KSAnswer: Question-answering System of Kangwon National University and Sogang University in the 2016 BioASQ Challenge

Hyeon-gu Lee[1], **Minkyoung Kim**[1], **Harksoo Kim**[1*], **Juae Kim**[2]
Sunjae Kwon[2], **Jungyun Seo**[2*], **Jungkyu Choi**[3], **Yi-reun Kim**[3]
Kangwon National University, Chuncheon, Korea[1]
Sogang University, Seoul, Korea[2]
Intelligence Lab, LG Electronics, Korea[3]
{nlphglee, kmink0817, nlpdrkim}@kangwon.ac.kr[1]
{juaeKim, sj91kwon, seojy}@sogang.ac.kr[2]
{stanley.choi, yireun.kim }@lge.com[3]

Abstract

This paper describes a question–answering system that returns relevant documents and snippets (with particular emphasis on snippets) from a large medical document collection. The system is implemented as part of our participation to Phase A of Task 4b in the 2016 BioASQ Challenge. The proposed system retrieves candidate answer sentences using a cluster–based language model. Then, it re–ranks the retrieved top-*n* sentences using five independent similarity models based on shallow semantic analysis. The experimental results show that the proposed system is the first to find snippets in batches 2 (MAP 0.0604), 3 (MAP 0.0728), 4 (MAP 0.1182), and 5 (MAP 0.1187).

1 Introduction

BioASQ 2016 is the fourth annual BioASQ challenge as an established international competition for large–scale biomedical semantic indexing and question–answering, since 2013 (Tsatsaronis et al., 2015). The challenge consists of two tasks: Task 4a on large–scale online biomedical semantic indexing and Task 4b on biomedical semantic question–answering. Task 4b is further divided into two phases: Phase A and Phase B. In Phase A, participating systems are required to return a maximum of 10 relevant concepts, documents, snippets, and triples during five batches. Participation in Phase A can be partial, which means that it is acceptable to participate in only some of the batches and to return only relevant documents without snippets, triples, and concepts. This paper describes a questionanswering system of Kangwon National University and Sogang University submitted for Phase A of Task 4b in BioASQ 2016. The proposed system is focused on returning relevant documents and snippets (with particular emphasis on snippets).

2 Question-answering system based on sentence retrieval and re–ranking techniques

KSAnswer consists of two submodules: A retrieval model for finding candidate answer sentences from a large medical collection and a re–ranking model for determining the final answer among the retrieved candidate answer sentences.

2.1 Sentence retrieval model

Prior to indexing documents, KSAnswer first splits documents into a sequence of sentences using LingPipe (Baldwin et al., 2003). Then, it performs morphological analysis of the sentences and extracts content words (i.e., proper noun, common noun, verb, number, and so on) from the sentences. This is followed by stemming of content words except proper nouns using Porter Stemmer (Porter, 1980). Finally, KSAnswer uses the stemmed content words and the proper nouns as indexing terms.

For cluster–based sentence retrieval, KSAnswer generates two types of indexing units from the document collection comprising full data sets of PubMed journals: a sentence trigram unit and a document unit. The sentence trigram unit consists of an indexing target sentence and its context sentences (the previous and the next sentences) to address the lexical disagreement between a query and an indexed sentence. If a document consists of three sentences, KSAnswer generates three sentence trigrams (NULL–1st sentence–2nd sentence, 1st sentence–2nd sentence–3rd sentence,

*Corresponding author.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 45–49,
Berlin, Germany, August 7-12, 2016.

