Large-Scale Semantic Indexing and Question Answering in Biomedicine

E. Papagiannopoulou*, Y. Papanikolaou*, D. Dimitriadis*, S. Lagopoulos*,
G. Tsoumakas*, M. Laliotis**, N. Markantonatos** and I. Vlahavas*

*School of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece
**Atypon, 5201 Great America Parkway Suite 510, Santa Clara, CA 95054, USA

Abstract

In this paper we present the methods and approaches employed in terms of our participation in the 2016 version of the BioASQ challenge. For the semantic indexing task, we extended our successful ensemble approach of last year with additional models. The official results obtained so-far demonstrate a continuing consistent advantage of our approaches against the National Library of Medicine (NLM) baselines. For the question answering task, we extended our approach on factoid questions, while we also developed approaches for the document, concept and snippet retrieval sub-tasks.

1 Introduction

The BioASQ project (Balikas et al., 2014) aims to provide a challenge framework for researchers dealing with classification (semantic indexing) and natural language processing (question answering) tasks in the field of bio-medicine. The challenge, similar to the previous three years, is divided into two tasks: automated semantic indexing (4A) and question answering (4B).

In Task 4A participants are given a set of new, unannotated articles and are required to automatically predict the relevant MeSH terms for each one of them in a given time. For each article only the abstract along with some meta-information is provided (journal, year and title). This task is particularly difficult, as the MeSH taxonomy is comprised of a large number of labels ($\sim$ 27000), with the label set following a distribution similar to power-law. Furthermore the terms are subject to a significant concept drift along time.

Task 4B is divided into 2 phases, called A and B. In phase A participants are given a set of questions and must return the 10 most relevant documents, snippets, concepts (from designated ontologies) and RDF triples. In phase B participants are given the gold standard documents and snippets and must provide exact and ideal answers.

This paper discusses the approaches we developed for this year's BioASQ challenge. In particular, Section 2 discusses our semantic indexing algorithms, Section 3 our document retrieval system, Section 4 our concept retrieval method, Section 5 our snippet retrieval approach and Section 6 discusses our question answering approach. Final considerations and conclusions are drawn in Section 7.

2 Task 4A: Semantic Indexing

In this section we present the methods that we used for the semantic indexing task. We first provide the pre-processing pipeline and subsequently the methods employed.

2.1 Pre-processing

In this year's participation, we used the 1,050,000 most recent documents from the BioASQ 2016 corpus using as a training set the first 1 million articles and the last 50 thousand as a validation set. The motivation behind using the latest articles of the corpus, stems from the hypothesis that more recent chronologically articles will tend to follow more similar labels distributions to new articles that have to be predicted, compared to older ones. Pre-processing of the articles was carried out similar to previous years; the abstract and the title were concatenated, uni-grams and bi-grams were used as features, removing stop-words and features with less than five occurrences in the corpus. We used the *tf-idf* representation for the features. Also, zoning of the features belonging to the title and those equal to a MeSH label was performed

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 50–54,
Berlin, Germany, August 7-12, 2016.

by increasing the *tf-idf* value of features that belonged to the title by $log2$ and those being equal to a label by $log1.25$. The above features were used in order to train several multi-label learning models, described in the following section.

2.2 Methods

Our participation to this year's contest included several multi-label classifiers (MLC) that were combined in various ensembles. As in the previous year, we used the Meta-Labeler (Tang et al., 2009), a set of Binary Relevance (BR) models with Linear SVMs (both tuned and with default parameters) and a Labeled LDA variant, Prior LDA (Rubin et al., 2012). For the tuned SVM models, we used different values for the C parameter and handled class imbalance by penalizing more heavily false negative errors than false positive ones by adjusting properly the weight parameter (Lewis et al., 2004). This year, we additionally employed Fast XML (Prabhu and Varma, 2014) and HOMER-BR (Tsoumakas et al., 2008).

All the above models were combined in an ensemble, using the MULE framework (Papanikolaou et al., 2014). MULE is a statistical significance multi-label ensemble that performs classifier selection. The key idea is to combine a set of multi-label classifiers aiming to optimize a selected measure (for the purpose of this challenge, we are mainly interested in the micro-F measure) and validate this combination through a statistical significance test; McNemar's test. This way, each label of the multi-label problem is predicted with a specific component model, the one that (a) contributes to the greatest improvement to the evaluation metric of interest and (b) is validated from the statistical test to indeed produce the aforementioned improvement. If (b) does not hold, in other words if the component model's improvement is not statistically significant, we predict that label with the globally optimal model.

2.3 Results

Since at the moment of writing this paper there are not sufficient official results yet(only the a small part of documents of the first batch are annotated), in Table 1 we present the performance of the multi-label classifiers used in our ensembles, in terms of the Micro-F and Macro-F measures, for the training set (one million documents) and the validation set (fifty thousand documents) used throughout the challenge.

Table 1: Performance of the multi-label classifiers used throughout the BioASQ challenge semantic indexing task 4a, in terms of Micro-F and Macro-F. Training set size was 1,000,000 documents and test set size 50,000 respectively.

