Yes/No Question Answering in BioASQ 2019

Dimitris Dimitriadis(✉) and Grigorios Tsoumakas

School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
{dndimitri,greg}@csd.auth.gr

Abstract. The field of question answering has gained greater attention with the rise of deep neural networks. More and more approaches adopt paradigms which are based primarily on the powerful language representations models and transfer learning techniques to build efficient learning models which are able to outperform current state of the art systems. Endorsing this current trend, in this paper, we strive to take a step towards the goal of answering yes/no questions in the field of biomedicine. Specifically, the task is to give a short answer (yes or no) for a question written in natural language, finding clues including in a set of snippets that are related with this question. We propose three different deep neural network models, which are free of assumptions about predefined specific feature functions, while the key elements of these are the ELMo embeddings, the similarity matrices and/or sentiment information. The results have shown that incorporating the sentiment, we can improve the performance of a yes/no question answering system while the proposed learning models significantly outperform the BioASQ baseline.

Keywords: Yes/no question answering · BioASQ challenge · ELMo embeddings · Deep neural networks

1 Introduction

The recent rise of deep neural networks is having a significant impact on the field of question answering. Especially after the introduction of the SQuAD benchmark [8], more and more approaches adopt deep learning techniques, while a lot of effort has been put into building powerful and general language representation models, such as BERT [1] and ELMo [7]. Furthermore, using transfer learning, models built on a specific classification task can be reused on another task with improved results compared to building a model from scratch trained on the latter task [15].

This interesting view of solving tasks has influenced biomedical question answering too. In the BioASQ [10] challenge, which provides a benchmark for the evaluation of biomedical question answering systems, more and more approaches adopt the above paradigm (i.e. language representations and transfer learning) to build efficient models that overcome the previous state of the art. For example,

P. Cellier and K. Driessens (Eds.): ECML PKDD 2019 Workshops, CCIS 1168, pp. 661–669, 2020.
https://doi.org/10.1007/978-3-030-43887-6_59

BioBERT [5], a fined-tuned version of BERT in biomedical text, has achieved state of the art results in biomedical question answering, while previously, pre-trained word embeddings were being used in the task of biomedical question answering [2,12].

In this paper, endorsing this current perspective in question answering, we deal with this task focusing on yes/no question type. Especially, using the definition provided by BioASQ, our aim is to give an answer (yes or no) given a set of snippets that are related with a question. This task is quite similar with the reading comprehension (RC) task but it differs in some points:

1. In RC, only one snippet is related with a question and it is important that the answer is included in this. In contrast, we must cope with several snippets written by different authors and no one guarantees that the answer is part of the snippets or inferred from these.
2. The sub language of biomedical domain is complex and there are plenty of biomedical terms making the task of building a representative language representation model a difficult task. On the other hand, the RC is based on general English which means that a large amount of resources around the web can be used to build a useful language representation model.

An additional issue, we must address, is the nature of the problem of yes/no question answering. Particularly, most of the current approaches (excluding those in yes/no question answering) are focused on finding part of text in the given textual sources (i.e. snippets), whereas in our case, the answer is inferred by the given textual sources.

The main contribution of this paper dealing with the above challenges is the introduction of three different deep learning architectures. The first one is based on ELMo embeddings. The second one extends the first one by enriching the feature space with sentiment information. The last one exploits the similarity between the words of a question and the snippets to build a similarity matrix that is given as input to a deep neural network. Furthermore, we show that sentiment has impact to yes/no question answering. To the best of our knowledge, these architectures have not been used in yes/no question answering.

The rest of this paper is organized as follows. Section 2 describes our methods. Section 3 presents experimental results in BioASQ 2019 along with results on the dataset provided by the BioASQ. Section 4 makes an overview of the existing approaches in yes/no question answering focusing on systems participated in the BioASQ challenge. Finally, Sect. 5 presents the conclusions of this work.

2 Methods

We present three methods for yes/no question answering. We use ELMo embeddings in two of our methods to represent the textual sources, one of which incorporates sentiment information by leveraging SentiWordnet [3]. Our last approach uses a similarity matrix, where each cell is the cosine similarity between a word from the question and a word from the snippets, which is passed as input to a neural network.

2.1 ELMo Embeddings

In the first step, the question and the related snippets are passed through the ELMo layers (one layer that gets the question as input and the other that gets the snippets). These layers are responsible for converting the question and each snippet to multi-dimensional vectors. Let us denote the question vector as q and each snippet vector as p_i where $1 \leq i \leq m$ and m is the number of snippets. Next, we concatenate all vectors ($X = [q; p_1; p_2; ...; p_m]$) to build a joint representation of question and snippets. The produced vector is then passed through a bidirectional LSTM that is fully connected with a two-layered neural network:

$$H = BILSTM(X)$$
$$dense_1 = ReLU(W_1 * H + b_1)$$
$$dense_2 = Sigmoid(W_2 * dense_1 + b_2)$$

where W_1, W_2 are the weights of each layer and b_1, b_2 the corresponding offsets (biases). Because yes/no question answering can be considered as a binary classification problem, the last layer ($dense_2$) consists of one unit which corresponds to the target of the learning model (no = 0, yes = 1).