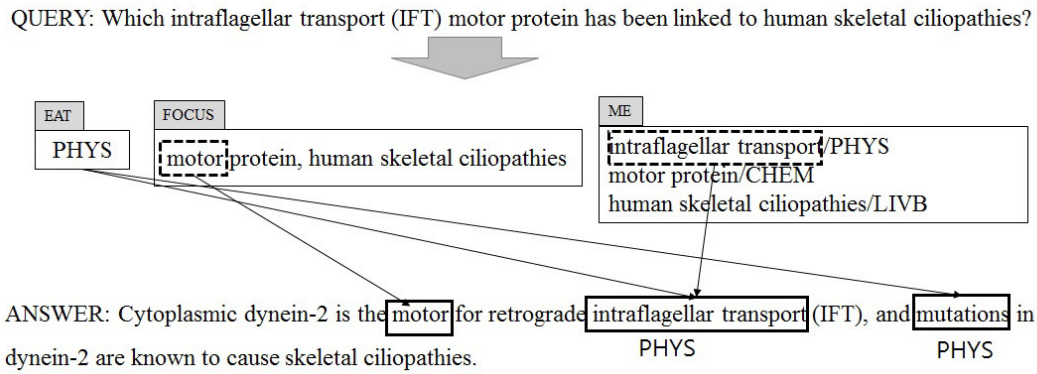

Figure 1: Relationship between a query and an answer sentence

2^{nd} sentence–3^{rd} sentence–NULL). The document unit consists of a title sentence and abstract sentences. The document unit assists in addressing the lexical disagreement between a query and a sentence trigram. Then, KSAnswer performs indexing of each unit and constructs two indexing databases using Lucene 4.0.0 (Białecki et al., 2012).

To rank candidate answer sentences, KSAnswer uses a cluster–based language model (Liu et al., 2004; Merkel et al., 2007), as shown in Eq. (1):

$$Sim_{IR}(Q,S) = \propto Sim_{tri}(Q,T) + (1- \propto) Sim_{doc}(Q,D) \quad (1)$$

where $Sim_{tri}(Q,T)$ is the similarity of the language model between the query Q and the sentence trigram T in the document D. Then, $Sim_{doc}(Q,D)$ is the similarity of the language model between the query Q and the document D. The weighting parameter $\propto$ has a value between 0 and 1. Finally, $Sim_{IR}(Q,S)$ returns similarities between the query Q and the indexing target sentence S, which is located in the middle of the sentence trigram T.

2.2 Sentence re–ranking model

Prior to re–ranking of candidate answer sentences, KSAnswer selects top–*n* retrieved sentences and normalizes their similarities, as shown in Eq. (2):

$$Sim'_{IR}(Q,S) = \frac{Sim_{IR}(Q,S) - m}{\sigma} \quad (2)$$

where m and σ are the average and standard deviation of similarity scores of top–*n* retrieved sentences, respectively.

KSAnswer re–ranks the top–*n* retrieved sentences using five independent similarity models, namely, $Sim_{SNT}(Q,S)$, $Sim_{EMB}(Q,S)$, $Sim_{EAT}(Q,S)$, $Sim_{FOCUS}(Q,S)$, and $Sim_{ME}(Q,S)$. $Sim_{SNT}(Q,S)$ is a similarity model between the query Q and the sentence S, which is located in the middle of the retrieved sentence trigram. $Sim_{EMB}(Q,S)$ is a similarity model between the sentence embedding of Q and the sentence embedding of S. The sentence embeddings are constructed by the sum of position–encoded word vectors in Word2Vec (so–called position encoding) (Sukhbaatar et al., 2015). $Sim_{EAT}(Q,S)$ is a similarity model between the expected answer type (EAT; a category name of expected answer) of Q and medical entity types (category names of medical entities) in S. $Sim_{FOCUS}(Q,S)$ is a similarity model between focus words (FOCUS; a clue word sequence to identify correct answers) in Q and content words in S. $Sim_{ME}(Q,S)$ is a similarity model between medical entities (MEs) in Q and medical entities in S. For example, in the sentence "Which drugs are utilized to treat eosinophilic esophagitis?", EAT, FOCUS, and ME are *[Chemicals & Drugs]*, *[eosinophilic esophagitis]*, and *[drugs, eosinophilic esophagitis]*, respectively. To obtain EAT, FOCUS, and ME, KSAnswer uses a sentence analyzer based on pattern matching and machine learning (Kim et al., 2004). The sentence analyzer extracts word chunks (generally noun phrases) from a query using lexico–semantic patterns. Then, it determines EAT and FOCUS by searching the syntactic chunks in MetaMap (Aronson et al., 2006). To obtain MEs, the sentence analyzer uses a special version of named entity recognizer based on Conditional Random Fields (CRFs), which is trained for medical documents (Abacha

