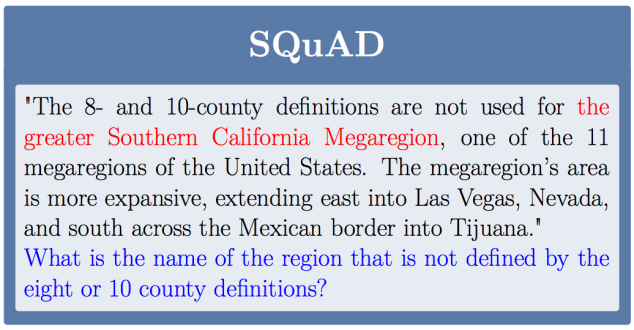


Figure 1: An example of SQuAD dataset

answer. We have experimented our model on the test set of SQuAD and achieved reasonable performance.

In what follows, we start with a brief literature review (Section2), followed by defining the MC task(Section 3). Then we discuss the details of our model(Section 4) and then evaluate our model (Section 5).

2 Related Work

The original paper from Rajpurkar et al. [4] adopted a sparse featured linear model based on the part-of-speech tags and n-grams in the question and the candidate part of the paragraph. Syntactic information was used in the form of the dependency paths in order that more general features would be extracted.

Weston et al. [5] in Memory Networks repeated computing the attention vector between question and paragraph through multiple layer; referred to as multi-hop. Shen et al. [6] used a combination of Memory Networks with Reinforcement Learning to control the number of hops dynamically. Wang et al [12] designed a multi-perspective context matching assuming the answer span of the paragraph being similar to the question.

Kadlec et al. [7] fed the a final prediction output layer with attention weights computed once. Cui et al. [8] used a similarity matrix between the question and paragraph tokens to compute the question to paragraph attention.

Bahdanau et al. [9] uses updates the attention weights dynamically given the question, the paragraph, and the previous attention. Hermann et al. [2] discussed the performance advantage of dynamic attention model over the models that use a fixed question vector to attend on paragraph tokens. Minjoon et al [?] in BiDAF used a memory-less paragraph-to-question and question-to-paragraph attention mechanisms.

3 Task Definition

Machine Comprehension involves a paragraph, a question from the paragraph, and the span of the answer inside the paragraph. (Figure 1) The MC model first comprehends the question, then compares tokens in the paragraph then identifies the answer span in the paragraph. The SQuAD dataset is a set of (P,Q,A) tuples; where $P = (p_1, \ldots, p_N)$ is an N-token-long paragraph, $Q = (q_1, \ldots, q_M)$ is an M-token-long question, and $A = (a_s, a_e)$ is the span of answer in the paragraph where a_s and a_e are the start and end points. In this work, we define the MC task as learning $P(A|P,Q)$. Then the answer span would be defined as $\arg\max_{A \in \mathcal{A}(p)} P(A|P,Q)$ where $\mathcal{A}(p)$ is the set of all possible answer span pairs in the paragraph. In this work, however, we have simplified the prediction as follows:

$$\arg\max_{1 \leq a_s \leq a_e \leq N} P(a_s|P,Q)P(a_e|P,Q)$$

4 Model Description

We propose a model architecture to estimate $P(a_s|P,Q)$ and $P(a_e|P,Q)$. As you observe, except the last layer, the $P(a_s|P,Q)$ and $P(a_e|P,Q)$ predictions are done by the same network with the following layers as depicted in Figure2:

- **Layer one: Word Embedding Layer** This layer maps each word of the paragraph and the question to a high-dimensional vector space. In this work, we have used the GloVe [11] word vectors. The output of this layer would be $P = [\mathbf{p}_1, ..., \mathbf{p}_N] \in \mathbb{R}^{d\times N}$ and $Q = [\mathbf{q}_1, ..., \mathbf{q}_M] \in \mathbb{R}^{d\times M}$.
- **Layer two : Paragraph Filtering Layer** The number of tokens in the paragraph is always bigger than the number of tokens in the question. In the general case, a large number of tokens in paragraph are not related to the answer span; the filter layer tries to alleviate the effect of those token. The method used here is inspired by [10]. First of all, make a Relevancy Matrix $R \in \mathbb{R}^{M\times N}$ between paragraph and character tokens is defined:

  $$r_{i,j} = cos(\mathbf{q}_i, \mathbf{p}_j)$$

  here *cos* is the cosine similarity function. Then for each word in paragraph we have:

  $$\forall j : P_j = (\max_i r_{i,j})P_j$$

  Therefore the irrelevant words of paragraph would be alleviated. In other words, if there is not even one similar token in question for a token in paragraph, it's effect would be alleviated.
- **Layer three: Contextual Embedding layer** In order to model the temporal interactions between words in the representation of each token of the paragraph and the question, we utilized a BLSTM to encode embeddings for each token in the question. Then, conditioned on a trainable scale of the last state vector of the question BiLSTM, we used the same BiLSTM on the paragraph tokens. Therefore, for $i = 1, \ldots, M$:

  $$\overrightarrow{h_i^Q} = \overrightarrow{LSTM}(\mathbf{h}_{i-1}^Q, \mathbf{q}_i)$$

  $$\overleftarrow{h_i^Q} = \overleftarrow{LSTM}(\mathbf{h}_{i+1}^Q, \mathbf{q}_i)$$

  Therefore, for $i = 1, \ldots, M$:

  $$\overrightarrow{h_i^Q} = \overrightarrow{LSTM}(\mathbf{h}_{i-1}^Q, \mathbf{q}_i)$$

  $$\overleftarrow{h_i^Q} = \overleftarrow{LSTM}(\mathbf{h}_{i+1}^Q, \mathbf{q}_i)$$

  and for $j = 1, \ldots, N$,:

  $$\overrightarrow{h_i^P} = \overrightarrow{LSTM}(\mathbf{h}_{i-1}^P, \mathbf{q}_i)$$

  $$\overleftarrow{h_i^P} = \overleftarrow{LSTM}(\mathbf{h}_{i+1}^P, \mathbf{q}_i)$$

  while $\overrightarrow{h_0^P} = k_{init} \times \overrightarrow{h_M^Q}$ and $\overleftarrow{h_{N+1}^P} = k_{end} \times \overleftarrow{h_1^Q}$. where k_{init} and k_{end} are constants to be trained.

  $$\overrightarrow{H}^P = \overrightarrow{LSTM}(P) \in \mathbb{R}^{h\times N}$$

  $$\overrightarrow{H}^Q = \overrightarrow{LSTM}(Q) \in \mathbb{R}^{h\times M}$$

  $$\overleftarrow{H}^P = \overleftarrow{LSTM}(P) \in \mathbb{R}^{h\times N}$$

  $$\overleftarrow{H}^Q = \overleftarrow{LSTM}(Q) \in \mathbb{R}^{h\times M}$$

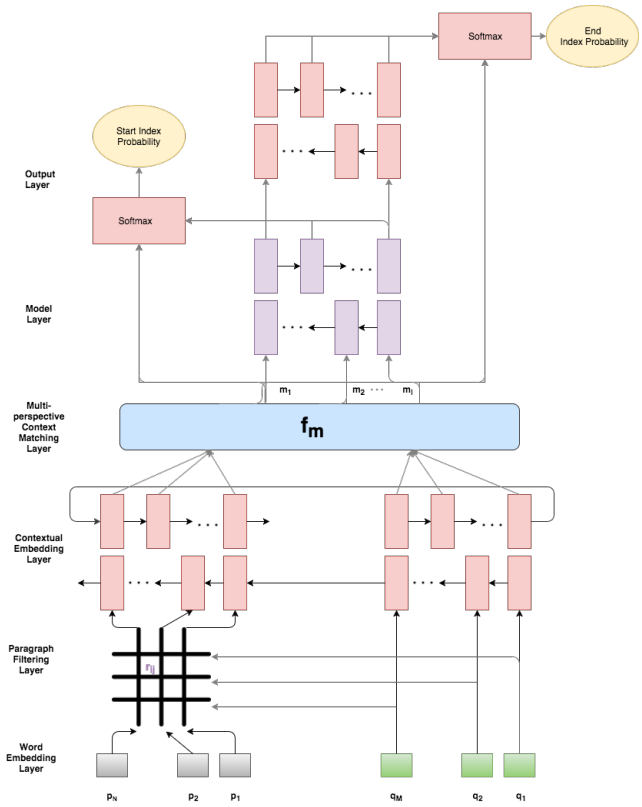


Figure 2: Network Model

- **Layer four: Multi-perspective context matching layer** Inspired by Wang et al [12], with multi-perspectives, each contextual embedding is compared with each question embedding. The perspective, are going to be learned in the training stage. First, consider the following operation that given two d dimensional vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ and a matrix W, returns an l dimensional multi-perspective matching vector of the two vectors:

$$\mathbf{m} = [m_1, ..., m_l] = f_m(\mathbf{v}_1, \mathbf{v}_2; W) \rightarrow m_k = cos(W_{k,:} \circ \mathbf{v}_1, W_{k,:} \circ \mathbf{v}_2)$$

where *cos* is the cosine similarity. Now for each word $\mathbf{p}_j$ in the paragraph we are going to design a 6 vectors as follows:

  - Full-matching vectors:

$$\overrightarrow{\mathbf{m}}_j^{full} = f_m(\overrightarrow{h_j^P}, \overrightarrow{h_M^Q}; W^1)$$

$$\overleftarrow{\mathbf{m}}_j^{full} = f_m(\overleftarrow{h_j^P}, \overleftarrow{h_1^Q}; W^2)$$

  - maxpooling matching vectors:

$$\overrightarrow{\mathbf{m}}_j^{max} = \max_i f_m(\overrightarrow{h_j^P}, \overrightarrow{h_i^Q}; W^3)$$

$$\overleftarrow{\mathbf{m}}_j^{max} = \max_i f_m(\overleftarrow{h_j^P}, \overleftarrow{h_i^Q}; W^4)$$

  - meanpooling matching vectors:

$$\overrightarrow{\mathbf{m}}_j^{mean} = \frac{1}{N} \sum_{i=1}^{N} f_m(\overrightarrow{h_j^P}, \overrightarrow{h_i^Q}; W^5)$$

$$\overleftarrow{\mathbf{m}_j}^{mean} = \frac{1}{N}\sum_{i=1}^{N} f_m(\overleftarrow{h_j^P}, \overleftarrow{h_i^Q}; W^6)$$

where $W^1, W^2, W^3, W^4, W^5, W^6 \in \mathbb{R}^{l\times h}$ are trainable.

these vectors are then concatenated to give a matching vector for each token in the paragraph.

$$M = [\mathbf{m}_1, \ldots, \mathbf{m}_N] \in \mathbb{R}^{6l\times N}$$

$$\text{for } j = 1, \ldots, N \quad \mathbf{m}_j = [\overrightarrow{\mathbf{m}_j}^{full}, \overleftarrow{\mathbf{m}_j}^{full}, \overrightarrow{\mathbf{m}_j}^{max}, \overleftarrow{\mathbf{m}_j}^{max}, \overrightarrow{\mathbf{m}_j}^{mean}, \overleftarrow{\mathbf{m}_j}^{mean}] \in \mathbb{R}^{6l}$$

- **Layer five: Modeling layer** In order to model the temporal interactions between matching vectors of each token in the paragraph with its surrounding tokens, we encode the matching vectors with a BiLSTM. to get $\mathbf{f}_i \in \mathcal{R}^{h_f}$s for $i = 1, \ldots, N$.

$$F = BiLSTM(M) \in \mathbb{R}^{2h_f\times N}$$

- **Layer six: output layer** Inspired from Seo et al 2017 [13], the output layer is application specific. The task defined in this work asks for a span of the paragraph as the answer. The probability distribution of the start index over the while paragraph is derived by :

$$\mathbf{p}^{\text{start}} = \text{softmax}(\mathbf{w}_1^T[M; F])$$

where $\mathbf{w}_1^T \in \mathbb{R}^{2h_f+6l}$ is a trainable weight vector. We pass $F \in \mathbb{R}^{2h_f\times N}$ through another BiLSTM layer to get $F^2 \in \mathbb{R}^{2h_f\times N}$. Then the end index probability is computed as follows:

$$\mathbf{p}^{\text{end}} = \text{softmax}(\mathbf{w}_2^T[M; F^2])$$

where $\mathbf{w}_1^T \in \mathbb{R}^{2h_f+6l}$ is a trainable weight vector. Then, for the start and end index we have;

$$a_s, a_e = \arg\max_{i=1,\ldots,end} \arg\max_{j=i,\ldots,N} (p_i^{\text{start}} \times p_j^{\text{end}})$$

- **Training** We minimize the sum of the cross entropies for start index and end index averaged over examples:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} (\log(\mathbf{p}_{a_{s_i}}^{\text{start}}) + \log(\mathbf{p}_{a_{e_i}}^{\text{end}}))$$

and θ is the set of all trainable parameters.

- **Model Complexity** As discussed, the model has two constants, two vectors, six Matrices and four BiLSTM layers to learn resulting in a total of 753,402 parameters.

5 Experiments

5.1 Implementation Details

The corpus was processed using NLTK tokenizer. We used the $d = 100$ dimensional GloVe word embeddings and for the words not in the dictionary random word embeddings were used. All hidden state sizes (h, h_f) were 100 and the multi-perspective matching vectors' size was $l = 50$. We didn't apply dropout in any layer. The optimizer used for minimizing the loss function was AdaDelta [14] with a batch size of 20, initial learning rate of 0.5, $\rho = 0.95$, and $= 10^{-6}$ and was ran for 10 epochs. In Figure 3 the per batch loss over simulation time(the number of batches fed) is depicted. The model was trained and implemented with Tensorflow.

We used the SQuAD test and development sets. The optimizer used was For evaluation, to metrics are used:

- **EM** calculates the exact string match between the predicted answer and the truth
- **F1** is the harmonic average of precision and recall. Precision is the ratio of true positives to all number of positives while recall is the number of true positives to the number of actua positives.

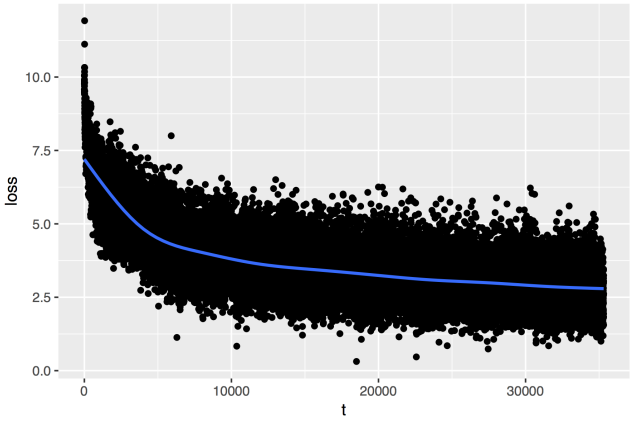


Figure 3: average loss per batch over training time

Our model

'Unarmed fox hunting on horseback with hounds is the type of hunting most closely associated with the United Kingdom ; in fact , ' hunting ' without qualification implies fox hunting . What in other countries is called ' hunting ' is called ' shooting ' (birds) or ' stalking ' (deer) in Britain . Originally a form of vermin control to protect livestock , fox hunting became a popular social activity for newly wealthy upper classes in Victorian times and a traditional rural activity for riders and foot followers alike . Similar to fox hunting in many ways is the chasing of hares with hounds . Pairs of Sight hounds (or long-dogs) , such as greyhounds , may be used to pursue a hare in coursing , where the greyhounds are marked as to their skill in coursing the hare (but are not intended to actually catch it) , or the hare may be pursued with scent hounds such as beagles or harriers . Other sorts of foxhounds may also be used for hunting stags (deer) or mink . Deer stalking with rifles is carried out on foot without hounds , using stealth .'

In England , what is hunted when ' shooting ' is called for ? birds

Figure 4: An example of the model's functional performance

The results of the model on the SQuAD hidden test set compared to human performance and the state-of-the-art single models is given in Table 1. Our model achieves an EM score of 39.253% and an F1 score of 52.608%.

Figure 4 demonstrates an instance of model's correct prediction for a rather difficult question answering with a large paragraph. Figure 5 demonstrates three instances of model's mistakes. In the first mistake, model gets confused. The second mistake depicts model answering longer than necessary and the third one demonstrates a correct but incomprehensive answer.

