Automatic Question Answering for Medical MCQs: Can It Go Further than Information Retrieval?

Le An Ha and Victoria Yaneva
Research Institute in Information and Language Processing,
University of Wolverhampton, UK
{ha.l.a, v.yaneva}@wlv.ac.uk

Abstract

We present a novel approach to automatic question answering that does not depend on the performance of an information retrieval (IR) system and does not require training data. We evaluate the system performance on a challenging set of university-level medical science multiple-choice questions. Best performance is achieved when combining a neural approach with an IR approach, both of which work independently. Unlike previous approaches, the system achieves statistically significant improvement over the random guess baseline even for questions that are labeled as challenging based on the performance of baseline solvers.

1 Introduction

Automatic question answering has seen a renewed interest in recent years as a challenge problem for evaluating machine intelligence. This has driven the development of large-scale question-answering data sets such as SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2016), WikiMovies benchmark (Chen et al., 2017), TriviaQA (Joshi et al., 2017) (to name a few), as well as the organisation of workshops such as the Machine Reading for Question Answering 2018 workshop[1]. In spite of the optimistic advances over crowd-sourced questions and online queries, automatic question answering for real exam questions is still a very challenging and under-explored area. For example, the Allen AI Science Challenge[2] invited researchers worldwide to develop systems that could solve standardized eight-grade science questions. The best system out of all 780 participating teams achieved a score of 59.31% correct answers using a combination of 15 gradient-boosting models (random baseline of 25%), while the authors report that using Information Retrieval (IR) alone results in a score of 55%.

The difficulties related to answering exam questions are partly due to the complexity of the reasoning involved and partly to the lack of large training data. Another significant reason is the fact that the existing approaches to question answering are dependent on the performance of IR systems and can rarely go far beyond the performance of such systems. While IR is a powerful method when answering questions where the correct answer is a string contained within a document, the systems fail when the sentences within the question do not individually hold a clue to what the correct answer might be (Clark et al., 2018). This is one of the characteristics of Multiple Choice Questions (MCQs) from the science domain that makes them so challenging for both machines and for humans.

In this paper we aim to address these shortcomings by developing an approach that: i) does not require that the training data (often unavailable) be in the form of multiple-choice questions and ii) does not depend on matching strings of text with one another. We use a challenging set of medical exam questions developed for the United States Medical Licensing Examination (USMLE®), a standardized medical exam that university students need to pass in order to obtain the right to practice medicine in the US. As such, the USMLE represents a very difficult set, requiring a high level of specialized professional knowledge and reasoning over facts. Furthermore, the USMLE contains a wide variety of question types such as selecting the most appropriate diagnosis, treatment, specific further examination needed, etc., all of which require application of clinical knowledge over facts.

[1] https://mrqa2018.github.io/

[2] https://www.kaggle.com/c/the-allen-ai-science-challenge

Proceedings of Recent Advances in Natural Language Processing, pages 418–422,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_049

Contributions We introduce and compare two approaches for automatic question answering that do not require training data in the form of MCQs, using Information Retrieval (IR) techniques and standard neural network models. Unlike previous work, our neural approach is independent of the performance of the IR system, as it does not build upon it. Thus, it is possible to achieve improvements over both systems by combining them, as each system has an individual contributions towards solving the problem. The best combination results in 18% improvement over a random guess baseline. The neural models achieve a statistically significicant improvement over the random baseline on the challenging sets. The code used in this study, as well as the public data[3] are made available at: https://bit.ly/2jNW2ym.

2 Related Work

Most of the recent work in the field focuses on answering reading comprehension questions from benchmark datasets such as SQuAD (Rajpurkar et al., 2016), the release of which ignited a rapid progress in the field. For example, Wang et al. (2017) use gated self-matching networks and report accuracy as high as 75.9% over a random guess baseline of around 4% and a logistic regression baseline of around 51%. Among the most successful approaches in other studies are ones that use neural models such as match-LSTM to build question-aware passage representation (Wang and Jiang, 2015), bi-directional attention flow networks to model question-passage pairs (Seo et al., 2016), or dynamic co-attention networks (Xiong et al., 2016).

As mentioned in the previous section, automatic question answering for science exams is a lot more challenging than for crowd-sourced reading comprehension questions. When applied to science questions, IR techniques: i) still perform somewhat close to the state-of-the-art and ii) fail on tasks where the correct answer is not specifically contained in relevant sentences. Clark et al. (2018) implement five of the best models from the studies on the reading comprehension data sets (TableILP (Khashabi et al., 2016), TupleInference (Khot et al., 2017), Neural entailment models (DecompAttn, DGEM, and DGEM -OpenIE) (Parikh et al., 2016), and BiDAF (Seo et al., 2016)), as well as IR models and test them on a total of 7787 science questions. The questions are divided into two sets, challenging and easy, and are targeted at students between the ages of 8 and 13. It is important to note that the authors define a question as being challenging or easy not on the basis of human performance or the age of the students it is targeted at, but based on whether it has been answered incorrectly by at least two of the baseline solvers. The results indicated that none of the algorithms performed significantly higher than the random guess baseline of 25% on the challenging set, while the performance on the easy set was within the range of 36% and 62%. According to the authors, a possible explanation for the low accuracy is that nearly all models use some form of information retrieval to obtain relevant sentences, and the retrieval bias in these systems is towards sentences that are very similar to the question, as opposed to sentences that individually differ but together explain the correct answer (Clark et al., 2018). Notably, the neural solvers performed poorly on the easy set, while the best result was achieved by an IR-only system.