We used two ELMo layers instead of one, without sharing the weights across the network because the training parameters must be updated independently. Particularly, the ELMo layer getting the question as input, should pay more attention to words such as "do", "does", "is", "are" etc. and to the syntax of the question which is different from the syntax of a snippet.

2.2 ELMo Embeddings and Sentiment

As previously, we converted the given question and snippets to multi-dimensional vectors. However, we also used SentiWordnet to get the sentiment scores for each word included in the question and snippets. SentiWordnet maps each word to a triple of sentiment scores (positive, negative, neutral score).

To build the question and snippets sentiment vectors we considered Algorithm 1. Let us denote the sentiment question vector as $qs = (qs_1, qs_2, ..., qs_n)$ where qs_i is the sentiment score of the i-th word contained in the question. For snippets, we denote the snippets sentiment vectors as $ps_i = (ps_{i1}, ps_{i2}..., ps_{im})$, where ps_{ij} is the sentiment score of the j-th word contained in the i-th snippet. The sentiment vectors update the question vector as follows:

$$a = ReLU(W_1 * qs + b_1)$$
$$b = tanh(W_2 * a + b_2)$$
$$probs = Softmax(W_3 * b + b_3)$$
$$mult = q \circ probs$$

where $\circ$ denotes the element-wise multiplication between question vector and question sentiment vector. With a similar way, we update the snippets with the

sentiment scores. However, on top of the last equation we apply bidirectional LSTM. Defining the last function as H we concatenated the outputs as follows: $X = [mult; H]$, which is fully connected with a two-layered neural network as in the first method.

```
Result: Sentiment Vector
vectorS = [] ;
for word ∈ text do
    pos,neg,neut[a] = SentiWordnetWrapper(word)[b];
    if pos > neg then
        if pos > neut then
            vectorS.append(pos);
        else
            vectorS.append(neut);
        end
    else
        if neg > neut then
            vectorS.append(-neg);
        else
            vectorS.append(neut);
        end
    end
end
```

Algorithm 1. Text to Sentiment Vector

[a] The neutral score in SentiWordnet is referred as objective score

[b] SentiWordnet returns the sentiment scores of a specific synset but a word could correspond to many of these synsets, thus, we built a wrapper function that finds the most common synset corresponds to the given word and returns the sentiment scores of this synset.

Sentiment is an important information for yes/no question answering because it helps us to recognize agreements/contradictions between the given question and the related passages. Considering the question "Is the protein Papilin secreted?", the passage "the protocadherin cdh-3, and two genes encoding secreted extracellular matrix proteins, mig-6/papilin and him-4/hemicentin." agrees with the question because there aren't negative words to transform the passage to a negative statement.

2.3 Similarity Matrix

Instead of passing the question and snippets as input to a neural network, we built a similarity matrix. We first use pre-trained word vectors to represent the words of both the question and the snippets. Then, we estimate the cosine similarity for each pair of question and snippets words. Thus, each row in the similarity matrix corresponds to the similarities of a question word with all

the words contained in the snippets. This similarity matrix ($Smatrix$) passes through the following equations:

$$a = BILSTM(Smatrix)$$
$$dense_1 = tanh(W_1 * a + b_1)$$
$$dense_2 = Sigmoid(W_2 * dense_1 + b_2)$$

The inspiration of this work was from [13] which proposes a QA Matrix where each cell is the semantic similarity between a term of a question and a term of an answer. However, our similarity matrix encodes the similarity between words of the question and words from snippets. Furthermore, our bidirectional LSTM captures the dependencies between the words in snippets where each word is a vector and each dimension of this vector corresponds to the similarity of this word with a word of the question.

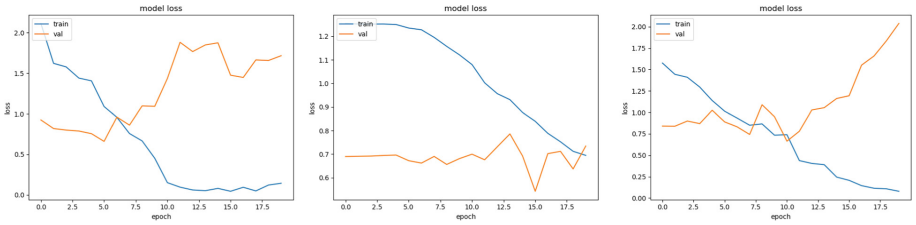
Although, recurrent neural networks aim to process sequences, the similarity matrix fits as input to these networks, considering as timesteps the rows of the similarity matrix and as dimensionality of the input, the columns of the matrix.

