BioASQ 2020: a biomedical Question Answering System

Marco Cardia
m.cardia@studenti.unipi.it
Student ID: 530567

Gabriele Merlin
g.merlin@studenti.unipi.it
Student ID: 599981

Francesco Sabiu
f.sabiu@studenti.unipi.it
Student ID: 533231

ABSTRACT

In this paper, we describe the development of a state-of-the-art biomedical Question Answering system. The addressed problem is the one proposed in the Phase B of Task 8b of 2020 edition of the BioASQ challenge, a competition focused on large-scale biomedical semantic indexing and question answering. Given a set of natural language questions of several types, we developed a system able to provide precise, human-understandable answers by means of AI-based techniques. For each type of answer, a different system has been developed, achieving state-of-the-art results in almost all of them. The project, along with some comments concerning their implementation, are available at the team GitHub repository[1]. Our system achieved state-of-the-art results for two types of questions.

KEYWORDS

Natural Language Processing, Question Answering Systems, BioASQ

ACM Reference Format:
Marco Cardia, Gabriele Merlin, and Francesco Sabiu. 2020. BioASQ 2020: a biomedical Question Answering System. In ,. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The automated processing of scientific articles, journals, publications and other bibliographic entities can provide knowledge and even question answers from data concerning one or more contexts. More specifically, with respect to the Information Retrieval problem[1], the Question Answering (QA) one consists of retrieving and providing point-to-point answers from given data rather than just locate and return relevant references related to a query, which may contain the answer.

As a result of the combination of several techniques, QA tasks receive attention from the information retrieval, information extraction, machine learning, and natural language processing communities. Contextually to the current hype of Artificial Intelligence, such research interest is constantly increasing[2], since many of the current QA techniques are based on AI.

In this work, we develop AI-based solution in order to provide precise, accurate answers to biological questions of different types. Such questions have been released in the Task B (Phase B) of the 8th edition of the BioASQ challenge. The texts used for the training and testing of the techniques that are illustrated in this paper, including bibliography and questions, has been provided in English. Furthermore, the biological nature of the questions allowed us to avail ourselves of pre-trained biology-oriented pre-processing models, as described in the following sections. For our purpose, the words representation provided by them results in better performance with respect to generic models.

The paper is structured as follows. The next section provides a description to the challenge by which this work is inspired, as well as some information about the available data. Then, an overview of the current state-of-the-art architecture adopted in the QA problem is provided. In Section 4, the developed system solving the *Yes-No*, *Factoid* and *List* Question Answering problems is described. Next, we illustrate our results in Section 5, while Section 6 gathers some final considerations about this and future works, together with the conclusions.

2 BIOASQ8 CHALLENGE

BioASQ proposes challenges on biomedical semantic indexing and question answering (QA). Usually, challenges include tasks related to hierarchical text classification, machine learning, information retrieval, QA from texts and structured data, multi-document summarization and other areas.

[1] **Project Repository**
https://github.com/fsabiu/BioASQ2020

HLT '20, Human Language Technologies, 2020

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$0.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

The 8th edition of BioASQ challenge[3] was released on February 25th, 2020. In this section, the tasks composing the challenge are described with particular attention to those performed by us. Next, a brief description of the dataset is provided.

Tasks and Assumptions

The 2020 BioASQ challenge was made up of three tasks, namely *A*, *B* and *MESINESP8*. In turn, task B was composed by two phases:

- Phase A: in which, given a question, the participating systems were required to return at most 10 relevant units among concepts (from designated terminologies and ontologies), articles (from designated article repositories), snippets (from the relevant articles), and relevant RDF triples (from designated ontologies).
- Phase B: consisting of responding given questions with exact and ideal answers, making use of the snippets of the previous phase.

Phase B of the considered task has constituted the main inspiration of our work. Starting from the dataset described in the following, we developed 3 subsystems that fulfil the challenge requirements.
The official evaluation measures to which we referred are those stated by BioASQ[2]: accuracy for *yes/no questions*, mean reciprocal rank for *factoid questions*, mean F-measure for *list questions*. Furthermore, for each question the systems were required to provide a single paragraph-sized text summarizing the most relevant information. Such requirement is manually evaluated by means of a set of criteria.

The dataset

One training set and five test sets have been released during the different phases of the challenge. The former is a set of 4673 questions complemented by:

- A set of related snippets, containing useful information for the answer.
- A set of URLs at which relevant documents for the question are available.
- The type of the required answer.
- The correct answer.

Several test sets have been released during the challenge period.

Overall, training data included 4 types of questions, whose distribution is illustrated in Figure 1.

Concerning list questions, their average number of element contained in the answers was 4.90, while the following number of snippets were included for each type of question: 12.6 for yes/no, 13.72 for list, 12.36 for factoid and 12.18 for summary questions.

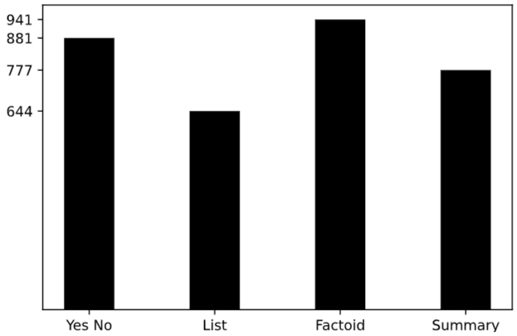

Figure 1: Question types distribution

Problem formulation

The problem addressed by our proposal, i.e. Question Answering, consists in automatically provide a solution to queries expressed in natural language[4]. Questions can be related to a specific context or not. Moreover, queries solutions can be constituted by several entities rather than by single ones. Depending on their granularity and required precision, they can be binary answers (yes/no), specific answers (factoid), list of concepts/entities, or parts of texts containing the concepts on which the answer should rely on.
In our work we propose a solution to three of these typologies: yes/no, factoid and list questions.

3 RELATED WORK

To the best of our knowledge, three different approaches exist in literature to address QA problems, namely knowledge based, information retrieval-based and AI based. Motivated by the significant gains reached in the last years by AI -based approaches, we decided to rely on such techniques, exploiting the advances in language pre-training that characterize many of the state-of-the-art models.

Recent QA systems that achieve state-of-the-art results rely on language models, which became the de-facto standard for many NLP tasks. Informally, they provide contextual representation of words by means of probability distributions over the words sequences. ALBERT[5], XLNet [6], BERT[7], RoBERTa[8] and ELMo [9] are the most adopted language models in literature providing accurate words representations. However, their performance depend on both the corpora their are trained with and the context in which they are adopted. Thus, before developing our work, we took into account several implementation of biological Question Answering systems, in which some experiment involving the mentioned model have been performed.

[2] Further details can be found at http://participants-area.bioasq.org/Tasks/b/eval_meas/

In 2019, Yoon et. al [10] availed themselves of BioBert [11]: in addition to being trained on general domain corpora, its training set is made up of PubMed articles, allowing authors' model to outperform the state-of-the-art models in list, factoid and yes/no questions on SQuAD [12] and SQuAD 2.0 [13]. In the same year, Resta et. al [14] experimented embedding extraction by means of ELMo [9], BERT [7], BioBert [11] and ELMo-Pubmed [15] for the BioASQ2019 challenge, obtaining excellent results on yes/no questions.

4 OUR APPROACH

In order to fulfill the challenge requirements (i.e. provide various types of accurate and precise answers) we developed several QA subsystems, one for each type of question. The types of the questions to which participants' systems were required to answer were Yes/No, list, summary and factoid[3]. Moreover, for each question, the systems had to return a single paragraph-sized text summarizing the most relevant information of the retrieved concepts, articles, snippets, and triples of Phase A. In this section, we provide a concise description of the implemented models.

Yes/No questions

The model that we considered for the Yes/No question answering is ELMo-Pubmed. We avail ourselves of it since, to the best of our knowledge, it represents one of the best state-of-the-art language representation model for YesNo QA tasks [10, 14].
Given such model, that is pre-trained on a large biomedical corpora, we used it to get the score of the two possible answers. The overall architecture of the implemented system is composed by Elmo illustrated in Figure 2 a pooling layer and a Feed Forward Neural Network built on the top of it.

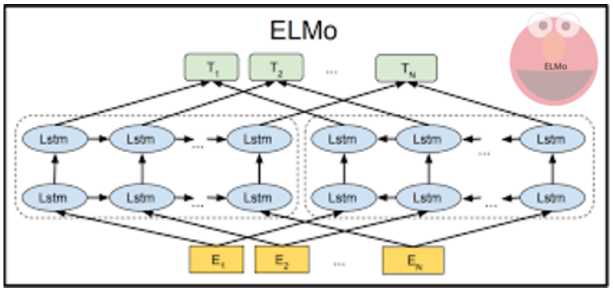

Figure 2: Yes-no architecture

Given the complete dataset released by BioASQ, we performed the following steps:

