MindLab Neural Network Approach at BioASQ 6B

Andrés Rosso-Mateus, Fabio A. González
MindLab Research Group
Universidad Nacional de Colombia
Bogotá, Colombia
aerossom,fagonzalezo@unal.edu.co

Manuel Montes-y-Gómez
Laboratorio de Tecnologías del Lenguaje
INAOE
Puebla, Mexico
mmontesg@inaoep.mx

Abstract

Biomedical Question Answering is concerned with the development of methods and systems that automatically find answers to natural language posed questions. In this work, we describe the system used in the BioASQ Challenge task 6b for document retrieval and snippet retrieval (with particular emphasis in this subtask). The proposed model makes use of semantic similarity patterns that are evaluated and measured by a convolutional neural network architecture. Subsequently, the snippet ranking performance is improved with a pseudo-relevance feedback approach in a later step. Based on the preliminary results, we reached the second position in snippet retrieval sub-task.

1 Introduction

The development of methods that contribute to bypass the manual checking of candidate documents, is playing an important role in the closed domain information access and will be the next step in information retrieval systems (Zadeh, 2006). The number of published documents grows continuously and pertain to a large variety of topics. More than 3000 articles are indexed every day in biomedical journals (Tsatsaronis et al., 2012), making it harder for patients and physicians to access valuable information. The produced data needs to be mined in order to have a positive impact on public health, although it also represents a challenge.

The Question Answering (QA) paradigm can help to retrieve concise information in a natural way, given the precise answer and the supporting passages for any information need. The research in QA has been pulled by organizations and challenges that encourage academic community to develop new systems and methods to tackle this complex task.

One of the most important challenges is BioASQ, focused on indexing and question answering tasks over biomedical articles (Tsatsaronis et al., 2015).

In this work, we describe our first participation in the sixth edition of the BioASQ challenge. We participated in task B, which is composed of two phases.

- Phase A: Given a question the system must return relevant concepts (from designated terminologies and ontologies), relevant documents (from PubMed articles baseline (pub)), relevant snippets (extracted from articles), and relevant RDF triples (from designated ontologies) (Tsatsaronis et al., 2015).
- Phase B: Given a question and a set of relevant articles and snippets. The system must provide an exact answer (e.g., named entities) and ideal answers (summaries) (Tsatsaronis et al., 2015).

BioASQ challenge rules allow teams to participate in any of the two phases, and also send results for any or all of the sub-tasks in the desired phase. We chose **Phase A** for our first participation, and we submitted results for (1) document retrieval and (2) snippet retrieval.

2 Methods

2.1 Model Architecture

The whole system is composed of two main modules as shown in Figure 1. A document retrieval module searches the PubMed Baseline Repository (MBR) (pub) for relevant articles, and a fine-grained information retrieval model to identify the 10 most relevant snippets. For document retrieval we used Elastic Search (ES) engine (Gormley and Tong, 2015) with BM25 as relevance ranking function (Agichtein et al., 2006). To improve

Proceedings of the 2018 EMNLP Workshop BioASQ: Large-scale Biomedical Semantic Indexing and Question Answering, pages 40–46
Brussels, Belgium, November 1st, 2018. ©2018 Association for Computational Linguistics

the performance we added to the index the title, abstract and concepts for all the documents. When a search is performed, all fields are compared against the search query.

Most related documents are analyzed in depth. We split the documents into sentences and those sentences feed the snippet retrieval stage. We process the snippets with a Convolutional Neural Network (CNN) to obtain a semantic similarity relevance score.

Finally, the scored snippets are sorted in descending order and the 10 with the highest scores are selected. The documents are re-ranked based on a standardized linear combination between Elastic Search score and the average of their snippets scores. The 10 most related documents and snippets were submitted to BioASQ server.

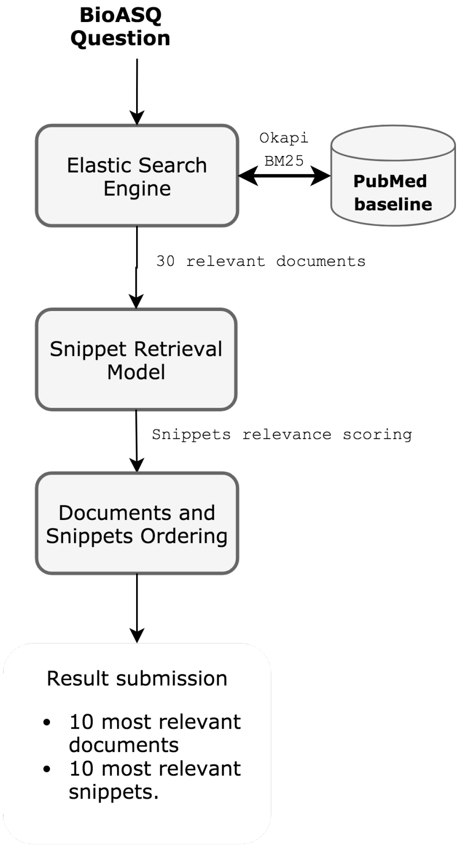

Figure 1: BioASQ Model Diagram

A detailed description of the model will be presented in the following sections.

2.2 Document Retrieval

Question answering systems make use of document retrieval methods to provide relevant documents that could contain the answer to a user's question.

The document retrieval system affects the question answering method effectiveness: if a retrieval system does not find relevant documents for a question, the later stages will inevitably fail.

In the BioASQ challenge the document retrieval system has to index approximately 27 millions medical articles. This huge amount of data makes it necessary to have a high-performance platform. We used Elastic Search (ES), a standalone search engine written in Java that stores the related data in a sophisticated format optimized for language based query searches (Gormley and Tong, 2015). ES is also easily scalable and comes with a default configuration that makes the whole learning process easy.

Elastic Search uses by default BM25, which is an improvement of TF-IDF ranking function that takes into account the length of documents and queries.

2.3 Snippet Retrieval

The main assumption of the snippet retrieval model is that the question and the answer are semantically related based on their terms. So the question-answer inter-correlation is given by the relationship between their component terms.

The proposed method has two stages. The first one (training phase) has the objective to learn the similarity patterns between question-answer pairs. In the second stage (prediction and re-ordering) the similarity model is used to obtain the first ranking between question and answer pairs, then a reordering is carried out using pseudo-relevance feedback based on the terms from the most related answer in the first ordering. The whole process is depicted in Figure 2.

The training phase is carried out to obtain the similarity model, then this model is used in the testing phase to rank the question-answer pairs. During training: (1) question-answer pairs (QA-pairs) are pre-processed, (2) the similarity matrix between QA-pairs terms is calculated, and (3) a convolutional neural network model is trained to predict the relevance of the answer to the question. Once the model is built it can be used to predict the rank of candidate answers. At testing time, for a particular question, the model is applied to predict the relevance score of the set of candidate answers, (4) answers are ranked according to their scores, (5) answers are re-ranked according to their similarity with the highest ranked answer at step (4), producing a new ranking of the answers.

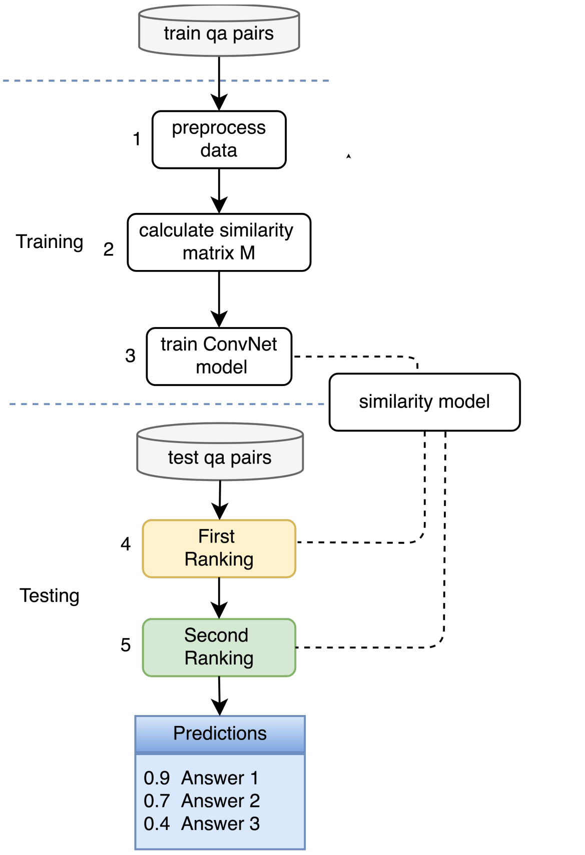

Figure 2: Two-Step Similarity Scoring Model Architecture

2.3.1 Step 1. Preprocess Data

Questions and candidate answers are processed using: tokenization to delimit terms; lowercasing to standardize the terms; POS-tagging, using the NLTK POS-tagger (Bird, 2006), to extract syntactical information that will be used in salience weighting; and transforming terms to a word2vec vector representation (Mikolov et al., 2013), to make possible their semantic similarity comparison.