3 Experimental Setup and Results

To build our models, we used the BioASQ benchmark[1], which contains 745 yes/no questions along with their related snippets. 67% of these pairs of questions and snippets was used as training set and the rest 33% as validation set. We used the ELMo embeddings available at TensorFlow Hub[2] and the pre-trained word2vec embeddings provided by BioASQ[3]. Our architectures were built with the Keras framework[4]. We set the batch size to 2^4, because a larger number would lead to fewer updates of the model weights slowing down convergence. We used the Adam optimizer [4] because it works well in practice using the default learning rate (0.001), while larger learning rates cause divergence of the training criterion. Binary cross entropy was used as loss function for training the supervised neural network via the back-propagation algorithm. We used the SentiWordnet from the nltk[5].

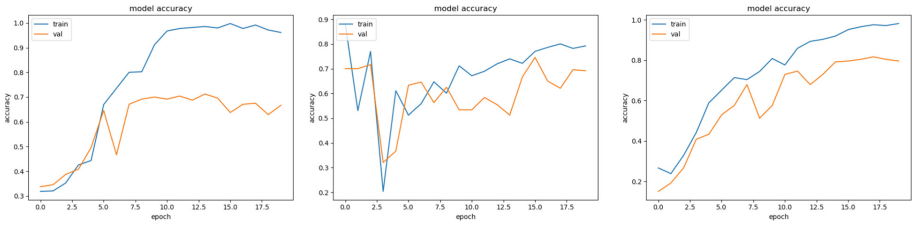
Figure 1 shows the training and validation loss of our methods. Typically, the validation loss should be similar to, but slightly higher than, the training loss. However, in our cases, this doesn't happen. The reason is the class weight that is used during training while, in the validation step, it is not defined. Thus, during training the "no" class gets more attention than during the validation. Furthermore, the convergence of the first and the last methods happens earlier than in the second method. We believe that this happens because the second method incorporates additional information (i.e. sentiment scores) to the model. Thus, the model must put more effort to incorporate this information in its

[1] http://participants-area.bioasq.org/Tasks/7b/trainingDataset/.
[2] https://www.tensorflow.org/hub.
[3] http://participants-area.bioasq.org/tools/BioASQword2vec/.
[4] https://keras.io/.
[5] http://www.nltk.org/.



(a) ELMo (b) ELMo + Sentiment (c) Similarity Matrix

Fig. 1. Training and validation loss of our methods



(a) ELMo (b) ELMo + Sentiment (c) Similarity Matrix

Fig. 2. Training and validation accuracy for our methods

feature space. In Fig. 2, we present the accuracy of each method both for training and validation. We observe that with ELMo and sentiment scores, we achieve the best accuracy before the model overfits on the training set. One phenomenon that we also observe is the increase of the validation accuracy despite the overfitting. This happens because the dataset is imbalanced, consequently, answering some questions randomly as yes, the accuracy is being increased. To participate in BioASQ 2019, we selected those models with the best accuracy before the model overfits on the training set.

Table 1 summarizes the results of our participated methods against the BioASQ baseline. As we observe, in 3/5 batches the architecture of ELMo embeddings fits better to the test sets rather than that architecture with the similarity matrix. Furthermore, sentiment seems to improve the MaF1 score in test batch 5. Finally, all methods overcome the BioASQ baseline excluding test batch 2 where our approach on Similarity Matrix is slightly worse than the baseline.

Based on the BioASQ leaderboard[6], our team (auth-qa-*[7]) is at the 2nd place in the first three batches, 5th in the fourth batch and 4th in the fifth and final batch. Furthermore, we observed that in some cases the performance of our systems is worse than the performance of other participated systems (e.g. BioBERT-DMIS, google-*-input) for a test batch, while there are some batches in which our systems overcome them. This means that there aren't clear evidences about a state of the art system in the challenge.

[6] http://participants-area.bioasq.org/results/7b/phaseB/.
[7] This is the prefix of our systems' names in the BioASQ Leaderboard.

Table 1. MaF1 score for each approach on each test batch - BioASQ 2019. Bold indicates the best score in a particular batch.