Table 1: Grid search for yes-no questions

Parameter	Values
Hidden layers	1, 2
Hidden units	50, 100, 150
Act. Function	relu, tanh
Learning rate	$1 \cdot 10^{-5}, 1 \cdot 10^{-6}, 1 \cdot 10^{-7}$
Optimizer	Adam, RMSProp
Pool size	1, 4, 8
Batch size	None, 8, 32

- Firstly, we filtered the questions by type, selecting those requiring a Yes/No type answer;
- Secondly, we pre-processed both questions and snippets by removing spurious lines and tokenizing them.
- Next, we got a vector-based representation of both questions and relevant snippets by means of ELMo. Such representation constituted the model input.
- Finally, a multi-layer perceptron model has been built. Its input are the vectors of the previous point, while the output consists in a single-neuron layer, one for each class.

For the training of the (fully connected, feed forward) Neural Network, we considered a further pre-processed version of the same dataset, differing for its shape: in the challenge dataset, snippets are provided in an aggregate fashion (i.e. $\langle question_i, \{snippets\}_i \rangle$), while we separated the single snippets provided together with the questions, obtaining inputs of the form $\langle question_i, snippet_{i,j} \rangle$. While in the former the target balancing was 80:20 (704-177), the latter consisted of 10285 'yes' and 1691 'no', i.e. 86:14 ratio. After some tests, we run all our experiments with the first version of preprocessed data (i.e. multi-snippet). In order to balance such ratio, we enriched training data with the similar data released in the previous versions of the challenge (5th, 6th and 7th editions). In this way, we got 182 additional no-targeted sets of snippets, corresponding to 3351 single snippets.

In order to train our model, we exploited the hold-out technique, using a 75% of the development dataset to train the model and the other 25% to validate it. As loss function we used the BinaryCrossentropy. Before performing the final grid search 1, we firstly performed some tests in order to find the most interesting hyperparameters. In particular, we observed that having more than 2 hidden layers didn't lead to better performance. Moreover, an increment in the learning rate led to a less stable solution, while a lower learning rate brought to a slower model learning.
Hyperparameters of the performed grid search included learning rate, batch size, hidden layers, hidden units among others. Their ranges are shown in table 1.

[3] While for list questions participating systems had to return a list of entity names constituting the answer, for factoid questions a list of entity names was required.

No post processing has been required in order to get the result of this type of questions.

Factoid

The model adopted for factoid questions is based on the pretrained BioBert, since it led Yoon et al. to considerably good results [10]. We adopted it for a two-fold goal: firstly, we availed of it in order to tokenize the inputs and create the embeddings. Then, we trained it for the question answering task. In order to optimize this last task, we fine-tuned it by means of the implementation of two softmax layers (see figure 3).

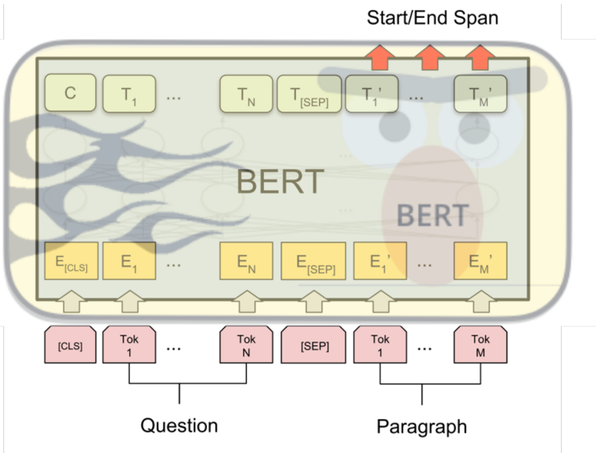

Figure 3: Factoid architecture

They respectively represent the probability for the tokens to be the starting and the ending token of the answer. Consequently, the answer provided by such model is the substring included within the tokens that maximize the two softmax probabilities (i.e. starting and ending indices).

Among the points addressed during the preprocessing phase, we highlight the variable length of the input phrases, since the maximum input length supported by BioBert is 512. For it, we adopted a padding-based solution for the inputs with lower length, while the others inputs have been discarded. Indeed, snippets have been separated by means of two types of markers: *CLS* (at the beginning of each question), *SEP*, which separated the snippets each other and *PAD*, representing the padding marker. Next, depending on the *max_length*, we added a padding within each tokenized snippet, so that every snippet had the same length with respect to the others. Concerning the training, an observation is in order with respect to the inputs: since we separated the set of snippets (in order to provide the model single $\langle question, snippet \rangle$ pairs), several pairs that didn't contain the answer in its snippets existed. In this cases, such snippets didn't provide useful information to the classification. Thus, we ignored them, avoiding these inputs to take part of the training phase.