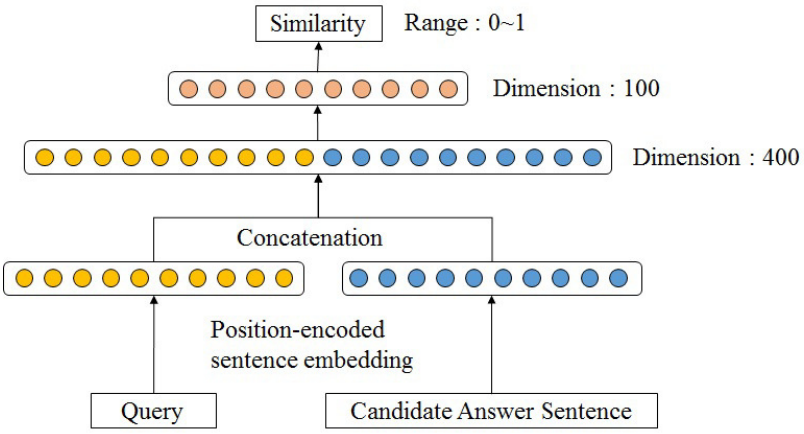

Figure 2: Vector-based similarity model based on a neural network

et al., 2012). The named entity recognizer extracts medical entities from a sentence and annotates them with predefined semantic categories. EAT and MEs use the same semantic categories as follows: ACTI (Activities & Behaviors), ANAT (anatomy), CHEM (chemicals & drugs), CONC (concepts & ideas), DEVI (devices), DISO (disorders), GENE (genes & molecular sequences), GEOG (geographic areas), LIVB (living beings), OBJC (objects), OCCU (occupations), ORGA (organizations), PHEN (phenomena), PHYS (physiology), and PROC (procedures). Figure 1 shows a relationship between Q and S at the view of EAT, FOCUS, and ME.

Eq. (3) shows the similarity scores between a query and each top–*n* retrieved sentences for re–ranking.

$$\begin{aligned} ReSim(Q,S) &= \alpha Sim'_{IR} \\ &+(1-\alpha)\{\beta Sim_{snt}(Q,S) \\ &+(1-\beta)\sum_{i=1}^{4}\gamma_i Sim^i_{sem}(Q,S)\}, \\ where\ 0 \le \alpha \le 1, 0 \le \beta \le 1, &\sum_{i=1}^{4}\gamma_i = 1 \end{aligned} \quad (3)$$

where $Sim^i_{sem}(Q,S)$ is the ith similarity model among $Sim_{EMB}(Q,S)$, $Sim_{EAT}(Q,S)$, $Sim_{FOCUS}(Q,S)$, and $Sim_{ME}(Q,S)$. Then, $\propto$, β, and γ are the weighting parameters set by experiments. The word–based similarity models (i.e., models for calculating similarities between words in Q and S), such as $Sim_{SNT}(Q,S)$, $Sim_{FOCUS}(Q,S)$, and $Sim_{ME}(Q,S)$, are calculated using the well-known Okapi BM25 (Robertson et al., 1999). Then, the category–based similarity model (i.e., a model for calculating similarities between category names in Q and S), $Sim_{EAT}(Q,S)$, is calculated using OR similarity of the Paice model (Paice, 1984), as shown in Eq. (4).

$$\begin{aligned} Sim_{EAT}(Q,S) &= \frac{\sum_{i=1}^{n}(r^{i-1}w_i)}{\sum_{i=1}^{n} r^{i-1}}, \\ where\ 0 \le r \le 1\ and\ w'_i s\ are& \\ considered\ in\ descending\ order& \end{aligned} \quad (4)$$

In Eq. (4), w_i is a TF·IDF value of the ith word in ME's of S that have the same semantic category with EAT of Q. Finally, the vector–based similarity model, $Sim_{EMB}(Q,S)$, is calculated using a feed–forward neural network with one hidden layer (Svozil et al., 1997), as shown in Figure 2.

The feed–forward neural network uses the sentence embedding vectors of Q and S as input values and uses a degree of relevance (from 0 to 1) between the two sentence embedding vectors as an output value. It is trained using gold standard answers as relevant snippets and by using top–*n* retrieved sentences except gold standard answers as irrelevant snippets.

3 Experiments

3.1 Experimental setting

We indexed the full data set of PubMed journals using Lucene 4.0.0. The number of document units was 12,208,342 and the number of sentence trigram units was 99,911,516. The language model parameters (μ values) for the document and sentence trigram units were set to 500 and 100, respectively. The weighting parameter $\propto$ in Eq. (1)

was 0.8. Then, the weighting parameters $\propto$, β, and γ_i in Eq. (3) were 0.5, 0.9, and 0.3, respectively.