We also conducted some more analysis on the Dev set to better understand our model. In figure 6(a), the performance across different paragraph lengths is demonstrated. Intuitively, it's expected for the model to perform better for shorter answers. However, as observed, there's no meaningful connection between performance and the paragraph length. Nonetheless, the performance generally degrades as the question length increases as in Figure 6(b). The same happens with answer length and the model is totally defunct for answers longer than 15 tokens as depicted in Figure 6(c) . In figure 6(d), the performance across different types of questions is demonstrated. The model is the most adept for the simple "when" quesiton compared to the complex "why" questions.

6 Conclusion

In this work, we proposed a new model using ideas of multi-perspective context matching [12] and bidirectional attention flow [13]. The model identified the span of the answer by matching each word in paragraph with the question via multiple perspectives. Experimental results on the test set of SQuAD showed that the model is capable of generating answers with an acceptable near baseline precision. The next step would be modifying the existing blocks and experimenting new blocks for the model to improve its performance. Additionally, being short on time, fine tuning is an important stage that was not done in our experiments.

Our model

- Mistake 1

"The roles and responsibilities of the minority leader are not well-defined . To a large extent , the functions of the minority leader are defined by tradition and custom . A minority leader from 1931 to 1939 , Representative Bertrand Snell , R-N.Y. , provided this " job description " : " He is spokesman for his party and enunciates its policies ... "

What despcription was assigned to minority leader in part ?

Truth: He is spokesman for his party and enunciates its policies .

Prediction: tradition and custom

- Mistake 2

"The Earth of the early Archean ... . During this time , the Earth 's crust cooled enough that rocks and continental plates began to form . Some scientists think because the Earth was hotter , that plate tectonic activity was more vigorous than it is today , resulting in a much greater rate of recycling of crustal material . This may have prevented cratonisation and continent "

What might have a very hot earth stopped from occurring ?

Truth: cratonisation and continent formation

Prediction: cratonisation and continent formation until the mantle cooled and convection

- Mistake 3

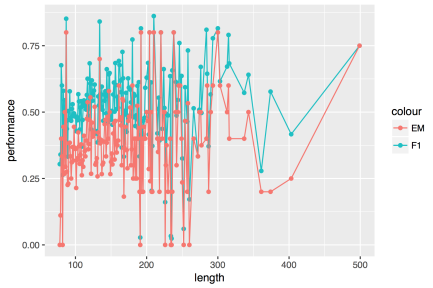
"Tnfection begins when a... Most humans are not easily infected . Those who are weak , sick , malnourished , have cancer or are diabetic have increased susceptibility to chronic or persistent infections . Individuals who have a suppressed immune system ... "

What group of humans have increased susceptibility to chronic or persistent infections ?

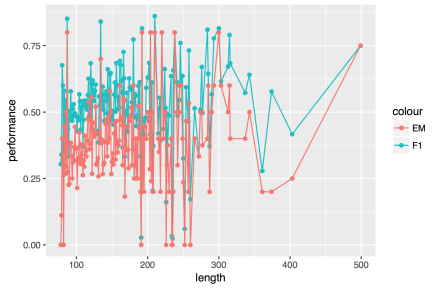
Truth: weak , sick , malnourished , have cancer or are diabetic

Prediction: malnourished

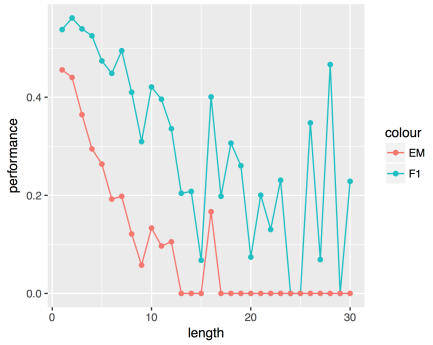
Figure 5: Three instances of model's mistakes



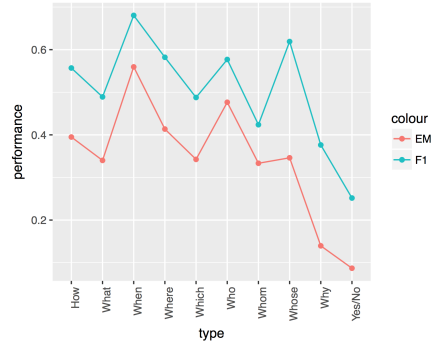
(a) Effect of paragraph's length on performance