3 Data

In the USMLE data each test item is a single-best-answer MCQ consisting of a stem (question) followed by several response options (distractors), one of which is the correct answer (key). An example of such an item is provided in Table 1. We divide our data into two sets: private and public (Table 2). The private data set consists of a total of 2,720 MCQs and they are not available to the public due to test security reasons. The public data set consists of 454 items from USMLE 2015 Step 1, USMLE 2016 Step 1, USMLE 2014 Step 2, and USMLE 2017 Step 2 sample leaflets. These are available at the USMLE website[4] and in our repository. For the purpose of this study, we have selected only those items that fulfill the following criteria: i) whose correct answer contains at least one heading from the Medical Subject Headings (MeSH[5]) database that is at most three words, and ii) have exactly 5 options that have at least one MeSH heading that is at most three words. The

[3] See Section 3. The Public data set used in this study consists of questions released as training materials by the USMLE.

[4] The items can be accessed at the USMLE web site at http://www.usmle.org/, for example: http://www.usmle.org/pdfs/step-1/samples_step1.pdf

[5] https://www.nlm.nih.gov/mesh/

A 56-year-old man comes to the emergency department because of a 4-day history of colicky right flank pain that radiates to the groin and hematuria. Ultrasound examination of the kidneys shows right-sided hydronephrosis and a dilated ureter. Which of the following is most likely to be found on urinalysis? (A) Erythrocyte casts (B) Glucose (C) Leukocyte casts (D) Oval fat bodies (E)* Uric acid crystals

Table 1: An example of an item from the USMLE exam (question 128, USMLE 2015 step 1 sample test questions)

	Public	Private
Number of Items	164	921
Average words per item	116	87

Table 2: Characteristics of the two sets

latter is in order to keep the random guess baseline at a constant for all items (20%). As a result, the final data that we have is 164 items for the public set and 922 for the private one.

4 Method

We develop and compare two methods for answering the USMLE questions, both of which do not require training data in the form of MCQs. The details of each method are described below.

4.1 IR-Based Method

We use a standard IR approach. First, we index 2012 MEDLINE abstracts using Lucene[6] with its default options. Then, for each item we build the five queries, where each query contains the stem and an option. We use three settings for the queries:

- All words (IR-All) (baseline)
- Nouns only (IR-Nouns)
- Nouns, Verbs, or Adjectives only (IR-NVA),

We then get the top 5 documents returned by Lucene and calculate the sum of the retrieval scores. The picked answer is the one that has the highest score when combined with the stem to form the query. This method is similar to the IR baseline described in Clark et al. (2018) and variations of it have been previously applied to medical MCQs for the purposes of distractor generation (Ha and Yaneva, 2018) and predicting item difficulty (Ha et al., 2019).

[6]https://lucene.apache.org/

4.2 Neural Network Method

For this approach we train neural networks to predict the MeSH headings for each abstract. The premise of this approach is that we hypothesise that the task of answering an USMLE item could be considered to be similar to the task of identifying the topics of a snippet of text: in the case of MEDLINE indexing, indexers read the abstract, and then choose the topics that are most relevant to the abstract; whereas in the case of taking USMLE exam, test takers read the stem, and then choose the option that is most relevant to the stem. Approaching the problem this way, we can benefit from the availability of the MEDLINE data, in which each abstract has been manually (or semi-manually) assigned most relevant subject headings. We focus only on headings that appear in the options of the set of items (see above). For our set, there are around 1000 headings. Our neural networks[7] were trained using Keras[8]. We use two main structures:

- Bidirectional LSTM (LSTM). Specifications: an input layer, followed by an embedding layer and a bidirectional layer, each of size 250. The final two layers are a flattening layer and a dense layer. The classes are weighted inversely to their frequency.
- Convoluted 1d with attention (Conv1d). Specifications: an input layer, followed by an embedding layer, three convolutional layers, and a concatenating layer, each of size 250.

[7]Preprocessing includes tokenization (using keras.preprocessing.text package in python), no lower case normalization, no number normalization, recording words with a min frequency of 5. The neural network models then further restrict the vocabulary to the first 200000 most frequent words. Out of vocabulary rate was 1%. Nadam optimizer was used with its default options (learning rate = 0.002, beta_1 = 0.9, beta_2 = 0.999, epsilon = None, schedule_decay = 0.004). Batch size = 128, activation function used in the last layer was Softmax.