2.3.2 Step 2. Calculate Similarity Matrix

The similarity matrix M represents the semantic relatedness of the i-th question term and the j-th answer term according to a similarity measure. Each element $M_{i,j}$ of this matrix is a composition of a similarity score and a salience score as described by Eq. 1.

$$M_{i,j} = scos(q_i, a_j) * sal(q_i, a_j) \tag{1}$$

2.3.3 Similarity Score

The similarity score for question-answer pair terms (q_i, a_j) is calculated using cosine similarity between their word2vec vectors as indicated by Eq. 2.

$$scos(q_i, a_j) = 0.5 + \frac{q_i \cdot a_j}{2 \left\| q_i \right\|_2 \left\| a_j \right\|_2} \tag{2}$$

In the case that there does not exist the word2vec representation for one of the terms, their similarity is measured based on their distance in Wordnet. In particular, we use as similarity measure the edge distance between the first common concept related with q_i and a_j (Wu and Palmer, 1994). If there is not a common concept between the terms, then we calculate the Levenshtein distance between the words (Levenshtein, 1966), defined as the number of operations (insertions and eliminations of characters) needed to transform q_i to a_j.

2.3.4 Salience Weighting

As not all terms are equally informative for measuring text similarities (Liu et al., 2009; Dong et al., 2015), we consider weighting the terms from the question and the answer based on part of speech functions: verbs, nouns, and adjectives are considered to be the most relevant. We model this information through a salience score.

The salience score is calculated as follows. If both terms are relevant then their score is 1. If only one of the terms is important then the score is 0.6, in the case none of them is relevant the score is 0.3. The salience function is defined in the Eq. 3.

$$sal(q_i, a_j) = \begin{cases} 1 & if\ imp(q_i) + imp(a_j) = 2 \\ 0.6 & if\ imp(q_i) + imp(a_j) = 1 \\ 0.3 & if\ imp(q_i) + imp(a_j) = 0 \end{cases} \tag{3}$$

Where $imp(q_i)$ and $imp(a_j)$ are the evaluation of importance weighting function for every question and answer term. The related function returns 1 if the term is a verb, noun or adjective, otherwise, returns 0.

Finally, we sort the calculated matrix M leaving the most related terms in the top left cell, and if the number of rows or columns exceeds 40, the remaining data is truncated. This step provides an invariable representation of the similarity patterns that can be exploited by the convolutional network.

2.3.5 Step 3. Convolutional Model

Convolutional neural networks (CNN) are a popular method for image analysis thanks to their ability to capture spatial invariant patterns. In the proposed method, they play a similar role, but instead of receiving an input image the CNN receives the similarity matrix M. The hypothesis is that it will be able to identify term-similarity patterns that help to determine the relevance of a question-answer pair. Patterns identified by the CNN are sub-sampled by a pooling layer. The output of the pooling layer feeds a fully-connected layer. Finally, the output of the model is generated by a sigmoid unit. This output corresponds to a score, **simScore**(q, a), that can be interpreted as a degree of relatedness between the question q and the answer a.

The architecture of the convolutional model is depicted in Figure 3.

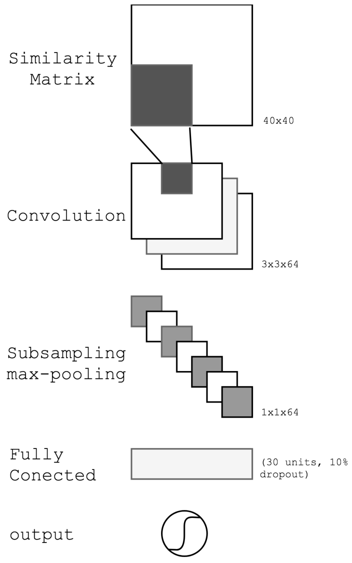

Figure 3: Convolutional Neural Network Model

2.3.6 Step 4 and 5. Two Ranking Stages

During the testing phase, a new query along with candidate answers are presented to the method. The candidate answers $(a_1, a_2, ..., a_k)$ are ranked using the CNN model producing the first rank of them. Based on the premise that the first candidate answer, a^*, is expected to be highly correlated with the question q, a second score, $simScore(a^*, a_k)$, is calculated by comparing each candidate answer with the highest ranked answer. A new ranking is calculated by using a new score corresponding to a linear combination of the first and second score as is shown in Eq. 4.

$$\begin{aligned} finalScore(q, a_k) = \\ (1 - \alpha) * simScore(q, a_k) + \\ \alpha * simScore(a^*, a_k) \end{aligned} \tag{4}$$

As we are introducing a weighting term α to scale the second score, we calculated this term based on the exploration carried out in a validation partition, which gives 0.32 as the optimal value.

This strategy promotes candidate answers which share similar terms with the highest ranked answer. This is a strategy analogous to pseudo-relevance feedback in information retrieval (Riezler et al., 2007), where the original query is extended with terms from the highest ranked documents.

2.4 Experiments

We indexed the full data of 2017 PubMed baseline in ElasticSearch engine (ES) version 6.2.2 with the default configuration. The number of processed files were 928 and the total number of medical articles was 26,759,399. For each article, we extracted the title, MESH concepts and abstract to be indexed. The indexing time was around 18 hours in an Intel Xeon processor Intel(R) at 2.60GHz with 82 GB RAM and GeForce GTX TITAN X.

The training was done with the question and answer pairs from 2016, 2017 and 2018 BioASQ Task B training data-set. The total number of question-answer pairs used were 124,144. The obtained data-set was very unbalanced, only 18% of the total number of pairs are labeled as an answer. To balance the data-set, the sample extraction in training phase is done with the same number of positives and negative samples, this strategy is also applied in the validation phase.

The model training was done using RMSprop optimization algorithm with 256 samples in mini-batch and the defined loss function is binary cross entropy. The number of maximum epochs was set to 500. In each epoch, we evaluate MAP and MRR, and after 20 epochs without any improvement in MAP metric, we apply early stopping to avoid over-fitting.

2.4.1 Model parameters

The model hyper-parameters were tuned using hyper-parameter exploration. The parameters chosen are listed next.

- **Convolution Parameters:** The number of convolutional filters used are 64, width 3 and length 3, the stride used is 1 without padding.
- **Convolution Activation Function:** After a convolutional layer, it is useful to apply a nonlinear layer (Goodfellow et al., 2016). We tested different activation functions and RELU gave us the best performance.
- **Pooling Layers:** For the pooling layer, we used max pooling.
- **Dropout Layer:** We add a dropout layer as a regularization strategy (Srivastava et al., 2014), setting the parameter in 10%.

Finally, the number of parameters to learn in our model is not very high (3,198) compared with other Convolution Neural approaches used in similar tasks (Question Answering) which are in order of millions and hundreds of thousands (Severyn and Moschitti, 2015; He and Lin, 2016)

2.5 Model Tuning

In this section, we will describe the strategy to improve the overall performance of our system. The metrics were calculated over the training dataset released by BioASQ for the 6th version.

- Mesh concept indexing: Document retrieval is mainly based on Elastic Search keyword matching evaluation with BM25 ranking function. We used a cross-fields query approach which looks for each term in the title, abstract and concepts indexed fields. Considering the retrieval of 10 most related documents, the performance using cross-fields approach were (Recall = 0.24, MAP = 0.19) while not using this were (Recall=0.278, MAP= 0.221).
- Word representation: The choice of a good word representation is important to generate a semantically good model where relations between terms or sentences are more easy to establish. We tested our system using different pre-trained word2vec models and the best representation was the skip-gram model provided by NLPLab, which is trained on Wikipedia and PubMed abstracts (Moen and Ananiadou, 2013). The MAP score in the snippet retrieval sub-task improved from 0.126 to 0.142.
- Training dataset generation: The training corpus was generated with questions and answer passages extracted from 2016, 2017 and 2018 BioASQ training datasets. We tested different rates of negative samples (passages in related documents that does contain the answer) in order to increase the negative sample coverage. This assumption is based on the hypothesis that it is not easy to determine that a related snippet does not contain the answer. With a higher negative sample generation, these cases are more common, and the method can learn a better discriminant function. The rate that experimentally achieved the best results considers using 10 negative samples per 1 positive sample. The MAP score in snippet retrieval sub-task, improved using 6b training partition from 0.142 to 0.151.
- Document re-ranking: After obtaining the similarity scores for snippets and the initial Elastic Search BM25 score for documents, the scores are combined as follows, eq. 5.

$$\begin{aligned} doc\_score(q, d_k) = \\ (1-\alpha) * es\_score(q, d_k) + \\ \alpha * avg(sim\_score(q, doc_k\_snippet\_j)) \end{aligned} \tag{5}$$

  where, $avg(sim\_score(q, doc_k\_snippet\_j))$ is the averaged similarity score between snippets of $document\_k$ and the query q. The calculated score is used to return the final list of documents. The parameter for the linear combination α, is calculated in evaluation step ($\alpha = 0.09$). Experimental results show that the contribution of Elastic Search score is higher (0.91). The improvement in document retrieval metrics was not significant but was around a 0.1 point in MAP and RECALL.