(b) Effect of question's length on performance



(c) Effect of answer's length on performance



(d) Effect of question's type on performance

Figure 6: Effects of paragraph's length, answer's length, and question's length and type on model's performance

Model	**EM**	**F1**
r-net	72.40	80.75
MPCM[12]	68.88	77.77
BiDAF[13]	72.40	80.75
ReasoNet	70.55	79.36
Our model	**39.253**	**52.608**
Human Performance[4]	82.30	91.22

Table 1: Result on the SQuAD hiddent test set

References

[1] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. *Mctest: A challenge dataset for the open-domain machine comprehension of text.* In EMNLP, volume 3, pp. 4, 2013

[2] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. *Teaching machines to read and comprehend.* In Advances in Neural Information Processing Systems, pp. 1693–1701, 2015.

[3] Danqi Chen, Jason Bolton, and Christopher D. Manning. *A thorough examination of the cnn/daily mail reading comprehension task.* In Association for Computational Linguistics (ACL), 2016.

[4] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. *Squad: 100,000+ questions for machine comprehension of text.* In Empirical Methods in Natural Language Processing (EMNLP), 2016.

[5] Jason Weston, Sumit Chopra, and Antoine Bordes. *Memory networks.* In ICLR, 2015.

[6] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. *Reasonet: Learning to stop reading in machine comprehension.* arXiv preprint arXiv:1609.05284, 2016.

[7] Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. *Text understanding with the attention sum reader network.* In ACL, 2016.

[8] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. *Attention-overattention neural networks for reading comprehension.* arXiv preprint arXiv:1607.04423, 2016.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural machine translation by jointly learning to align and translate.* ICLR, 2015

[10] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. *Sentence similarity learning by lexical decomposition and composition.* In Proceddings of Coling 2016.

[11] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014 *Glove: Global vectors for word representation.* In EMNLP, volume 14, pages 1532– 43.

[12] Zhiguo Wang, Haitao Mi, Wael Hamza and Radu Florian. *Multi-Perspective Context Matching for Machine Comprehension.* arXiv preprint arXiv:1611.01603, 2016

[13] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi. *Bidirectional Attention Flow for Machine Comprehension.* arXiv preprint arXiv:611.01603, 2017.

[14] Matthew D Zeiler. *Adadelta: an adaptive learning rate method.* arXiv preprint arXiv:1212.5701, 2012.

Appendix A: What we did and did not try on the proposed model

There were several options to improve the model we did not had the time for. Some of them wre as follows:

- fine tuning the parameters was not done which plays a ver important rule in performance
- We used the basic tokenizer and did not have the time to use CoreNLP.
- In all the state-of-the-art models, there exists a character embedding layer which output concatenates with the word vectors.
- exploiting different methods of attention was a path we didn't take
- Using an ensemble was an option that needed training our model several times which the timing did not allow us to do

There were several attempts we had to make this model better on the Dev set. Some of them were as follows:

- We tried this model with and without dropout and the result without using dropout was slightly better (around 1 EM score and around 2 F1 scores better)
- We utilized ADAM and SGD optimizer in addition to AdaDelta. ADAM performed well compared to SGD. AdaDelta, however, was slightly better
- In the last layer, instead of using an inner production before softmax, we tried using a two layer feedforward network. The results were significantly degraded. (F1:30 EM:19)
- After contextual matching layer, we tried concatenating the output of paragraph's BiLSTM with vectors $\mathbf{m}_j$ and then feed it to the next layer. The performance was not acceptable.

Appendix B: Other simple model we tried

Before reaching to the mentioned model, we tried a number of other models, all of which did not perform well: our first model consisted of:

- It had the paragraph filtering layer,
- then it had the bidirectional LSTMs for paragraph tokens and question tokens,
- then, for each word in paragraph, we concatenated it's bi-states with that of the last and first states of the question,
- and in the end we had a feedforward network for prediction

This simple model, however, with and without drop-out did not perform well; EM=19 and F1=26.