Table 2: Grid search for factoid questions

Parameter	Values
Max_length	200, 300
Batch_size	5, 15, 30
Learning rate	$5 \cdot 10^{-6}$, $1 \cdot 10^{-6}$, $5 \cdot 10^{-7}$, $1 \cdot 10^{-7}$

The parameters adopted within the grid search have been *max_len*, *batch_size*, *epochs*, *learning_rate*. A complete view of the performed training and evaluations is reported in Table 2, in which the parameter of the grid search are illustrated. Before setting them, we performed several trainings involving different ranges of parameters values. Next, we run the grid search for more restricted ranges on the basis of the initial results. A post-processing phase has also been required in order to reconstruct the answer starting from the most likely initial and ending indices of the provided input.

Concerning the *max_len* parameter, we intend it to represent the length of the input, made up of both question and snippet. Every input exceeding such length has been discarded in the training phase. Thus, although the maximum supported token length of Biobert is 512, the selected values for such parameter (i.e. 200 and 300) have been chosen after considering the trade-off between the execution time and the amount of questions remaining after the elimination of the exceeding inputs from the training set. Indeed, in the two cases we lost respectively 48 and 12 inputs, getting respectively a 2x and a 2.5x speedup.

In order to select the best hyper-parameter configuration, we firstly run the grid search considering the Sparse Categorical Cross Entropy, without getting into account the evaluation metrics adopted in the challenge [4].
A total of 24 models have been trained within the grid search: initially, we noticed that higher learning rate values led to overfitting, while lower ones led to undesired, slow model learning. Thus, the best learning value has been represented by an intermediate value with respect to the ones adopted at the beginning, that is $5 \cdot 10^{-7}$. Contrarily, *max_len* parameter did not show significant impact on the model performance. The best configuration obtained is reported in Section 5. Furthermore, we performed further trainings until we reached the minimum loss value within the validation set. Such value has been reached with 20 epochs.
For the evaluation of the model we availed of the HoldOut technique on the 80% of the original training set, while the

[4] Due to hardware limitations, it hasn't been possible to run every configuration of Table 2. For instance, running the training with $max\_len = 300andbatch_size = 30$ lead to GPU errors.

remaining 20% has been used for test. We divided the former into training e validation with a 85:15 ratio, testing our model through the latter.

List questions

The model used for list questions is similar to the one used for factoid questions. It is composed by BioBert trained on SQUAD and two softmax layers for the fine-tuning, to identify the start and the end of the response. The stages that vary with respect to the factoid model are pre-processing and post-processing. In fact, a list of possible answers is required for every question. In the pre-processing phase, we split each sample into single snippets and we replicated a it as many times as the number of the contained answers. In the post-processing, the difference consists in merging the answers from single snippets such as factoid post-processing. Although we already developed all the phases, we still need to perform the grid search. Nevertheless, we expect similar results compared to the ones obtained for factoid questions, since we exploit the same architecture.

5 RESULTS

A premise is necessary in order to understand the results provided in the previous sections. Indeed, they have been obtained by splitting the given training set into further train and test sets, using the latter to get the evaluation metrics for each type of question. Contrarily, the systems participating in the challenge availed of the batch tests released by BioASQ throughout the challenge period. Consequently, two considerations are in order while comparing results. On one hand, ours may be more optimistic than the ones obtained by the challenge systems, since we use our self-calculated metrics. On the other hand, we could not exploit the portion of the released training set that we used as test. Assuming that such advantages and disadvantages compensate each other, we obtained satisfactory results, comparable to the ones of [10] and [14].

Yes-no results

In order to evaluate our model we used different metrics: we employed accuracy and different kind of f1-score[5]. In this way we could compare our result against the ones in [10] and [14], that evaluate their model exploiting the same metrics. However, as described in Section 5, the results reported in Table 5 do not refer to the same test set. Nonetheless, if we consider the average of these results or every single batch, we can state that we have close results, in terms of both accuracy and F1-score. The lower results in F1-score "No" is due the unbalanced dataset, as stated in section 4. Indeed, in order to increase the F1-score on the "No" class we had to oversample the No class by using the previous versions of the challenge. Final model hyperparameters are reported in Table 3. The training has been stopped according to an early stopping fashion, guided by the validation loss.

