RETRO-BIDAF: A RETROSPECTIVE READER OVER BIDAF

STAT 8931 COURSE PROJECT REPORT

Yu Yang*
School of Statistics
University of Minnesota
yang6367@umn.edu

December 16, 2020

ABSTRACT

Machine reading comprehension (MRC) with unanswerable questions is a central problem in natural language understanding, which requires the machine to determine the correct answer as well as the answerability given the passage. In this paper, I propose a model named Retro-BiDAF, which combines the idea of retrospective reader and Bidirectional Attention Flow (BiDAF) to examine the effectiveness of retrospective reader without utilizing Pre-trained Contextual Embeddings (PCE). The proposed model is evaluated on the SQuAD2.0 benchmark dataset and the results show that the retrospective reading strategy indeed helps with the model performance in terms of EM and F1.

Keywords Retrospective reader · BiDAF · Question answering · SQuAD2.0

1 Introduction

Ever since the existence of multiple large-scale datasets (Hermann et al., 2015; Rajpurkar et al., 2016; Joshi et al., 2017; Trischler et al., 2017; Rajpurkar et al., 2018), machine reading comprehension (MRC) has become a central task in natural language understanding. The early MRC systems (Dhingra et al., 2017; Cui et al., 2017) implicitly hypothesize that all questions are answerable given the passage, which is not applicable to real-life scenarios. On the contrast, a good MRC system should not only produce the correct answer when the question is answerable but also can tell whether a question is answerable given the passage context. Such a requirement makes the MRC system much closer to real-world applications, and in the meanwhile requires some extra design for the MRC reader.

In this paper, we focus on the span-based MRC task, where the answer is a chunk of text taken directly from the passage if the question is answerable. Typically, a reader for such tasks with non-answerable questions works on three subtasks: (1) build a language model (LM) as the encoder; (2) construct a decoder to get the answer span; (3) design a verifier to check answerability.

To get a good reader, it is important to work on each individual part as well as the organization structure. For the encoder, much work has been done to get a good language model. Now, the state-of-art models at the SQuAD 2.0 leaderboard are based on Pre-trained Contextual Embeddings (PCE), including ELMo(Peters et al., 2018), BERT(Devlin et al., 2018), ALBERT(Lan et al., 2019), and many other derivatives. Whereas their non-PCE counterparts include traditional word embeddings, word2vec(Mikolov et al., 2013), GloVe(Pennington et al., 2014), FastText(Mikolov et al., 2018), and character-level embeddings(Seo et al., 2016), Transformers(Yu et al., 2018) and so on. PCE approaches are likely to outperform the best non-PCE models by a large margin. And a large proportion of the code will be externally-sourced if we choose PCE approaches.

As for the decoder, the **Bid**irectional **A**ttention **F**low (BiDAF) (Seo et al., 2016) model was a popular choice prior to the BERT era. And in recent years, the BERT-based(Devlin et al., 2018; Lan et al., 2019; Clark et al., 2020) backbone architectures are widely used due to their excellent performance. And for the organization structure, instead of simply stack these three parts in a pipeline or in a concatenation way, Zhang et al. (2020) proposed a novel idea: retrospective

*Check `https://github.umn.edu/YANG6367/squad` for code and more results.

reader, which consists of two parallel modules: a sketchy reading module and an intensive reading module, along with a rear verification module. They showed that the retro-reader over ELECTRA backbone architecture improves both the EM and F1 significantly and achieves the state-of-art results on two benchmark MRC challenge datasets SQuAD2.0 and NewsQA.

Motivated by their results, in this paper, I will examine the effectiveness of the retrospective reading idea along the non-PCE track. A model named Retro-BiDAF is proposed by applying the idea of retrospective reading to the BiDAF backbone architecture with GloVe word embeddings. The proposed model is evaluated on the SQuAD2.0 dataset and compared with the BiDAF baseline model.

This paper goes as follows. Section 1 describes some background of MRC systems with non-answerable questions and briefly summarizes the related work; Section 2 describes the architecture of the proposed model in detail; Section 3 gives the implementation settings and shows the experiment results; Section 4 concludes the paper with some discussion.