MLC	Micro-F	Macro-F
Meta-Labeler	**0.61936**	**0.57477**
Vanilla SVMs	0.58422	0.50080
Tuned SVMs	0.61365	0.54444
Labeled LDA	0.47399	0.39084
Fast XML	0.38053	0.28899
HOMER-BR (k=3)	0.59698	0.54972

3 Task 4B Phase A: Documents

Here we describe our document retrieval system. The system was written in Java. A variety of libraries have been used. The StAX Parser[1] for the input of XML files, the Stanford Parser[2] for natural language parsing and the GSON library[3] for output of JSON files. We build our system on open source Indri search engine from the Lemur Project[4].

3.1 Pre-processing of citations

We processed the full database of MEDLINE and extracted the citations that contained Title, Abstract and MeSH annotations. There are 14,938,869 documents.

3.2 Search Engine

We used Indri as our search engine. We normalized the text of all the processed citations and we inserted them to our search engine. No stemming or stop-words filtering has been done in order to avoid any distortion of bio-medical and other important terminology.

3.3 Question Parsing and Query

Our system processes and analyzes the input question before producing the final query. It removes any unwanted punctuation, it analyzes the question with the Stanford Parser and produces a bag of words. Finally, we form our query by combining the bag of words with the query language grammar of Indri.

[1] https://docs.oracle.com/javase/tutorial/jaxp/stax/api.html

[2] http://nlp.stanford.edu/software/lex-parser.shtml

[3] https://github.com/google/gson

[4] http://www.lemurproject.org/

3.4 Testing

We tested our system by using both the questions and the gold standard articles of the previous BioASQ challenges and the current challenge. We experimented with Indri's great variety of search terms and tried retrieving top-10, top-20 and top-50 documents. The table below provides the results of our experiments retrieving top-10 documents.

Table 2: Test results retrieving top-10 documents

Task	# questions	Precision	MAP
1b, 2b, 3b	940	0.279	0.141
4b TestSet 1	100	0.156	0.233
4b TestSet 2	100	0.230	0.198
4b TestSet 3	100	0.195	0.250
4b TestSet 4	100	0.235	0.321
4b TestSet 5	97	0.105	0.158

4 Task 4B Phase A: Concepts

We are working at the phase A task of returning a list of at most 10 relevant concepts from the designated terminologies and ontologies. The list is ordered by decreasing confidence. In our approach, we use MetaMap[5] and LingPipe[6] to detect the biomedical concepts and local ontology files (Disease ontology, Gene ontology, Jochem, Uniprot and MeSH) to retrieve the appropriate information. More particularly, we use RDF4J[7], a powerful Java framework for processing and handling RDF data of Disease ontology, Gene ontology, Jochem, and MeSH. This includes creating, parsing, storing, inferencing and querying over such data. Additionally, we use RDF4J's Lucene Sail that enables us to add full text search of RDF literals to find fast subject resources. As far as the Uniprot data are concerned which are not in obo format, we exploit them in XML format (not plain text that is recommended by the contest). Of course, Lucene indexing is necessary again. We present our methodology step by step:

1. The first step of our methodology is to remove stopwords from the given question. We use 2 stopwords lists: a basic list with 634 words and the Pubmed stopword list[8]. Then, we detect keywords using MetaMap and LingPipe. We give a boosting score to those concepts that come from MetaMap/LingPipe and a smaller score in any other word that appears in the question and MetaMap/LingPipe does not recognize it as biomedical concept.

2. Then, we expand the list with the candidate concepts exploiting the MeSH ontology (up to 15 candidate concepts, totally, enriching the list with ExactSynonyms). We retain two lists with candidate concepts: a list with all possible biomedical concepts for search in Disease ontology, Gene ontology, Jochem, and MeSH and a list that contains only proteins or genes for search in Uniprot XML data.

3. We search for each candidate term separately combining search in RDF4J's Lucene Sail index for fast detection of relevant terms and search in RDF4J RDF Repositories via SPARQL queries to filter the results which are returned as relevant terms by RDF4J's Lucene Sail index. More specifically, for the 4 ontologies we examine if the candidate term appears in properties: label, ExactSynonym, RelSynonym, Synonym, NarrowSynonym, BroadSynonym in order to add to Lucene score an additional boosting score and return the corresponding URI. If the candidate term does not appear in the above properties, then we just keep the Lucene score. Additionally, we exploit the properties (Positively/Negatively) Regulates in order to return the corresponding URI, too. Similarly, we conduct search in Uniprot data but instead of SPARQL queries, we use XPath, focusing in the following XML elements: fullName, shortName, alternativeName and innName.

4. Finally, we take the top 10 concepts with the biggest scores.

Here, we present experimental results on 2 different sets of questions (the sets belong to the training set of BioASQ contest).

