DeepDoc: Natural Language Processing with Deep Neural Networks for the American Board of Internal Medicine Certification Exam

Jonathan Wang
Biomedical Informatics
jonwang1@stanford.edu

Britni Chau
Computer Science
britnic@stanford.edu

Kinbert Chou
Computer Science
klchou@stanford.edu

Abstract

High quality practice and decision making is dependent on knowledge of a small team of physicians. With the growing amount of diagnoses, tests, and procedures, there is much room for improvement with clinical decision support technologies. We train a model to answer review questions for the American Board of Internal Medicine Certification Exam. We adapt approaches traditionally used for question answer tasks to our multiple choice exam, as well as experiment with the following enhancements: PubMed Embeddings, BiDAF, DrQA, SAR, GA, and RACE. Ultimately we find that GA models perform best (Accuracy: 0.38, AUROC: 0.64). Our work is an initial study towards the development of a intelligent medical QA system, demonstrating the capability of modern day machine learning to answer questions clinicians typically take many years to study for.

1 Introduction

Despite the rapid and widely successful incorporation of artificial intelligence into a plethora of different industries, when it comes to medicine, much of high quality practice and decision making is dependent almost entirely upon a single physician. Providing high quality medical care consistently involves a combination of clinical experience and knowledge derived from literature(1; 2). However, with the progressively growing number of possible medications, diagnoses, and procedures, applying high quality care becomes more difficult to maintain, requiring larger amounts of time and resources (3; 4). Hence, physicians tend to rely on personal intuition, rather than data from robust scientific studies, in clinical practice (5).

To our knowledge, no clinical information systems exist which query and answer natural language questions in a way similar to Google search engine for medical professionals. The current study proposes an initial foray into the development of an intelligent question and answer bot for medical questions. This serves to save physicians valuable time through two functions 1) information retrieval of protocols and facts 2) literature review to maintain clinical knowledge. Our interest is to take a first pass at this through the development of an algorithm which answers American Internal Medicine Board Examinations—a certification that all physicians must go through to practice general internal medicine. Though a number of companies claim to have had success in developing these algorithms (6; 7; 8), to our knowledge, none of them have reported these results or algorithms. Without published findings, it is difficult to reproduce and further progress on the development of these algorithms (9). Thus, this remains a potentially impactful problem to solve and would facilitate further development of medical question answering algorithms.

1.1 Objective

First, we will gather data points by scraping board exam questions from a reputed question resource (that we can not disclose due to privacy reasons). Then we will implement a variety of neural

network-based algorithms to demonstrate the capability of algorithms perform in the task of medical question answering.

2 Related Work

Open-domain Question-Answering (QA) systems generally consist of two parts 1) a document retriever that retrieves relevant information for answering a question, and 2) reading comprehension to find answer within a smaller selection of text.

The first document retriever systems developed by Simmons in 1964 focused on matching dependency parses of questions and answers to find relevant parts within a text corpus (10). Since then, a variety of approaches have been pioneered namely Murax in 1993 (11) and NIST TREC QA in 1999 (12). IBM"s DeepQA (13) brought much attention to this problem but ultimately, DrQA is widely known as one of the first effective neural reading comprehension information retrievers (14). For this reason, we choose to use DrQA's information retrieval in our analysis due to its demonstrated effectiveness and readily available code.

Initial natural language processing (NLP) for reading comprehension focused on simpler reading comprehension methods developed by Schank, Hirschman, and Burges (15; 16; 17). In 2016, advances in computing power and data availability resulted in the first well-performing neural systems for reading comprehension. The Stanford Question Answering Dataset (SqUAD) has facilitated lots of advancement in the field especially through the public leaderboard (18). Namely two simple networks have came out of these developments, Stanford Attentive Reader (SAR) and Gated Attention Reader (GA) (19; 20). We choose both of these due to their simplicity of implementation and high performance.

Our NLP task is multiple choice, thus it strays away from open-domain Question-Answering because we have one more input, the set of answers, and a different output, the probability of each answer. RACE is the largest dataset to our knowledge that uses modern day machine learning to answer multiple choice questions (21). We adapt code for their SAR and GA, and use it for our own task.

Finally, we looked at QA systems currently being implemented in medical contexts. The main clinical decision support system used today is UpToDate which simply retrieves articles of relevant information (22). MedQA and MEANS attempt to answer questions through ontologies and semantic web technologies, but neither leverage neural networks for reading comprehension (23; 24). We could not find any literature citing the use of neural networks to answer questions for the American Medical Board Examinations.