2 Proposed Model

The proposed model combines the idea of BiDAF and retrospective reader, and it is described in Figure 1. There are in total three parts: the sketchy reading module, the intensive reading module, and the rear verification module. The basic idea of retrospective reader is to mimic human reading. Human usually scan through the questions and text to have a coarse judgement on whether the question is answerable or not, and then read carefully to make the final decision. In a similar manner, retro-reader uses the sketchy module to make the first round of front verification, and the intensive reader does both the span prediction and the second round of answerability verification. The final answerability is determined by the rear verification, based on the aggregation of the two front verification layers.

The sketchy and intensive reading modules should be trained separately and can be trained in parallel. After training, data samples will go through rear verification and threshold-based answer verification, after which the model will output the span prediction if answerable and a null string otherwise. Throughout this section, let c represent the context, q represent the question, N be the length of the context, M be the length of the question, D be the embedding size, H be the hidden size of the model, and n be the number of samples in the training dataset.

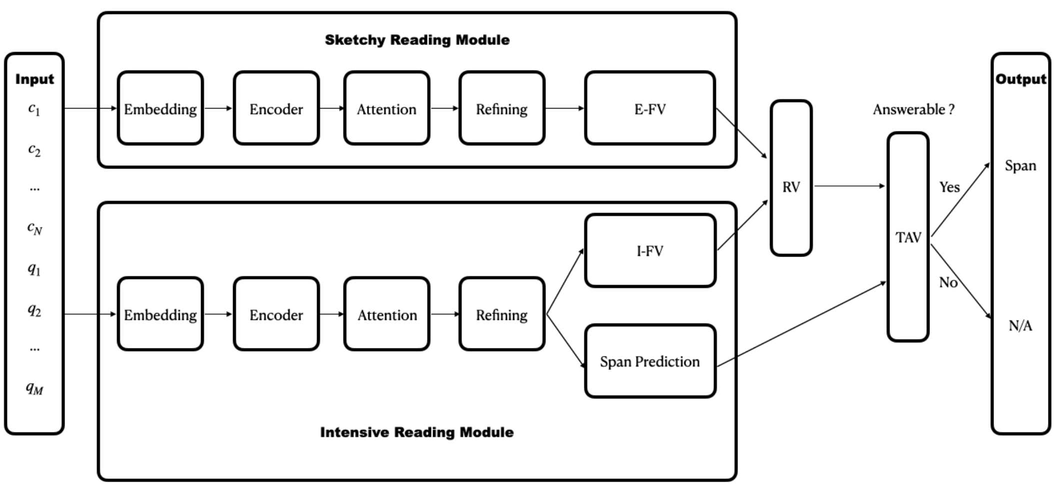

Figure 1: Model Architecture

2.1 Sketchy Reading Module

Embedding The raw input text sequences are firstly represented as embedding vectors by GloVe look up, and then they are passed to a projection layer to get dimension H, and then a Highway Network (Srivastava et al., 2015) is used to refine the embeddings. Denote the embeddings of the context as $c_1, c_2, \cdots, c_N \in \mathbb{R}^H$, and the question as $q_1, q_2, \cdots, q_M \in \mathbb{R}^H$.

Encoder The encoder takes the embedding vectors as input and uses a one-layer bidirectional LSTM (Hochreiter and Schmidhuber, 1997) to learn the temporal dependencies of the input sequences. The output is the concatenation of the forward and backward hidden states. The encoded sequences are then $c_1, c_2, \cdots, c_N, q_1, q_2, \cdots, q_M \in \mathbb{R}^{2H}$.

Attention A bidirectional attention flow layer is applied to allow attention flow from the context to the question and from the question to the context so as to get a better context representations. In brief, by utilizing the similarity matrix defined as $S_{ij} = w_{sim}^T[c_i; q_j; c_i \circ q_j] \in \mathbb{R}$, we can perform Context-to-Question (C2Q) Attention to get C2Q attention outputs $a_i, i = 1, \cdots, N$, and perform Question-to-Context (Q2C) Attention to get Q2C attention output $b_i, i = 1, \cdots, N$. And finally, we get the output g_i defined as below.

$$
\begin{aligned}
\overline{S}_{i,:} &= \text{softmax}(S_{i,:}) \in \mathbb{R}^M, \quad \forall i \in \{1, \cdots, N\} \\
a_i &= \sum_{j=1}^{M} \overline{S}_{i,j} a_j \in \mathbb{R}^{2H}, \quad \forall i \in \{1, \cdots, N\} \\
\overline{\overline{S}}_{:,j} &= \text{softmax}(\overline{S}_{:,j}) \mathbb{R}^N, \quad \forall j \in \{1, \cdots, M\} \\
S' &= \overline{S}\, \overline{\overline{S}}^T \in \mathbb{R}^{N \times N} \\
b_i &= \sum_{j=1}^{N} S'_{i,j} \mathbb{R}^{2H}, \quad \forall i \in \{1, \cdots, N\} \\
g_i &= [c_i; a_i; c_i \circ a_i; c_i \circ b_i] \in \mathbb{R}^{8H}, \quad \forall i \in \{1, \cdots, N\}
\end{aligned}
$$

where $\circ$ represents elementwise multiplication.

Refining Given the input vectors g_i, a two-layer LSTM is used to refine the sequence vectors after the attention layer, and outputs $m_i \in \mathbb{R}^{2H}, i = 1, \cdots, N$. This refinement incorporates the temporal information between context representations conditioned on the question.

External Front Verification In External front verification (E-FV), the pooled last token $m_N \in \mathbb{R}^{2H}$ is passed to a fully connected layer to get classification logits. We use binary cross entropy loss as the training objective:

$$\mathbb{L}^{ans} = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)],$$

where $\hat{y_i} \propto \textit{SoftMax}(\textit{Linear}(m_N))$ denotes the predicted probability by E-FV and y_i is the target indicating the answerability.

2.2 Intensive Reading Module

The input text sequences go through the same procedure as the sketchy reader except E-FV to get a decent representation, namely, $g_i \in \mathbb{R}^{8H}$ and $m_i \in \mathbb{R}^{2H}$ for $i = 1, \cdots, N$.

Span Prediction The target is to produce two vectors of probabilities: $s, e \in \mathbb{R}^N$, where s_k represents the predicted probability that the answer span starts at position k in the context, and similarly e_l is the predicted probability that the answer span ends at position l in the context.

Firstly, a one-layer bidirectional LSTM is applied to the refining output $m_1, \cdots, m_N \in \mathbb{R}^{2H}$, producing $m'_1, \cdots, m'_N \in \mathbb{R}^{2H}$. Let $G \in \mathbb{R}^{8H \times N}$ be the matrix composed of $g_1, \cdots, g_N$, and $M, M' \in \mathbb{R}^{2H \times N}$ be the matrix composed of $m_1, \cdots, m_N$ and $m'_1, \cdots, m'_N$ respectively. Then s and e are calculated as follows.

$$s = \text{softmax}(W_s[G; M]), \quad e = \text{softmax}(W_e[G; M']),$$

where $W_s, W_e \in \mathbb{R}^{1 \times 10H}$ are learnable weight matrices.

The training objective of answer span prediction is the sum of the cross entropy loss for the start and end predictions,

$$\mathbb{L}^{span} = -\frac{1}{n} \sum_{i=1}^{n} [\log(s_{y_i^s}) + \log(e_{y_i^e})],$$

where y_i^s and y_i^e are the ground-truth start and end positions of example i.

Internal Front Verification Similar to E-FV, the pooled last token $m_N \in \mathbb{R}^{2H}$ is passed to a fully connected layer to get classification logits and we use binary cross entropy as the training objective of classification verification:

$$\mathbb{L}^{ans} = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log \overline{y}_i + (1 - y_i) \log(1 - \overline{y}_i)],$$

where $\overline{y}_i \propto \textit{SoftMax}(\textit{Linear}(m_N))$ denotes the predicted probability by I-FV and y_i is the target indicating the answerability.

Joint Loss Span prediction and internal front verification are trained jointly with the loss being defined as the weighted sum of the span loss and verification loss.

$$\mathbb{L} = \alpha_1 \mathbb{L}^{span} + \alpha_2 \mathbb{L}^{ans},$$

where are α_1 and α_2 are weights.

2.3 Inference

At test time, the ith data sample will go through the sketchy reader and the intensive reader respectively, obtaining the E-FV predicted probability $\hat{y}_i$ and the I-FV predicted probability $\overline{y}_i$, and the span prediction probability vectors $s, e \in \mathbb{R}^N$.

Rear Verification Rear verification (RV) is the combination of the predicted probabilities given by E-FV and I-FV,

$$v = \beta_1 \hat{y} + \beta_2 \overline{y},$$

where β_1 and β_2 are weights.

Threshold-based Answerable Verification Threshold-based answerable verification (TAV) is a heuristic strategy to check the answerability using the predicted answer start and end logits (Devlin et al., 2018; Lan et al., 2019). A slight modification is used in this project. Denote the output start and end predicted probabilities as s and e, and the rear verification score as v. We define the has-answer score $score_{has}$ and the no-answer score $score_{na}$ as follows:

$$\begin{aligned} score_{\textit{has}} &= \max(s_k \cdot e_l), 1 \leq k \leq l \leq N, \\ score_{\textit{na}} &= \lambda_1 (s_0 \cdot e_0) + \lambda_2 v^2, \\ score_{\textit{diff}} &= score_{\textit{has}} - score_{\textit{na}}. \end{aligned}$$

where λ_1 and λ_2 are weights.

Note that for inference purpose, an Out-of-Vocabulary (OOV) token is appended to each context paragraph in preprocessing, so s_0 and e_0 represent the probabilities at the OOV token position. With a pre-specified threshold δ, the model predicts the answer span if $score_{\textit{diff}} > \delta$ and predicts no-answer otherwise.

Discretized Predictions If $score_{\textit{diff}} > \delta$, the soft prediction vectors s and e are then discretized to get start and end indices respectively. Concretely, we choose the pair (k, l) of indices that maximizes $s_k \cdot e_l$ subject to $k \leq l$ and $l - k + 1 \leq L_{\max}$, where $L_{\max}$ is a hyperparameter which sets the maximum length of a predicted answer and it is set as 15 by default.

3 Experiments

3.1 Setup

For the BiDAF baseline model, the learning rate is set as 0.5 and L2 weight decay is set as 0. The batch size per GPU is 64, and the number of epochs is 30. The dev metric begins to decrease at about epoch 22. For the sketchy reader, Adam optimizer with a warm-up learning rate of 0.02 is used. The number of epochs is 30, but the model arrives at plateau early at about epoch 5. And for the intensive reader, most parameters are the same as the baseline model, despite of the number of epochs, which is set as 50, since the dev metric doesn't decrease till epoch 47.

The manual weights are $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = \lambda_1 = \lambda_2 = 0.5$. And the threshold value $\delta = -0.006$ is tuned using the dev set.

3.2 The SQuAD2.0 Challenge Benchmark Dataset

The paragraphs in SQuAD2.0 are from Wikipedia. There are around 150k questions in total, and about half of the questions are not answerable given the provided paragraph. And if a question is answerable, then the answer is a span of text in the context paragraph. An example of a $\langle$question, context, answer $\rangle$ triple is as below[2].

[2]Check more examples on the challenge website: `https://rajpurkar.github.io/SQuAD-explorer/explore/v2.0/dev/`

Question: Who can be in the Victorian cabinet?
Context Paragraph: The Premier of Victoria is the leader of the political party or coalition with the most seats in the Legislative Assembly. The Premier is the public face of government and, with cabinet, sets the legislative and political agenda. Cabinet consists of representatives elected to either house of parliament. It is responsible for managing areas of government that are not exclusively the Commonwealth's, by the Australian Constitution, such as education, health and law enforcement. The current Premier of Victoria is Daniel Andrews.
Ground Truth Answers: (1) representatives; (2) representatives elected to either house of parliament; (3) representatives elected to either house of parliament

3.3 Evaluation

Metrics Two official metrics are used to evaluate the model performance: F1 score and Exact Match (EM) score. And AvNA is provided alongside to get a better idea of the answerability prediction. Exact Match (EM) is a binary measure of whether the output matches the ground truth answer exactly. This is a fairly strict metric and a minor difference from the ground-truth will give a score of 0. While F1 is a less stric metric and it is defined as the harmonic mean of precision and recall[3]. When a question has no answer, both the F1 and EM score are 1 if the model predicts no-answer, and 0 otherwise. And AvNA stands for **A**nswer **v**ersus **N**o **A**nswer and it measures the classification accuracy when we only consider answerability prediction. Note that the EM and F1 scores are averaged across the entire evaluation dataset to get the reported scores.

3.4 Results

Table 1 compares the results of the baseline BiDAF model and the proposed Retro-BiDAF model. We can see that the two official evaluation metrics get improved a lot, but the answerability score decreases slightly. To better understand such a phenomenon, I checked some dev prediction results (some are shown in Appendix 5.1), and found that the retro-reader tends to predict more no-answer than the baseline model. Recall that when a question has no answer, both the F1 and EM score are 1 if the model predicts no-answer, and 0 otherwise. Given the prior information that the proportion of answerable and unanswerable questions are the same, it is possible that the improvement of F1 and EM score cannot be fully credited to the increasing learning capability of the model.

Also, note that since I have no access to the test leadboard in CS224N course, I just evaluate my model on the development set. This is not a perfect estimate of the generalizability, but it should give us some idea about the comparison between the BiDAF baseline model and the proposed Retro-BiDAF model.

Model	*Dev*		
	EM	*F1*	*AvNA*
BiDAF	58.28	55.13	**64.70**
Retro-BiDAF	**61.15**	**59.45**	63.94

Table 1: The results (%) from single models for SQuAD2.0 challenge.

4 Conclusion

In this paper, I explored the idea of retrospective reader and proposed a model called Retro-BiDAF, which combines the the retro-reader idea and the BiDAF backbone architecture with GloVe embeddings. The results show a great boost in two official evaluation metrics, but a slight downgrade in answerability prediction accuracy. The increment in EM and F1 scores implies the effectiveness of the retro-reader idea. At the same time, the reduced value of answerability accuracy raises an unsolved issue, which requires more effort to unveil the hidden mechanism in the future.

References

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 593–602, 2017.

[3]Read more about F-score on Wikipedia: `https://en.wikipedia.org/wiki/F-score`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1832–1846, 2017.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28: 1693–1701, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Tomáš Mikolov, Édouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

CS224n Course Staff. Cs 224n default final project: Question answering on squad 2.0. *Last updated on February*, 5, 2020. URL `https://web.stanford.edu/class/cs224n/project/default-final-project-handout.pdf`.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *ACL 2017*, page 191, 2017.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*, 2018.

Zhuosheng Zhang, Junjie Yang, and Hai Zhao. Retrospective reader for machine reading comprehension. *arXiv preprint arXiv:2001.09694*, 2020.

5 Appendix

5.1 Sample Prediction Results

Both Succeed

dev/13_of_25/text_summary
tag: dev/13_of_25/text_summary

test/baseline-01

step 0

- **Question:** What is the process in which neutrophils move towards the site of inflammation called?
- **Context:** Neutrophils and macrophages are phagocytes that travel throughout the body in pursuit of invading pathogens. Neutrophils are normally found in the bloodstream and are the most abundant type of phagocyte, normally representing 50% to 60% of the total circulating leukocytes. During the acute phase of inflammation, particularly as a result of bacterial infection, neutrophils migrate toward the site of inflammation in a process called chemotaxis, and are usually the first cells to arrive at the scene of infection. Macrophages are versatile cells that reside within tissues and produce a wide array of chemicals including enzymes, complement proteins, and regulatory factors such as interleukin 1. Macrophages also act as scavengers, ridding the body of worn-out cells and other debris, and as antigen-presenting cells that activate the adaptive immune system.
- **Answer:** chemotaxis
- **Prediction:** chemotaxis

dev/13_of_25/text_summary
tag: dev/13_of_25/text_summary

test/retro_reader-01

step 0

- **Question:** What is the process in which neutrophils move towards the site of inflammation called?
- **Context:** Neutrophils and macrophages are phagocytes that travel throughout the body in pursuit of invading pathogens. Neutrophils are normally found in the bloodstream and are the most abundant type of phagocyte, normally representing 50% to 60% of the total circulating leukocytes. During the acute phase of inflammation, particularly as a result of bacterial infection, neutrophils migrate toward the site of inflammation in a process called chemotaxis, and are usually the first cells to arrive at the scene of infection. Macrophages are versatile cells that reside within tissues and produce a wide array of chemicals including enzymes, complement proteins, and regulatory factors such as interleukin 1. Macrophages also act as scavengers, ridding the body of worn-out cells and other debris, and as antigen-presenting cells that activate the adaptive immune system.
- **Answer:** chemotaxis
- **Prediction:** chemotaxis