Table 3: Results of our approach

# questions	Precision	Recall	F1	MAP
238	0.167	0.511	0.223	0.120
286	0.209	0.513	0.267	0.167

[5] https://metamap.nlm.nih.gov/

[6] http://alias-i.com/lingpipe/

[7] http://rdf4j.org/

[8] http://www.ncbi.nlm.nih.gov/books/NBK3827/table/pubmedhelp.T.stopwords/

5 Task 4B Phase A: Snippets

In order to extract relevant snippets to a query, we exploit our knowledge given by the ontologies we referred to in Section 4 (Disease ontology, Gene ontology, Jochem, Uniprot and MeSH). Briefly:

1. Detect keywords using MetaMap and LingPipe

2. Search for synonyms for each keyword in order to make query expansion. Consider we have K keywords and for each one we find a few synonyms, e.g. for i-th keyword, $i = 1, ..., K$, we detect N synonyms. Each synonym is denoted by $syn_{j_{key_i}}$, that is the j-th synonym (syn_j), $j = 1, ..., N$ of i-th keyword (key_i).
   Format of query after the expansion step: Suppose K=2, key_1 has N synonyms and key_2 has M synonyms, so the query is going to be the following:
   ((key_1 OR $syn_{1_{key_1}}$ OR $syn_{2_{key_1}}$ OR ... OR $syn_{N_{key_1}}$)
   AND
   (key_2 OR $syn_{1_{key_2}}$ OR $syn_{2_{key_2}}$ OR ... OR $syn_{M_{key_2}}$))
   The total number of the candidate concepts (i.e. keywords with their corresponding synonyms) should contain up to 15 concepts.

3. Retrieve top 100 relevant documents (use of Lucene index). More particularly, we are interested in their title, abstract and pmid.

4. Split titles/abstracts returned in step 3 into sentences.

5. Calculate sematic similarity between each one of the sentences and the (expanded) query using the semantic similarity measure described in (Han et al., 2013). (At this point, we experiment using clustering algorithms in order to select the sentences that are located in the same cluster with the query, regarding them as the most relevant snippets.)

6. Return the top 10 sentences that are more similar to our query according to the similarity measure.

6 Task 4B, Phase B: Exact Answers

We developed a system that extracts answers from factoid questions under a scoring mechanism. In our approach, we applied numerous measurements that rank the candidate answers based on their relations with the questions. Some of them were applied in our previous system, but we realized that were not enough to estimate the correct answer. Thus, we extended the previous scoring mechanism in order to include the measures describing below.

- *distance:* The words, being near to the LAT of the question into the snippets, it is more possible to be a candidate answer.

- *wordnet synonyms:* We strongly believe that words with many synonyms in wordnet are more likely to be used in common language rather than in biomedicine. Thus, they take a punishment according to the number of synonyms that they have.

Furthermore, in the previous work, the system selected some of the words of an article as candidate answers. It selected the words that were produced by MetaMap parsing. Although, the results of the previous system were promising in the BioASQ training set, in the BioASQ challenge were quite low. The system's failure was caused by the lack of candidate answers. That's why we decided to expand the set of candidate answers considering all words including in the related snippets of a question.

Finally, the specificity measure in our previous work changed because of the execution time. We had implemented that measure to count the number of instances of a candidate answer in all PubMed documents. Thus, we decided to seek the documents including the candidate answers with a document retrieval system. For each retrieving document, the candidate answer take a punishment.

Table 4: Results of factoid system

LACC	SACC	MRR
0.54	0.237	0.305

7 Conclusions

In this paper we presented the participation of our team in the BioASQ challenge 2016. Building on the successful approaches in the past three challenges, we further extended our line of work to

improve the performance of our systems. Additionally, our methodology for relevant concepts retrieval gives quite good results based on our evaluation in a variety of bio-medical questions that are provided by BioASQ's training set. Moreover, the semantic information from ontologies could be exploited for other tasks.

References

[Balikas et al.2014] Georgios Balikas, Ioannis Partalas, Axel-Cyrille Ngonga Ngomo, Anastasia Krithara, and Georgios Paliouras. 2014. Results of the bioasq track of the question answering lab at CLEF 2014. In *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.*, pages 1181–1193, july.

[Han et al.2013] Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Johnathan Weese. 2013. UMBC_EBIQUITY–CORE: Semantic Textual Similarity Systems. In *In Proceedings of the 2nd Joint Conf. on Lexical and Computational Semantics*. Association for Computational Linguistics.

[Lewis et al.2004] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397.

[Papanikolaou et al.2014] Yannis Papanikolaou, Dimitrios Dimitriadis, Grigorios Tsoumakas, Manos Laliotis, Nikos Markantonatos, and Ioannis P. Vlahavas. 2014. Ensemble approaches for large-scale multi-label classification and question answering in biomedicine. In *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.*, pages 1348–1360.

[Prabhu and Varma2014] Yashoteja Prabhu and Manik Varma. 2014. Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 263–272. ACM.

[Rubin et al.2012] Timothy N. Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. 2012. Statistical topic models for multi-label document classification. *Mach. Learn.*, 88(1-2):157–208, July.

[Tang et al.2009] Lei Tang, Suju Rajan, and Vijay K. Narayanan. 2009. Large scale multi-label classification via metalabeler. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 211–220, New York, NY, USA. ACM.

[Tsoumakas et al.2008] G. Tsoumakas, I. Katakis, and I. Vlahavas. 2008. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, pages 30–44.