3 Methods and Experiments

3.1 Data

3.1.1 Collection

Our data is pulled from 3,600 American Board of Internal Medicine Certification Exam review questions. Due to privacy concerns, we are unable to disclose the source. Each question is comprised of a question, accompanying context passage, and 4 or 5 answer choice selections. Once an answer choice is selected, the correct answer, explanation passage, key point, and learning objective are revealed. There may be images or tables in the question and/or explanation. The exam website dynamically loads questions using Javascript, thus downloading and parsing HTML files directly did not provide the information we desired. We use Charles Web Debugging Proxy to identify the location the of the API that the Javascript calls to request the information (25). We then use a Python script adapted from StackOverflow to scrape these examples from the API to obtain raw text data (26; 27). Ultimately, 3564 examples were scraped from 2012, 2015, and 2018 exams, where 36 were removed due to missing information.

3.2 Preprocessing

We used Regex and BeautifulSoup to parse the following fields for each question (28). An example of a passage, question, explanation, and answer choices can be found in Appendix A.

- Question ID (str): UUID of example
- Question (str): Question
- Passage (str): Contextual information for question
- Answer Choices (dict): key is answer choice (char), value is answer choice descriptor (str)
- Learning Objective (str): Learning objective of the question
- Key Point (str): Key idea needed to answer the question properly
- Distribution of Answer Selections (list[float]): the percent distribution of answer selections made my human test takers
- Question Type (str): Category of question (cardiovascular, neurology, etc)
- Year (str): Year question was published
- Table in Explanation (bool)
- Image in Explanation (bool)
- Table in Passage (bool)
- Image in Passage (bool)

The resulting splits are: Train: 2364 examples (2012, 2016 data), Dev: 600 examples (half of 2018 data), Test: 600 examples (half of 2018 data).

3.3 Prediction Task

Our question answering task is defined as the following. Given a passage p, question q, and a list of 4-5 candidate answers a_i, our system will select the the correct answer.

We also adapt the task such that the question answering system has access to a corpus of text to help it answer the question (Fig 1). In this case, we use explanations from the training set as this corpus of text, with the intuition that similar question's explanations may contain useful information. In this case, the true explanation e is used for the training set, and an explanation retrieved from the training set via an information retriever $\hat{e}$ is used for the dev and test set. This is explained in more detail within the DrQA section.

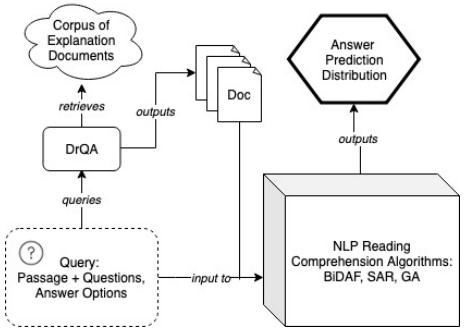

Figure 1: High-level architecture of our system. DrQA retrieves top three relevant explanations. The retrieved explanations, passage, question, and answer options are fed as inputs into a reading comprehension algorithm to output a prediction.

3.4 Architectures

For this project, our focus was on the adaption of existing work for a new problem. Additional experiments and architectures are described in detail. Diagrams for each of the different architectures can be found in Appendix B.

3.4.1 BiDAF Baseline

The Bidirectional Attention Flow model (BiDAF) is a machine comprehension question answering model (29). The BiDAF model takes as input a context and a question answerable by a span of that context. This input is transformed into pre-trained word embedding vectors, which are further refined through another embedding layer that learns to adjust the word vectors based on context of the input. Finally, a bidirectional attention layer that models the similarity of words between the context and answer conditions an LSTM layer to generate the predicted spans.

For our baseline, we adapted the default BiDAF model provided to suit multiple choice question answering. Instead of using the SQuAD dataset, we use our passage, question, and answer choices from the medical exam.

To adjust the model for multiple choice question answering, we model each example as follows:

$$PA = \{p; < sep >; a_1; ...; < sep >; a_5\}$$
$$Q = \{q\}$$

PA consists of the question passage pre-pended to each answer choice descriptor. Each answer choice descriptor is pre-pended by a separation token token as shown above. Q consists of the corresponding question. The max tokenized length of PA is 461 and Q is 37. We pad all inputs to this corresponding length.

We first use the embedding layer to derive glove word embeddings from our padded inputs. The embeddings are then passed through an adapted highway, encoding, and modeling layer from the default BiDAF baseline. For our output layer, we used linear transformations to combine the attention and LSTM outputs. However, since the question is multiple choice, we modify the architecture to identify the SEP token for the corresponding answer. To do this, we remove the second LSTM and linear layer used find the ending of the span. To find the appropriate SEP token, we use a mask at all indices except those occupied by SEP tokens prior to applying the final softmax. Thus, this gives us a probability distribution over our 4 or 5 answer choices. In prediction, we take the argmax of the resulting probabilities as our predicted answer selection.

The modeling of the multiple choice question structure has, to our knowledge, never been tried before. The intuition is that SEP token hidden states will learn to highlight whether or not the answer follows it. The bidirectional flow will help identify what portions of question and the context/answer that matter when looking for the SEP token of interest.

3.4.2 Stanford Attentive Reader Baseline

We adapt the implementation of the Stanford Attentive Reader (SAR) baseline for RACE, a multiple choice question answering dataset (19; 21). The input to SAR is a tuple of query, u, composed of question passage prepended to question, the correct answer explanation passage, e, and answers: $\{u, e, a_1, ...a_5\}$. Instead of bidirectional LSTMs, RACE uses bidirectional Gated Recurrent Units (GRU) to encode the embedding representation of u to h^u, e to $h^e_1...h^e_n$, and a_i to h^{a_i}. GRUs are used because they are easy to modify and do not require memory units; they perform similarly to LSTMs and train more quickly. RACE then uses a bilinear attention defined by $\alpha_i = softmax_i((h^e_i)^T W_1 h^u)$, $s^e = \Sigma_i alpha_i h^e_i$ between each explanation passage position and the question to summarize the most relevant part of the explanation with respect to the question (s^e). Next, once more using the bilinear attention a similarity score is determined between summarized explanation and each option. We alter the model such that if the question originally had four options, the dummy coded fifth option has a score that is masked to 0. The argmax of the softmax of the scores is returned as the model's answer prediction: $pred_i = softmax_i(h^{a_i} W_2 s^e)$.

3.4.3 Gated Attention Reader Baseline

We adapt the implementation of the Gated Attention Reader (GA) baseline for RACE as well (20; 21). The input to Gated Attention Reader (GA) is a tuple of query, u composed of question passage prepended to question, the correct answer explanation passage, e, and answers: $\{u, e, a_1, ...a_5\}$. GA reader derives a multi-hop representation of the query and explanation by fine tuning the query and explanation's embedding representations iteratively across hops. The idea is that its learning mimics the comprehension processing pattern of humans, in that the semantic understanding of the words of the passage are developed over multiple passes of the passage in relation to having understood the query. This is the benefit of multiple hop architecture in contrast to previous models that are restricted to token or sentence level attention. The passage is read over k hops, where the input x to the kth layer is the bidirectional GRU encoding of the embedding of the explanation from the $k-1$ layer: $e^k = \overleftrightarrow{GRU}^k_e (x^{(k-1)})$. Over k hops, the query is is refined in tangent using a separate bidirectional GRU: $u^k = \overleftrightarrow{GRU}^k_u (y)$. Per hop, a multiplicative attention mechanism of the form

$\alpha_i = softmax(u^T e_i)$, $\tilde{u_i} = u\alpha_i$, $x_i = e_i \odot \tilde{u_i}$ is applied to extract the most relevant parts of the explanation in relation to query. Multiplicative attention is used because it has been found to be more effective as a fine-grained sentiment filter than additive attention. Once all k hops have been made and the final representation of the explanation, s^e, has been attained, RACE's implementation of GA applies a bilinear attention between option and summarized explanation to derive a similarity score. We alter the model such that if the question originally had four options, the dummy coded fifth option has a score that is masked to 0. The argmax of the softmax of the scores is returned as the model's answer prediction: $pred_i = softmax_i(h^{a_i} W s^e)$. Note that RACE's implementation of GA disregards character level word embeddings.

3.5 Modifications and Experiments

3.5.1 Varying Options Lengths

We altered the BiDAF Baseline to answer multiple choice questions, as described in 3.4.1. Additionally, we alter the SAR and GA baselines to answer 4-5 multiple choice questions through a mask applied to the dummy coded fifth option prior to the final softmax.

3.5.2 Bio-NLP Embeddings

Bio-NLP 2016 is a neural word embedding trained on scientific literature from PubMed (30). In place of GloVe embeddings, we use Bio-NLP embeddings to encode p, q, $a_1 ... a_5$. Our initial GLoVE embeddings yielded embeddings for 74% of tokenized words in our vocabulary. The implemented PubMed embeddings yielded embeddings for 90% of words, suggesting vocabulary within our examples contain rarer medical nomenclature. Additionally, the PubMed embeddings demonstrate a slight increase in performance in comparison to the raw baselines (Table 1). Note that we used default embeddings for our baselines, this meant that the GLoVE embeddings used in SQuAD were dimension 300, while in SAR and GA they were dimension 100. Meanwhile, the PubMed embeddings were dimension 200. Thus, this improvement in accuracy may be due to the differing embedding sizes rather than the embeddings themselves.

3.5.3 DrQA

DrQA is a question answering system characterized by its unique breakdown of the question answering task. Unlike rote KB QA methods which typically uses a non-sophisticated search function to extract documents from a domain, DrQA applies state of the art methods to extract the most relevant documents in its knowledge domain (Document Retriever), then predict the answer from those documents (Document Reader).

This differs from search and Knowledge-Based (KB) QA systems such as IBM Watson (13) whose performance heavily relies on repetitively seeing accurate information, or SQuAD which assumes no prior knowledge in its answers and whose answers can be found in the short accompanying text to its questions (18). Due to our limited data we can't rely on breadth of knowledge domain to answer our questions. Furthermore, unlike SQuAD, the exam requires prior knowledge to solve its questions. Thus we try to implement a document retriever method to improve performance for our task.

We adapt DrQA's document retriever to our task as follows: First, each explanation provided in the training set serves as a document. In total, there are 2364 documents. TF-IDF weighting using bigram feature vectors determine the similarity between document and a query represented by the concatenation of passage, question, and the four-five answer options $\{p; q; a_1; ...a_5\}$. Due to the additional complexity added from the use of bigram instead of bag-of-words feature vectors, Murmur3 hashing is used to preserve time and space efficiency. We then feed the top three retrieved documents along with the original inputs $\{p; q\}, \{a_1 ... a_5\}$ into SAR and GA for the dev and test set (described in more deetail below). Thus, instead of having the inputs of $\{p\}$, $\{q\}, \{a_1 ... a_5\}$, the DrQA model will have inputs of $\{\hat{e}\}$, $\{p; q\}, \{a_1 ... a_5\}$. In the training set, we use the true explanations e instead of DrQA retrieved explanations $hate$ so the model learns on documents that actually contain signal. The intuition here is that similar explanations from previous iterations of the test may have useful information for answering a given question in the dev/test set.

To retrieve documents from DrQA, we query using passage, question, and answers concatenated into a single string $\{p; q\}, \{a_1 ... a_5\}$. To determine what to query with, we performed a brief analysis of

what queries would yield the true explanation most frequently. We found that question, passage, and answer yielded over 87% accuracy within the top 3 documents. However, dropping answers yielded only around 46% accuracy within the top 3 documents, which suggests there is a highly correlated mapping between answers and explanations. When we tried answers alone, this yielded an even higher accuracy of around 93% in the top three. Using such a small amount of text to query would not yield robust matches when looking for relevant explanations that don't directly match the question, so we choose to use the top 3 documents from a $\{p; q\}, \{a_1...a_5\}$ query.

To make use of the three retrieved articles in the dev/test set, we run the model three times on each of the articles separately before ensembling using a 0.5, 0.3, 0.2 weighting scheme on the probability outputs. This weighted scheme is based on the percentage of times the first, second, or third document retrieved was the true explanation as decribed above.

3.5.4 Hyperparamater Tuning

To improve the performance of the final models, we perform hyperparameter tuning using random search over a hyperparameter space. Appendix C contains information on the default hyperparameters used for the models, as well as the search space used in our hyperparameter search.

Out of 138 sets of parameters, the set {learning rate 0.570811, dropout rate 0.80538, gradient clipping 10.730696, epochs 50, hidden size 6} resulted in the best performance for GA with a dev accuracy of 38.6%. Out of 153 sets of parameters, the set {learning rate 0.079253, dropout rate 0.605804, gradient clipping 5.93294, epochs 67, hidden size 58} resulted in the best performance for SAR with a dev accuracy of 38.1%.

We tuned each model for 32 hours, totaling 64 GPU hours.

Our results show that tuning improved performance for SAR, but decreased performance on GA (Table 1). We suspect that this is likely due having too large of a search space and not enough GPU computing hours.

3.5.5 Ensembling

We ensemble the two tuned models by averaging their probability outputs. This gave a very slight improvement in model performance (0.2% accuracy) (Table 1).

4 Evaluation

4.1 Evaluation Metrics

We only report multi-class accuracy in lieu of f1, precision, and recall. This is because our questions have between four and five answer choices. Thus, when we reconstructed the analysis into a one vs all format (essentially flatenning the predicted probabilities in the nx5 array and removing the fifth column for rows without a fifth question), micro-averaged f1, precision, and recall are mathematically equivalent to accuracy.

Our scraped review questions also contain the percentage of people who answered the question correctly. We create a weighted score metric that weights each question by (1-percentage answered correctly). This captures how well the algorithm does on questions people are not generally good at answering. For example, if 90% of people answer a question correctly, it will have a 0.1 weight.

Finally we report micro-averaged AUROC, which represents the ability of the model to distinguish between correct and incorrect answers.

4.2 Results

Overall, our results demonstrate that GA w/ DrQA appears to be the best model for this problem with an accuracy of 0.37, weighted score of 0.36, and AUROC of 0.64. (Table 1). This is actually reasonable performance, as passing the exam generally requires around 50-60% accuracy, with a top performers scoring around 85%. Additionally, on the RACE dataset, these same state of the art algorithms only perform with an accuracy of 43% (with a human ceiling performance of 93%).

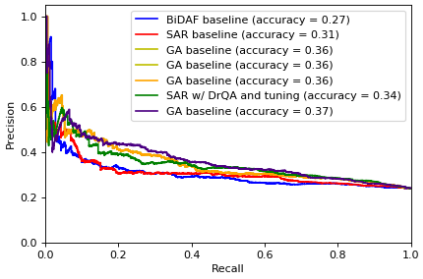

Figure 2: Precision-Recall curve for top performing models and baselines.

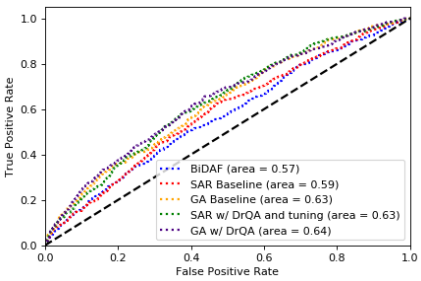

Figure 3: ROC curve for top performing models and baselines.

Table 1: Evaluation metrics show outperformance of GA over SAR models. Here, the ensembled model represents averaged probability outputs from the two tuned models. BiDAF = Bidirectional Attention Flow model; SAR = Stanford Attnetive Reader model; GA = Gated Attention Model; BioEmbeddings = embeddings extracted from Bio-NLP group; *= includes BioEmbeddings; bold indicates highest score in each category.

Model	Accuracy	Weighted Score	AUROC
Random	0.222	0.234	0.500
BiDAF Baseline	0.273	0.255	0.571
SAR Baseline	0.310	0.304	0.588
GA Baseline	0.360	0.341	0.626
SAR w/ BioEmbeddings	0.322	0.301	0.605
GA w/ BioEmbeddings	**0.377**	0.344	0.638
SAR w/ DrQA*	0.325	0.299	0.616
GA w/ DrQA*	0.373	**0.357**	**0.640**
SAR w/ DrQA and tuning*	0.335	0.309	0.634
GA w/ DrQA and tuning*	0.335	0.302	0.628
Ensembled Model	0.337	0.308	0.633

We speculate this may be due to the ability of GA to capture long-term dependencies through its multi-hop architecture.

We plot Precision-Recall Curve to demonstrate the tradeoff between true positive rate and positive predictive value for our top performing SAR and GA models compared to baselines 2. We also plot an ROC curve to demonstrate the tradeoff between sensitivity and specificity for these same models 3. These are generated using a flattened version of the array of scores, with a removed fifth column for questions without a fifth answer to account for varying number of options. This is also known as micro averaging.

Additionally, we evaluate the ensembled model (average of the two tuned SAR and GA models) with true explanations e in comparison to using drQA to find relevant explanations from the training set $\hat{e}$ (Table 2). Surprisingly, we perform only slightly better with correct explanations, despite training on data with correct explanations. This strongly suggests that our reading comprehension model is not performing well for this task as it is unable to find relevant information from even the true explanations. We believe this may be due to the longer lengths that our inputs now contain, as GRUs perform better at understanding shorter spans of text, and these longer spans of text may not work well with the neural network architectures employed.

5 Qualitative analysis

It is hard to diagnose whether errors are a result of information retrieval or reading comprehension due to the blackbox nature of neural network architectures. Thus, we looked at the top 5 examples (15 explanations) that had the highest probability score for the correct answer. Additionally, we looked at the bottom 5 examples (15 explanations) that had the lowest probability score for the correct answer

Table 2: Ensembled model with correct explanations shows little improvement over predictions using DrQA retrieved explanations. Bold indicates highest score in each category.

Model	Accuracy	Weighted Score	AUROC
Correct Explanations	**0.340**	0.304	**0.639**
DrQA Explanations	0.337	**0.308**	0.633

from our ensembled model. These examples were then randomized, and we had a team member label whether the retrieved explanations were relevant to the question or whether the explanation was helpful in answering the question or not (Table 3).

As shown, in only very few cases the returned explanation appears to be helpful (7-13%). Thus, when we retrieve the top three explanations, only about 30% of the questions contain useful information in the retrieved explanations. Many times the retrieved explanation appears to contain information relevant to only one answer choice rather than all four or five answer choices. Ultimately this reveals that our information retrieval system does not seem to be grabbing documents with information as useful as the explanations from . In light of the results from Table 2, it appears that both reading comprehension and our document retriever could benefit from additional modifications.

Table 3: Qualitative analysis reveals our information retrieval system only retrieves helpful explanations 6-13% of the time.

Examples	Relevant Explanation (%)	Helpful Explanation (%)
Top 5	**26.6**	**13.3**
Bottom 5	20.0	6.6

We additionally performed an analysis of the model performance on questions with images and with tables within the passage. We found there was little difference (within 2% accuracy) in performance on these questions.

6 Conclusions, Limitations, Future Work

This study is the first to our knowledge to tackle the problem of answering questions to the American Internal Medical Board physician certification examination. Within the span of 5 weeks, we are able to adapt existing state of the art network architectures used for the RACE dataset to answer questions with an accuracy of 38%. This is surprisingly good performance, considering a passing score on the exam is around 50-60% and initial models trained on the RACE dataset were around 43%. Upon making the leaderboard public, scores on the RACE dataset improved to 75%, we hope a similar dataset and effort can be made for medical questions in the future. This could rapidly progress the field in a way that could save physicians time and improve the standard of care.

As noted in our discussion, there is great room for improvement in both our neural reading comprehension algorithms as well as our document retriever. We are limited by the amount of data publicly available for these exams, as well as time to tune our hyperparameters. Additionally, as with many neural network architectures, diagnosing errors in the model is difficult to the black box nature of the algorithms. Having access to past explanations for questions in a model is also a relatively unrealistic way to represent information gathered from outside sources, however, in theory, physicians should have access to this information as well prior to taking an exam.

Future work in this area would include more creative ways of leveraging outside information with DrQA, for example, through Wikipedia or PubMed. Additionally, we are interested in experimenting with other network architectures including character level-embeddings or Bio-BERT (31). Finally, it would be compelling to externally validate our model on a real-world exam made publicly available.

7 Additional Information

- **Mentor:** Suvadip Paul

- **External Collaborators:** We have two advisors for the project: Yuhao Zhang, a graduate student with Christopher Manning, and Jonathan Chen, an assistant professor in the division of Biomedical Informatics Research in the Department of Medicine
- **Sharing Project:** Jonathan Wang is sharing the data with CS270 class, where they are working on the same dataset for a different prediction task.

References

[1] D. L. Sackett, W. M. C. Rosenberg, J. A. M. Gray, R. B. Haynes, and W. S. Richardson, "Evidence based medicine: what it is and what it isn't," 1996.

[2] G. H. Guyatt, D. L. Sackett, J. C. Sinclair, R. Hayward, D. J. Cook, R. J. Cook, E. Bass, H. Gerstein, B. Haynes, A. Holbrook, and Others, "Users' guides to the medical literature: IX. A method for grading health care recommendations," *Jama*, vol. 274, no. 22, pp. 1800–1804, 1995.

[3] S. Timmermans and A. Mauck, "The promises and pitfalls of evidence-based medicine," *Health Affairs*, vol. 24, no. 1, pp. 18–28, 2005.

[4] D. T. Durack, "The weight of medical knowledge.," *The New England journal of medicine*, vol. 298, no. 14, pp. 773–5, 1978.

[5] R. Madhok, "Crossing the Quality Chasm: Lessons from Health Care Quality Improvement Efforts in England," *Baylor University Medical Center Proceedings*, vol. 15, no. 1, pp. 77–83, 2017.

[6] Dom Galeon, "This robot has passed a medical licensing exam with flying colours | World Economic Forum."

[7] "CloudMedx Clinical AI outperforms human doctors on a US medical exam."

[8] "This AI Just Beat Human Doctors On A Clinical Exam."

[9] Sam Finnikin, "Babylon's 'chatbot' claims were no more than clever PR | Article | Pulse Today."

[10] R. F. Simmons, "Natural language question-answering systems: 1969," *Communications of the ACM*, vol. 13, no. 1, pp. 15–30, 1970.

[11] J. Kupiec, "Murax: A robust linguistic approach for question answering using an on-line encyclopedia," in *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 181–190, ACM, 1993.

[12] E. M. Voorhees and D. M. Tice, "The trec-8 question answering track evaluation," in *TREC*, vol. 1999, p. 82, Citeseer, 1999.

[13] D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, and E. T. Mueller, "Watson: beyond jeopardy!," *Artificial Intelligence*, vol. 199, pp. 93–105, 2013.

[14] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions," mar 2017.

[15] R. C. Schank, "The yale ai project," *SAM–A story understander, Research Rept*, vol. 43, 1975.

[16] L. Hirschman, M. Light, E. Breck, and J. D. Burger, "Deep read: A reading comprehension system," in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pp. 325–332, Association for Computational Linguistics, 1999.

[17] M. Richardson, C. J. Burges, and E. Renshaw, "Mctest: A challenge dataset for the open-domain machine comprehension of text," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 193–203, 2013.

[18] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," *CoRR*, vol. abs/1806.03822, 2018.

[19] D. Chen, J. Bolton, and C. D. Manning, "A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task," jun 2016.

[20] B. Dhingra, H. Liu, Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Gated-Attention Readers for Text Comprehension," jun 2016.

[21] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "RACE: Large-scale ReAding Comprehension Dataset From Examinations," apr 2017.

[22] T. Isaac, J. Zheng, and A. Jha, "Use of uptodate and outcomes in us hospitals," *Journal of hospital medicine*, vol. 7, no. 2, pp. 85–90, 2012.

[23] M. Lee, J. Cimino, H. R. Zhu, C. Sable, V. Shanker, J. Ely, and H. Yu, "Beyond information retrieval—medical question answering," in *AMIA annual symposium proceedings*, vol. 2006, p. 469, American Medical Informatics Association, 2006.

[24] A. B. Abacha and P. Zweigenbaum, "Means: A medical question-answering system combining nlp techniques and semantic web technologies," *Information processing & management*, vol. 51, no. 5, pp. 570–594, 2015.

[25] "Charles Web Debugging Proxy • HTTP Monitor / HTTP Proxy / HTTPS & SSL Proxy / Reverse Proxy."

[26] G. v. . C. v. W. e. I. C. Rossum, "Python tutorial," *Python*, 1995.

[27] "How to scrape a website that requires login first with Python - Stack Overflow."

[28] "Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation."

[29] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional Attention Flow for Machine Comprehension," nov 2016.

[30] B. Chiu, G. Crichton, A. Korhonen, and S. Pyysalo, "How to Train good Word Embeddings for Biomedical NLP," pp. 166–174, 2016.

[31] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," jan 2019.

8 Appendix

8.1 Appendix A: Example of a Review Question

Passage: A 76-year-old woman is evaluated in the emergency department for dizziness, shortness of breath, and palpitations that began acutely one hour ago. She has a history of hypertension and heart failure with preserved ejection fraction. Medications are hydrochlorothiazide, lisinopril, and aspirin.

On physical examination, she is afebrile, blood pressure is 80/60 mm Hg, pulse rate is 155/min, and respiration rate is 30/min. Oxygen saturation is 80% with 40% oxygen by face mask. Cardiac auscultation reveals an irregularly irregular rhythm, tachycardia, and some variability in S1 intensity. Crackles are heard bilaterally one-third up in the lower lung fields.

Electrocardiogram demonstrates atrial fibrillation with a rapid ventricular rate.

Question: Which of the following is the most appropriate acute treatment?

Answer Options: A. Adenosine B. Amiodarone C. Cardioversion D. Diltiazem E. Metoprolol

Correct answer: C. Cardioversion.

Explanation: This patient with atrial fibrillation is hemodynamically unstable and should undergo immediate cardioversion. She has hypotension and pulmonary edema in the setting of rapid atrial fibrillation. In patients with heart failure with preserved systolic function, usually due to hypertension, the loss of the atrial "kick" with atrial fibrillation can sometimes lead to severe symptoms. The best treatment in this situation is immediate cardioversion to convert the patient to normal sinus rhythm. Although there is a risk of a thromboembolic event since she is not anticoagulated, she is currently in

extremis and is at risk of imminent demise if not aggressively treated. In addition, she acutely became symptomatic 1 hour ago, and while this is not proof that she developed atrial fibrillation very recently, her risk of thromboembolism is low if the atrial fibrillation developed within the previous 48 hours.

Adenosine can be useful for diagnosing a supraventricular tachycardia and can treat atrioventricular node-dependent tachycardias such as atrioventricular nodal reentrant tachycardia, but it is not useful in the treatment of atrial fibrillation.

Amiodarone can convert atrial fibrillation to normal sinus rhythm as well as provide rate control, but immediate treatment is needed and amiodarone may take several hours to work. Oral amiodarone may be a reasonable option for long-term atrial fibrillation prevention in this patient given the severity of her symptoms, especially if she has significant left ventricular hypertrophy.

Metoprolol or diltiazem would slow her heart rate; however, she is hypotensive and these medications could make her blood pressure lower. In addition, she is in active heart failure, and metoprolol or diltiazem could worsen the pulmonary edema.

Key Point: Patients with atrial fibrillation who are hemodynamically unstable should undergo immediate cardioversion.

8.2 Appendix B: Neural Network Architecture Diagrams

Images taken from their respective papers.

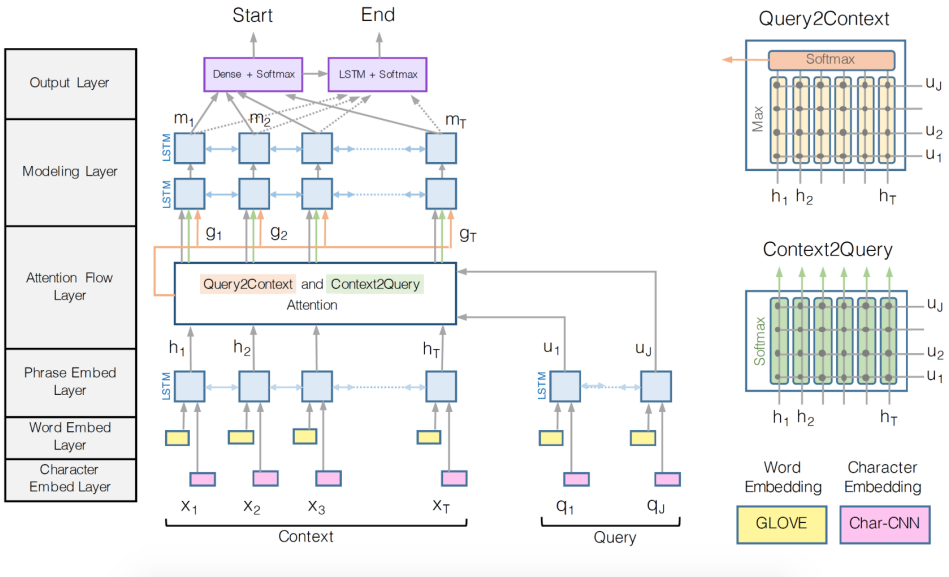

Figure 4: BiDAF Model (29)

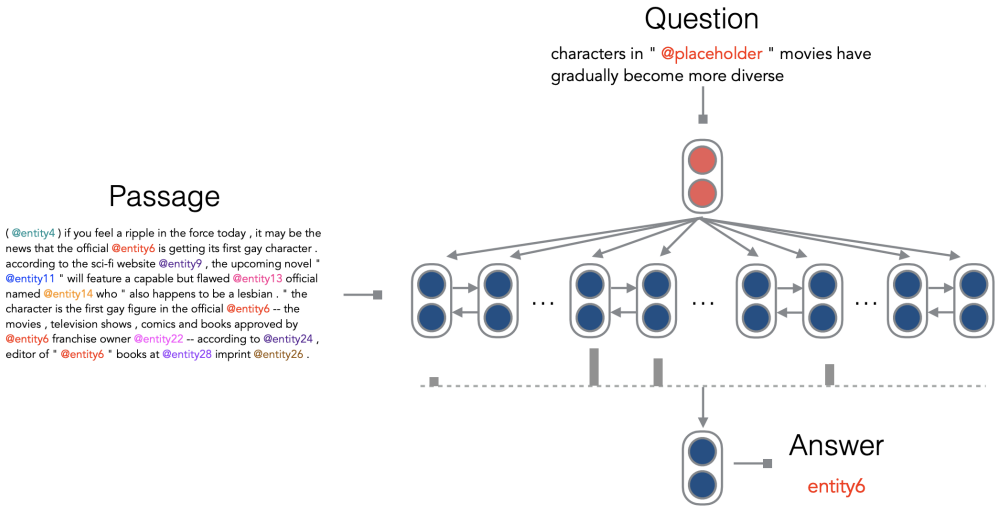

Figure 5: Stanford Attentive Reader Model (19)

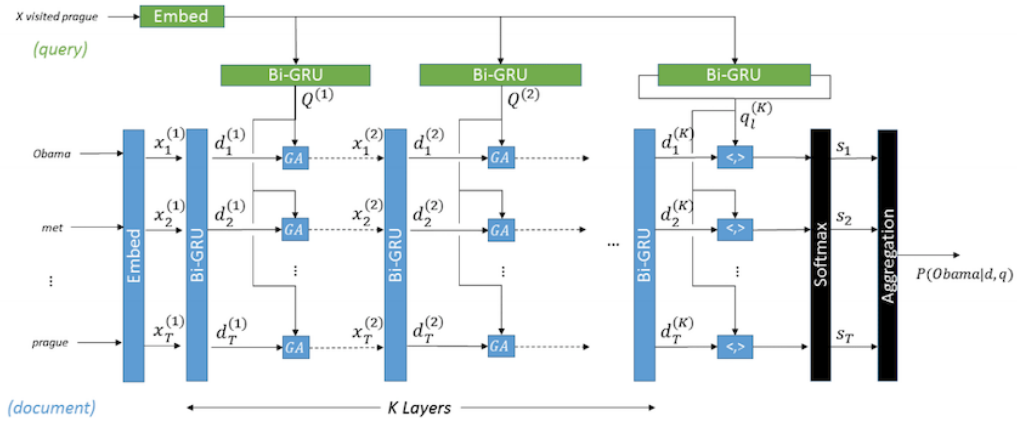

Figure 6: Gated Attention Reader Model (20)

8.3 Appendix C: Hyperparameter tuning details

For the BiDAF baseline, we use the default tuning parameters provided. The initial SAR parameters are: dropout rate 0.5, batch size 64, num epochs 100, sgd optimizer, learning rate 0.01, gradient clipping 10, hidden size 100.

The initial GA parameters are: dropout 0.5, batch size 64, num epochs 100, sgd optimizer, learning rate 0.3, gradient clipping 10, hidden size 125. These are based on best parameters found in the RACE paper.

Since both models appear to be overfitting (by greater than 10% difference between training and dev set accuracy), we increased dropout rate. We tuned parameters in the following ranges: learning rate with logarithmic distribution (0.01,1), number of epochs with uniform distribution (50,80), dropout rate with uniform distribution (0.4,1), gradient clipping with uniform distribution (4,14), and hidden size with uniform distribution (SAR: (50,125), GA: (60,130)). All other parameters maintained their default values.