Figure 2: Both models succeed on answerable questions

dev/14_of_25/text_summary
tag: dev/14_of_25/text_summary

test/baseline-01

step 0

- **Question:** What cells do not play a role in long-term active memory?
- **Context:** Long-term active memory is acquired following infection by activation of B and T cells. Active immunity can also be generated artificially, through vaccination. The principle behind vaccination (also called immunization) is to introduce an antigen from a pathogen in order to stimulate the immune system and develop specific immunity against that particular pathogen without causing disease associated with that organism. This deliberate induction of an immune response is successful because it exploits the natural specificity of the immune system, as well as its inducibility. With infectious disease remaining one of the leading causes of death in the human population, vaccination represents the most effective manipulation of the immune system mankind has developed.
- **Answer:** N/A
- **Prediction:** N/A

dev/14_of_25/text_summary
tag: dev/14_of_25/text_summary

test/retro_reader-01

step 0

- **Question:** What cells do not play a role in long-term active memory?
- **Context:** Long-term active memory is acquired following infection by activation of B and T cells. Active immunity can also be generated artificially, through vaccination. The principle behind vaccination (also called immunization) is to introduce an antigen from a pathogen in order to stimulate the immune system and develop specific immunity against that particular pathogen without causing disease associated with that organism. This deliberate induction of an immune response is successful because it exploits the natural specificity of the immune system, as well as its inducibility. With infectious disease remaining one of the leading causes of death in the human population, vaccination represents the most effective manipulation of the immune system mankind has developed.
- **Answer:** N/A
- **Prediction:** N/A

Figure 3: Both models succeed on unanswerable questions

Both Fail

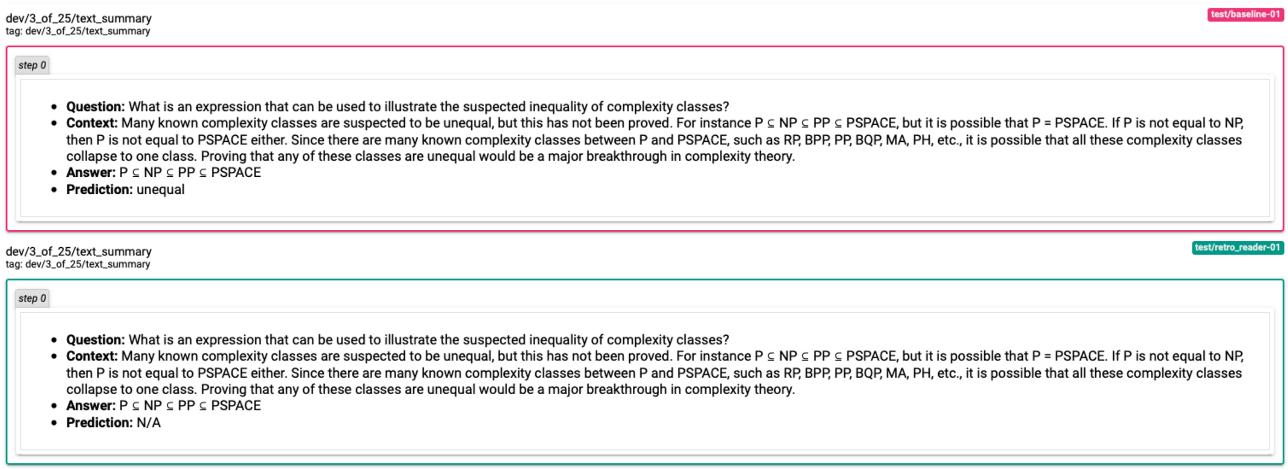

Figure 4: Both Models fail on answerable questions

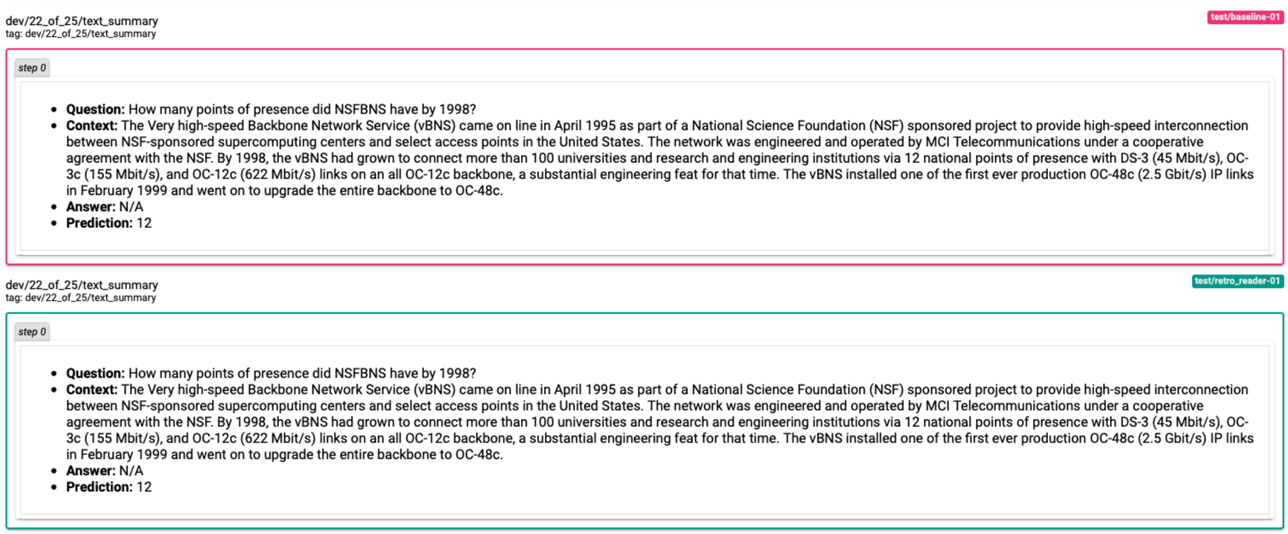

Figure 5: Both Models fail on unanswerable questions

Retro-Reader Outperforms Baseline

dev/21_of_25/text_summary
tag: dev/21_of_25/text_summary
test/baseline-01

step 0

- **Question:** What religion did the Yuan discourage, to support Buddhism?
- **Context:** Western musical instruments were introduced to enrich Chinese performing arts. From this period dates the conversion to Islam, by Muslims of Central Asia, of growing numbers of Chinese in the northwest and southwest. Nestorianism and Roman Catholicism also enjoyed a period of toleration. Buddhism (especially Tibetan Buddhism) flourished, although Taoism endured certain persecutions in favor of Buddhism from the Yuan government. Confucian governmental practices and examinations based on the Classics, which had fallen into disuse in north China during the period of disunity, were reinstated by the Yuan court, probably in the hope of maintaining order over Han society. Advances were realized in the fields of travel literature, cartography, geography, and scientific education.
- **Answer:** Taoism
- **Prediction:** Tibetan

dev/21_of_25/text_summary
tag: dev/21_of_25/text_summary
test/retro_reader-01

step 0

- **Question:** What religion did the Yuan discourage, to support Buddhism?
- **Context:** Western musical instruments were introduced to enrich Chinese performing arts. From this period dates the conversion to Islam, by Muslims of Central Asia, of growing numbers of Chinese in the northwest and southwest. Nestorianism and Roman Catholicism also enjoyed a period of toleration. Buddhism (especially Tibetan Buddhism) flourished, although Taoism endured certain persecutions in favor of Buddhism from the Yuan government. Confucian governmental practices and examinations based on the Classics, which had fallen into disuse in north China during the period of disunity, were reinstated by the Yuan court, probably in the hope of maintaining order over Han society. Advances were realized in the fields of travel literature, cartography, geography, and scientific education.
- **Answer:** Taoism
- **Prediction:** Taoism

Figure 6: Retro-Reader outperforms baseline

Baseline Outperforms Retro-Reader

dev/5_of_25/text_summary
tag: dev/5_of_25/text_summary
test/baseline-01

step 0

- **Question:** What country was under the control of Norman barons?
- **Context:** Subsequent to the Conquest, however, the Marches came completely under the dominance of William's most trusted Norman barons, including Bernard de Neufmarché, Roger of Montgomery in Shropshire and Hugh Lupus in Cheshire. These Normans began a long period of slow conquest during which almost all of Wales was at some point subject to Norman interference. Norman words, such as baron (barwn), first entered Welsh at that time.
- **Answer:** Wales
- **Prediction:** Wales

dev/5_of_25/text_summary
tag: dev/5_of_25/text_summary
test/retro_reader-01

step 0

- **Question:** What country was under the control of Norman barons?
- **Context:** Subsequent to the Conquest, however, the Marches came completely under the dominance of William's most trusted Norman barons, including Bernard de Neufmarché, Roger of Montgomery in Shropshire and Hugh Lupus in Cheshire. These Normans began a long period of slow conquest during which almost all of Wales was at some point subject to Norman interference. Norman words, such as baron (barwn), first entered Welsh at that time.
- **Answer:** Wales
- **Prediction:** N/A

Figure 7: Baseline outperforms retro-reader

5.2 Code

The code is accessible at `https://github.umn.edu/YANG6367/squad`. The skeleton of the code refers to the one in CS224N Default Final Project (Staff, 2020).