3 Results and Discussion

In this section, we present the preliminary results for the sixth version of BioASQ challenge in task

B phase A. The first sub-task is to retrieve the most related articles based on a question posed in natural language. The second one is to retrieve the snippets that have more correlation with the question in order to use them to compose an answer. The answer composition is carried out in phase B, which was not the scope of our participation.

3.1 Document Retrieval

The results shown in the Table 1 reveal, that our ES document retrieval implementation did not have a good performance, the recall obtained is low in all the batches. In the first batch, we had a technical issue that corrupted the results, it also happened for snippet retrieval. The best result was obtained in batch 3 (Recall = 0.49), the team leader in this batch reached 0.56, an important difference. As it was mentioned before, document retrieval is very important for snippet retrieval, it is the first information filter and it feeds the method to rank their snippets. Despite the low recall in this step, we will see in the next section that snippet retrieval scores are very promising.

Batch	**Document Retrieval**	
	Mean precision	Recall
	F-Measure	MAP
1	- -	- -
2	0.1150 0.1621	0.4685 0.0709
3	0.1320 0.1782	0.4984 0.0891
4	0.1240 0.1717	0.4467 0.0846
5	0.0890 0.1260	0.2961 0.0540

Table 1: Document retrieval results

3.2 Snippet Retrieval

In this stage, we analyzed in depth the returned set of documents from the previous method, and identify the text snippets that can answer the posed question.

Based on the evidence shown in Table 2, the snippet retrieval approach obtained a good performance. We could have had a better performance in snippet retrieval with a higher score in document retrieval, but it was enough to reach the second position in all the batches except the first one (due to the technical issue).

We can state that the proposed method exhibits a very competitive performance compared with other methods.

Batch	**Snippet Retrieval**	
	Mean precision	Recall
	F-Measure	MAP
1	- -	- -
2	0.1111 0.1416	0.2426 0.0938
3	0.1614 0.1877	0.2657 0.1344
4	0.1043 0.1306	0.2180 0.0980
5	0.0404 0.0542	0.1134 0.0475

Table 2: Snippet retrieval results

4 Conclusion

This work presents our first participation in BioASQ (task B phase A) document retrieval and snippet retrieval tasks. Our system was based on Elastic Search platform with the BM25 scoring function for document retrieval. For snippet retrieval, we presented a novel method based on a convolutional neural network with a pseudo-relevance-feedback re-ranking step.

The preliminary results are promising in snippets retrieval sub-task, where the proposed method reached the second position in all batches except the first one. This result gain in importance based on the fact that the chosen approach for document retrieval sub-task did not give good results.

The future work will be focused on improving document retrieval sub-task to feed the snippet retrieval method with a more complete (and higher quality) list of candidate answers. We will also work in a better question-answer pair representation with the incorporation of structured data sources for gain information.

Acknowledgments

COLCIENCIAS, REF. Agreement N°. 727, 2016 provided financial as well as logistical and planning support. Mindlab research group (Universidad Nacional de Colombia sede Bogotá) with the coperation of INAOE (Instituto Nacional de Astrofísica, Óptica y Electrónica) also provided technical support for this work.

References

Pubmed baseline repository.

Eugene Agichtein, Eric Brill, and Susan Dumais. 2006. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26. ACM.

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, volume 1, pages 69–72. ACL.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multi-column convolutional neural networks. In *ACL*, volume 1, pages 260–269.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT Press.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine*. " O'Reilly Media, Inc.".

Hua He and Jimmy J Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *HLT-NAACL*, volume 1, pages 937–948.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of human language technologies.*, volume 1, pages 620–628. ACL.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

SPFGH Moen and Tapio Salakoski2 Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan*, pages 39–43.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *ACL*.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *38th ACM SIGIR*.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

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 Artiéres, Axel-Cyrille Ngonga Ngomo, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(1):138.

George Tsatsaronis, Michael Schroeder, Georgios Paliouras, Yannis Almirantis, Ion Androutsopoulos, Eric Gaussier, Patrick Gallinari, Thierry Artieres, Michael R Alvers, Matthias Zschunke, et al. 2012. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *AAAI fall symposium: Information retrieval and knowledge discovery in biomedical text*.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *32nd Proceedings ACL*, volume 1, pages 133–138. Association for Computational Linguistics.

Lotfi A Zadeh. 2006. From search engines to question answering systems—the problems of world knowledge, relevance, deduction and precisiation. *Capturing Intelligence*, 1:163–210.