[8]https://keras.io/

Accuracy	Public set	Private set
Baselines		
Random guess baseline	0.2 (.16-.26)	0.2 (.175-0.225)
IR-All baseline	0.25 (.18-.32)	0.332 (.302-.364)
IR		
IR-NVA	0.32* (.24-.39)	0.362* (.332-.395)
IR-Nouns	0.33* (.26-.41)	0.311* (.282-.342)
Neural		
LSTM	0.29 (.22-.37)	0.29* (.26-.32)
Conv1d attention	0.31* (.23-.37)	0.32* (.292-.353)
Ensemble(Conv1d+LSTM)	0.30* (.24-.39)	0.311* (.282-.342)
Neural +IR		
log(IR_NVA)+log(conv1d)	0.32* (.25-.40)	0.340* (.310-.373)
Neural as tie breaker	**0.37**** (.3-.45)	**0.396**** (.365-.429)
Neural correct when IR incorrect	0.29* (.21-.38)	0.276* (.24-.31)
Neural correct when IR tie	9 out of 15	26 out of 89

Table 3: Accuracy of the different systems. The values marked with * signify statistically significant difference over the random guess baseline and ** signifies statistically significant improvement over both baselines.

These are followed by an attention layer and a densely connected layer.

We train the models on 10,000,000 MEDLINE abstracts (the same set used in the IR approach), going through them twice. We experiment with pre-trained GloVe840b (Pennington et al., 2014) and word2vec[9], but the results are inferior to training the embedding layers from scratch. We then use the trained models to predict the probability of a MeSH heading in an option given the stem. We then average the probabilities if the option contains more than 1 heading.

4.3 Combined Method

We use two methods to combine the IR and neural model scores. The first method just adds the log value of the two scores together (log(IR_Noun)+log(Conv1d)). The second method uses the neural model scores as a tie breaker ('Neural as tie breaker'): if the IR method returns a single option, we take the result from the IR. If the IR method returns more than one options, we take the results from the neural model instead.

4.4 Baselines

We compare our results to two baselines: the probability of a random guess to pick the correct answer and the IR-All model described above.

[9]https://code.google.com/archive/p/word2vec/

5 Results and Discussion

The results from our study are presented in Table 3. Best performance is achieved by using neural model scores as a tie breaker. This result significantly outperforms both the random guess baseline and the IR-All approach.

It is interesting to note that while neural approaches alone present a significant improvement only over the random guess baseline, using neural approaches to solve ties leads to an overall increase in performance for the best combined models. The independent nature of the neural approach is best illustrated when testing its performance over items that were incorrectly solved by the best IR approaches. This is the case for 110 items from the public data set, and 587 items from the private data set, which, if we follow the definition of Clark et al. (2018), can be regarded as "challenging" since the best IR solver could not answer them correctly. In the case of Clark et al. (2018) none of the tested solvers achieved significant improvement over the random guess baseline when evaluated on the challenging questions. In our case, the neural approaches achieve 29% accuracy (32 items) for the public data set and 27.6% accuracy (162 items) for the private one, which are both statistically significant when comparing to random guess. This independence, resulting from the use of humanly produced subject headings, indicate that these headings do provide additional information with regards to the task.

A drawback of the neural approach proposed in this paper is that it relies on the availability of a manually indexed database such as MEDLINE. This limits the applicability of the approach to other domains, however, this may change when more resources become available in the future. It is important to note that in this restricted setting the method solves a very difficult problem better than any other approach so far. In the future, instead of using the adhoc neural network architectures presented in this paper, we plan to utilise state-of-the-art architectures such as Elmo (Peters et al., 2018) or BERT (Devlin et al., 2018), while using the prediction of MESH headings as an additional learning objective.

6 Conclusion

We presented an approach to automatic question answering that does not rely on training data in the form of MCQs and can perform independently from IR. We first train neural networks to predict the MeSH headings for a set of MEDLINE abstracts and then use the trained network to predict the correct answers of medical MCQs. Best performance was achieved when combining this approach with an information retrieval approach and the model significantly outperformed both a random guess baseline and one based on a common IR approach.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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

Le An Ha and Victoria Yaneva. 2018. Automatic distractor suggestion for multiple-choice tests using concept embeddings and information retrieval. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 389–398.

Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019. Predicting the difficulty of multiple choice questions in a high-stakesmedical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*.

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

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Peter Clark, Oren Etzioni, and Dan Roth. 2016. Question answering via integer programming over semi-structured knowledge. *arXiv preprint arXiv:1604.06076*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering complex questions using open information extraction. *arXiv preprint arXiv:1704.05572*.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

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

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

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

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

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.

Shuohang Wang and Jing Jiang. 2015. Learning natural language inference with lstm. *arXiv preprint arXiv:1512.08849*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 189–198.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.