Systems	Batches - MaF1				
	1	2	3	4	5
ELMo embeddings	**.5397**	**.6296**	.4866	**.5490**	.5658
ELMo embeddings + Sentiment	–	–	–	–	**.6274**
Similarity matrix	–	.4223	**.5165**	.5461	.4697
BioASQ baseline	.4727	.4258	.1481	.4348	.4643

4 Related Work

Our work shares the high-level goal of answering yes/no questions with many works before us. Due to the fact that we cannot do full justice of related works given space constraints, we focus on two works participating in the BioASQ challenge.

Yes/No question answering can be considered as a binary classification problem where a supervised model learns to predict the truthiness of a question. In this direction, the OAQA system [14] uses a set of hand-written features that were extracted from the given question and snippets to build a binary classifier. Our work shares the main idea with the OAQA system. Particularly, we also consider the yes/no question answering as a binary classification task as well as we incorporate sentiment in one of our methods which helped to improve the accuracy. However, we enforce non-linear functions with millions of parameters to better map the input textual sources to the answer. Furthermore, we use a language representation model to capture the syntax and semantics of the raw input sources (i.e. questions and snippets) letting the model to learn from these representations to predict the answer rather than from a predefined set of features provided by an expert. Finally, instead of incorporating sentiment as a single feature in our model, we firstly find the sentiment of each word of question and snippets and next we input question and snippets sentiment vectors in the model where each dimension corresponds to the sentiment of a specific word either in question or in a snippet from the set of snippets.

A score mechanism was enforced by [9] to answer yes/no questions. Particularly, they used SentiWordnet to get the sentiment score for each word of each snippet. Then, they calculated the sentiment score for each snippet while the decision for the answer either as "yes" or "no" is based on the number of positive and negative snippets. We also use SentiWordnet to get the sentiment scores for each word of a question and snippets, however our aim is to use these sentiment scores as additional information in the feature space of our learning model rather than making these scores the central part of our methods.

Although, answering yes/no questions is very challenging in biomedicine, a few works have been proposed to solve this task in BioASQ challenge, either because the dataset provided by BioASQ was extremely imbalanced and those

participants who answered yes to the questions got very good results (e.g. [6]), or because the dataset was quite small and one cannot build efficient learning models. However, the rise of transfer learning and fine-tuned language representation models as well as the introduction of MaF1 to BioASQ challenge as additional measure to evaluate yes/no question answering systems, motivated the participants to deal with the task this year.

5 Conclusions

In this work, we present three methods for solving the yes/no question answering task. The incorporation of sentiment improved the final results w.r.t the MaF1 score. We expect that if we used language representation models fine-tuned on biomedical texts (e.g. BioBERT), the results would be better. Grid search, random search or even hyper-parameter optimization could be considered for tuning our models. The presented methods overcome the BioASQ baseline while we observed that despite the imbalanced dataset and without exhausted tuning, the models can capture some negative cases presented in the test sets.

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
2. Dimitriadis, D., Tsoumakas, G.: Word embeddings and external resources for answer processing in biomedical factoid question answering. J. Biomed. Inform. **92**, 103118 (2019)
3. Esuli, A., Sebastiani, F.: SentiWordNet: a publicly available lexical resource for opinion mining. In: LREC, vol. 6, pp. 417–422. Citeseer (2006)
4. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
5. Lee, J., et al.: BioBERT: pre-trained biomedical language representation model for biomedical text mining. arXiv preprint arXiv:1901.08746 (2019)
6. Mao, Y., Wei, C.H., Lu, Z.: NCBI at the 2014 BioASQ challenge task: large-scale biomedical semantic indexing and question answering. In: CLEF (Working Notes), pp. 1319–1327 (2014)
7. Peters, M.E., et al.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
8. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)
9. Sarrouti, M., El Alaoui, S.O.: A yes/no answer generator based on sentiment-word scores in biomedical question answering. Int. J. Healthc. Inf. Syst. Inform. (IJHISI) **12**(3), 62–74 (2017)
10. Tsatsaronis, G., et al.: An overview of the BioASQ large-scale biomedical semantic indexing and question answering competition. BMC Bioinformatics **16**(1), 138 (2015)
11. Weissenborn, D., Wiese, G., Seiffe, L.: Making neural QA as simple as possible but not simpler. arXiv preprint arXiv:1703.04816 (2017)

12. Wiese, G., Weissenborn, D., Neves, M.: Neural question answering at BioASQ 5B. arXiv preprint arXiv:1706.08568 (2017)
13. Yang, L., Ai, Q., Guo, J., Croft, W.B.: aNMM: ranking short answer texts with attention-based neural matching model. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 287–296. ACM (2016)
14. Yang, Z., Zhou, Y., Nyberg, E.: Learning to answer biomedical questions: OAQA at BioASQ 4B. In: Proceedings of the Fourth BioASQ Workshop, pp. 23–37 (2016)
15. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, pp. 3320–3328 (2014)