Table 3: Final model yes-no

Learning rate	$1 \cdot 10^{-5}$
Epochs	89
Batch size	*Full batch*
NN Hidden layers	2
NN Hidden units	150
Activation function	*ReLu*
Optimizer	*RMSprop*
Pool size	1
Train loss	0.0239
Validation Loss	0.2231

Factoid results

Our system is evaluated using some metrics provided by BioASQ guidelines. In particular:

- Strict Accuracy
- Lenient Accuracy
- Mean reciprocal rank (MRR)[5]

The description of those metrics can be found at [5]. We select our final parameter with the model selection. The final hyperparameters are reported in Table 4.

Table 4: Final model factoid

Learning rate	$5 \cdot 10^{-7}$
Epochs	20
Batch size	30
Max Len	200
Optimizer	Adam
Train loss	0.5085
Validation Loss	0.8359

The results obtained are similar to those of competing systems. In Table 6 are reported the comparisons between our system and the best system of every batch. As mentioned, our test is different with respect to those proposed by the comptetitors systems. Nevertheless the results are in the same range.

6 CONCLUSIONS

We designed, implemented, and evaluated different question answering systems based on state-of-the-art models. In particular, two of them provided results comparable to the

[5] Further details about evaluation measures can be found at http://participants-area.bioasq.org/Tasks/b/eval_meas/

Table 5: Results yes-no

	Batch 1	Batch 2	Batch 3	Batch 4	Batch 5	Batch avg	Our system
Acc.	0.8800	0.9444	0.9032	0.8462	0.8529	0.8853	0.8760
F1 Yes	0.9091	0.9630	0.9091	0.8571	0.8571	0.8991	0.8897
F1 No	0.8235	0.8889	0.8966	0.8333	0.8485	0.8582	0.8585
Macro F1	0.8663	0.9259	0.9028	0.8452	0.8528	0.8786	0.8741
Best	KoreaUniv-5	KoreaUniv-1	KoreaUniv-1	KoreaUniv-5	dice-c-1.0		

Table 6: Results factoid

	Batch 1	Batch 2	Batch 3	Batch 4	Batch 5	Batch avg	Our system
Lenient acc.	0.3750	0.2800	0.3214	0.5588	0.5625	0.41954	0.51798
Strict acc.	0.5938	0.4400	0.5357	0.7353	0.7188	0.60472	0.6330
MRR	0.4688	0.3533	0.3970	0.6284	0.6354	0.49658	0.56366
Best system	Umass_czi_1	KoreaUniv-DMIS-4	FudanLabZhu2	FudanLabZhu5	Umass_czi_3		

ones of the participating systems. Although the discussed differences in terms of evaluation of the systems, we retain such results satisfactory. Furthermore, the developed project allowed us to understand, learn and experiment the methodologies underlying the QA systems, i.e. transformers and language representation models. It goes without saying that a test of the QA system providing answers to list should be performed.

Future further experiments may involve the comparison with other Bert-based models trained on medical corpora. Another interesting task on our model may be represented by the ablation test, by means of which the most significant part of the input could emerge.

As a final consideration, we think that a focus on the dropout technique could have improved the performance: instead, due to our limited resources, we preferred to pay attention to the other described hyper-parameters.

REFERENCES

[1] Dan Moldovan and Mihai Surdeanu. On the role of information retrieval and information extraction in question answering systems. In Maria Teresa Pazienza, editor, *Information Extraction in the Web Era*, pages 129–147, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[2] Google. Google ngram viewer. http://books.google.com/ngrams/datasets, 2012.

[3] George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artieres, Axel Ngonga, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16:138, 2015.

[4] Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. Question answering in webclopedia. *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, 03 2001.

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

[6] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[9] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. 02 2018.

[10] Wonjin Yoon, Jinhyuk Lee, Dong hyeon Kim, Minbyul Jeong, and Jaewoo Kang. Pre-trained language model for biomedical question answering. *ArXiv*, abs/1909.08229, 2019.

[11] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, feb 2020.

[12] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *ArXiv*, abs/1606.05250, 2016.

[13] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don’t know: Unanswerable questions for squad. *ArXiv*, abs/1806.03822, 2018.

[14] Michele Resta, Daniele Arioli, Alessandro Fagnani, and Giuseppe Attardi. Transformer Models for Question Answering at BioASQ 2019. Technical report.

[15] Qiao Jin, Bhuwan Dhingra, William W. Cohen, and Xinghua Lu. Probing biomedical embeddings from language models. *ArXiv*, abs/1904.02181, 2019.