3.2 Experimental results

In Phase A of Task 4b, our best submission was the first to find snippets in batches 2, 3, 4, and 5. In batch 1, we indexed the limit set of PubMed and achieved the second place in finding snippets. Table 1 shows the best performances of KSAnswer.

Table 1: Evaluation results of submitted runs

Batch	Document	
	Precision	Recall
	F1	MAP
1	0.0840(0.0840)	0.2258(0.1664)
	0.1065(0.1116)	0.0486(0.1223)
2	0.1675(0.1675)	0.4056(0.2758)
	0.2122(0.2084)	0.0949(0.1905)
3	0.1380(0.1380)	0.3946(0.2686)
	0.1786(0.1823)	0.0992(0.2095)
4	0.1720(0.1720)	0.5333(0.3470)
	0.2247(0.2300)	0.1257(0.2871)
5	0.1103(0.1103)	0.3752(0.2560)
	0.1546(0.1542)	0.0752(0.1742)
Batch	Snippet	
	Precision	Recall
	F1	MAP
1	0.0482(0.0418)	0.0952(0.1071)
	0.0534(0.0602)	0.0266(0.0738)
2	0.1021(0.0967)	0.1615(0.1930)
	0.1104(0.1288)	0.0604(0.1381)
3	0.0873(0.0823)	0.1208(0.1460)
	0.0886(0.1053)	0.0728(0.1440)
4	0.1504(0.1377)	0.2023(0.2653)
	0.1554(0.1813)	0.1182(0.2549)
5	0.0771(0.0773)	0.1272(0.1434)
	0.0798(0.1004)	0.0582(0.1187)

The parenthesized values are informal performances that are calculated using gold standard answers for each batch. In an additional experiment, we found that the degree of the sub-model importance in the re-ranking model is as follows: $Sim_{SNT}(Q,S) >> Sim_{EAT}(Q,S) > Sim_{FOCUS}(Q,S) \approx Sim_{ME}(Q,S) \approx Sim_{EMB}(Q,S)$

4 Conclusion

We proposed a question-answering system for finding candidate answer snippets from a large medical document collection. The proposed system retrieves candidate answer sentences using cluster–based language model. Then, it re–ranks top–*n* retrieved sentences using various similarity models based on shallow semantic analysis of sentences. In Phase A of task 4b, the proposed system showed excellent performance by being the first to find snippets in batches 2,3,4 and 5.

Acknowledgments

This research was supported by LG Electronics. It was also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2013R1A1A4A01005074).

References

Alan R. Aronson. 2006. Metamap: Mapping text to the umls metathesaurus. *Bethesda*, MD: NLM, NIH, DHHS, 1–26.

Andreas Merkel, and Dietrich Klakow. 2007. Comparing improved language models for sentence retrieval in question answering. *LOT Occasional Series 7*, 35–50.

Andrzej Białecki, Robert Muir, and Grant Ingersoll. 2012. Apache lucene 4. *SIGIR 2012 workshop on open source information retrieval*.

Ben Abacha, Asma, and Pierre Zweigenbaum. 2012. Medical question answering: translating medical questions into sparql queries. *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium.* ACM.

Breck Baldwin, and Bob Carpenter. 2003. LingPipe. *Available from World Wide Web: http://alias-i.com/lingpipe*.

Chris D. Paice. 1984. Soft evaluation of Boolean search queries in information retrieval systems. *Information Technology: Research and Development,* 3(1):33-41.

Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. 1997. Introduction to multi–layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems,* 39(1):43–62.

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 Artires, Axel–Cyrille N. 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.*

Harksoo Kim and Jungyun Seo. 2004. A high performance question-answering system based on a two–pass answer indexing and lexico-syntactic pattern matching. *IEICE Information and Systems,* Vol.E87–D (12):2855–2862.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program,* 14(3):130–137.

Sainbayar Sukhbaatar, Jason Weston, and Rob Fergus. 2015. End–to–end memory networks. *Advances in Neural Information Processing Systems.*

Stephen E. Robertson, Steve Walker, and M. Beaulieu. 1999. Okapi at TREC–7: automatic ad hoc, filtering, VLC and interactive track. *Nist Special Publication SP,* 253–264.

Xiaoyong Liu, and W. Bruce Croft. 2004. Cluster–based retrieval using language models. *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM.