LABDA at the 2016 BioASQ challenge task 4a: Semantic Indexing by using ElasticSearch

Isabel Segura-Bedmar, Adrián Carruana, Paloma Martínez
Computer Science Department, University Carlos III of Madrid
Avd. Universidad, 30, Leganés, 28911, Madrid, Spain
isegura,acarruan,pmf@inf.uc3m.es

Abstract

This paper describes the participation of LABDA team in the 2016 BioASQ Task 4a on large-scale online biomedical semantic indexing. Our approach is based on the use of the open source search engine ElasticSearch. Experimental results show that our approach achieves high recall while keeping processing time low. Although more work needs to be done to improve our results, we can conclude that ElasticSearch is a competitive and scalable system for indexing biomedical literature.

1 Introduction

Biomedical Natural Language Processing (BioNLP) has made great advances in the last decade thanks to different community-wide challenge evaluations, such as BioCreative (Krallinger et al., 2015), BioNLP shared tasks (Kim et al., 2011; Nédellec et al., 2013), i2b2 (Stubbs et al., 2015) DDIExtraction (Segura-Bedmar et al., 2011; Segura Bedmar et al., 2013), etc. While most of them have pursued the further development of research on informations extraction tasks, the BioASQ Challenge[1] focuses on biomedical semantic indexing and question answering fields.

Biomedical Semantic Indexing is to identify the MeSH categories that best describe a PubMed article and is a crucial task to facilitate literature search. This process is manually performed by human experts, thus becoming a costly, time-consuming and laborious task (Huang et al., 2011). Therefore there is an urgent need to explore automatic methods to support this task.

As in previous editions (Tsatsaronis et al., 2015; Balikas et al., 2015), BioASQ 2016 consists of two different tasks: large-scale online biomedical semantic indexing (Task 4a) and question answering (Task 4b). This paper describes our participation in Task 4a. The goal of the task is to automatically predict the most relevant MeSH labels for a given document. One of the major challenges of the task is to manage scalability due to the great amount of documents that have to be indexed. More than 750,000 articles were added in 2014 with a load of 2000-4000 documents per day.[2] Search systems such as ElasticSearch, an open source search engine, could be adequate frameworks to cope with this information overload problem.

To the best of our knowledge, this is the first work that addresses semantic indexing by using ElasticSearch. Due to the horizontal scalability provided by ElasticSearch, it is possible to index large collections of documents, as is the case of the Medline/PubMed database with more than 22 million citations to date. Our approach is to index the training set provided by the BioASQ organizers with ElasticSearch. Then, each document in the test set is translated into a query, that is fired against the index built from the training set, returning the most relevant documents and their MeSH categories. Finally, each MeSH category is ranked using a scoring system based on the frequency of the category and the similarity of relevant documents, which contain the category, with the test document to classify. Up to date at which we write this paper, no official definitive results have been published for any of our submissions yet. To evaluate our approach, we generated our own development set from a random sample of 1099 training documents. To avoid any potential bias, these documents were removed from the training set. Tested on this development set, our approach achieves a recall of 80.6%, precision of 45.4% and an F1 of

[1] http://www.bioasq.org/

[2] https://www.nlm.nih.gov/pubs/factsheets/medline.html

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 16–22,
Berlin, Germany, August 7-12, 2016.

56.3%. In comparison to the Medical Text Indexer (MTI) (Mork et al., 2013), which is considered the baseline system of the task, our system does not only provide an improvement of more than 1% in F1, but also has a much better time response (15 seconds per document) than the MTI system (30-45 seconds per document).[3]

The rest of the paper is organized as follows: related work is presented in Section 2. Section 3 presents a description of our method and the datasets used in this study. Then, we report and discuss some preliminary results of our approach in section 4. Finally, section 5 presents conclusion and future work.

2 Related Work

Semantic indexing of MEDLINE articles is a manual laborious task which could be helped by information technology. The objective is to tag an article with a set of MeSH categories, hence it is a multilabel classification problem.

The main challenge of this shared task is to work with MeSH, a big hierarchy that includes a controlled vocabulary composed of 15 root concepts, such as organisms and diseases, with more than 25,000 categories. Most of works restrict the scope of MeSH hierarchy using only a particular branch in the MeSH tree (for instance Heart Diseases) (Ruiz and Srinivasan, 2002), or a subset of tags, generally those appearing in the training collection (Yepes et al., 2015).

Current state-of-the art includes approaches whose general architecture comprises two differentiated phases: a first phase that obtains an initial set of MeSH categories that could represent the document to classify and a second phase that rerank these categories to select the top K that better fit the input document. In both phases different document features can be used; the most frequent feature model is the so called bag-of-words (where words could follow a ngram model or be a word, phrase, concept, etc. storing a value that represents its presence frequency in the document or any other model such as TF*IDF).

Doing a review of BioASQ previous editions (Partalas et al., 2013; Balikas et al., 2014; Balikas et al., 2015; Tsatsaronis et al., 2015), the main characteristics of participants systems are: approaches that use flat methods which consider each MeSH category independently of the others or hierarchical methods that take into account the MeSH tree structure; the machine learning techniques used to select the initial set of MeSH labels (SVM, logistic regression, K nearest neighbor, etc.); the word model (unigram, bigram, trigram); if Natural Language Processing (NLP) tools are included to preprocess documents (POS taggers, chunkers, syntactic parser); if domain specific resources are used (for instance, UMLS ontology or WordNet lexical database); if the system is built over a search-based platform (such as Lucene); if curator annotation guidelines are considered and the processing and storage requirements both in the definition of models to multilabel training and classification process.

In 2013 edition, the best systems (depending on the batch) were the Medical Text Indexer (MTI) (Mork et al., 2013) with a micro F measure of 0.5481 and the system AUTH (Tsoumakas et al., 2013) with a micro F measure of 0.578. The MTI system, which is considered the baseline system of the task, is based on a combination of Metamap indexing and Pubmed related citations to recognize MeSH concepts that then are clustered and ranked. AUTH system preprocessed the articles using the Stanford parser and bigram frequencies were extracted. The meta-labeler tool (Tang et al., 2009), which is based on SVM binary classifiers trained for each label present in a subset of training collection, was used to rank the labels and a regression model is used to predict the K top labels.

In 2014 edition, several systems outperformed the MTI baseline system (micro F measure of 0.547), the system of NCBI (Mao et al., 2014), with a micro F measure of 0.605 and the Antinomyra system (Liu et al., 2014), with a micro F measure of 0.619. The NCBI system selected the relevant MeSH labels for a given article from its k-nearest neighbor documents. This set was also extended with the MeSH labels proposed by the MTI system. Then, a learning-to-rank algorithm was used to sort the MeSH labels based on the learned associations between the article text and each MeSH label. This system also used SVM binary classifiers (trained for each MeSH label in the training data) to predict the MeSH labels in the test data. The Antinomyra system followed a similar approach but instead of using SVM classifiers it used a logistic regression method.

[3] https://www.nlm.nih.gov/mesh/MeSHonDemand.html

The winner in BioASQ 2015 (Liu et al., 2015) used a learning to rank approach that returns an ordered list of MeSH categories for each instance using a combination of binary classifiers, similar articles to the article to annotate, pattern matching between MeSH categories and title of the article as well as the prediction of the MTI baseline system. This system achieved a micro F measure of 0.615.

Concerning the annotation guidelines followed by curators, some works such as (Mork et al., 2013) make use of MEDLINE annotation guidelines to postprocess the ranking of MeSH categories. The overview of BioASQ 2013 systems (Tsatsaronis et al., 2015) suggests that it is difficult to know the utility of the is-a relations in the MeSH hierarchy due to human curators do not seem to follow the annotation guidelines concerning the use of most specialized tag.

Out of the scope of BioASQ forum, the approach described in (Rak et al., 2007), was based on association rule mining from the OHSUMED corpus (Hersh et al., 1994), which contains approximately 340.000 articles from 1987 to 1991 (the rules are a kind of information retrieval techniques where a set of words determine the class of the document). A more recent work (Yepes et al., 2015) analyzed different representations of articles based on lexical, syntactic and semantic information. This system was tested over a collection of 143,853 citations and 63 selected MeSH categories (those with at least 1,500 citations indexed). Application of NLP features do not exhibit good performance although combination of all features performs better than individual sets. Participants in BioASQ such as (Ribadas et al., 2014) achieved poor results when NLP techniques are included.

3 Method

The goal of the task is to automatically predict the most relevant MeSH categories for each article in a test set. The predictions should be compared to MeSH categories proposed by human curators. This section describes the method and data used in this study.

3.1 Data

The training data for the BioASQ task 4.a consist of PubMed articles that were manually annotated with MeSH terms by human curators. In addition to the new 2016 training dataset, the training datasets of the previous BioASQ challenges are available too. The main difference between those datasets is the version of the MeSH vocabulary that was used to annotate their articles. It should be noted that each year a new release of MeSH including updates of its structure (for example, 310 new MeSH Headings were added to MeSH in 2015) is published. Typically, articles are not re-indexed with the new MeSH terms.

The teams are permitted to use any resource to train their systems, however we only use the 2016 training dataset because the evaluation will be performed using the MeSH version 2016. There are two versions of the training data: (1) Training v.2016a with more than 12 million of documents, and (2) Training v.2016b with almost 5 million of documents from the pool of journals that the BioASQ organizers use to select the articles for the test data. This dataset was built using only journals with small average annotation periods. In both datasets, the average number of MeSH terms assigned to an article is 12-13.

In order for the teams to evaluate their systems, a new test set is available every Monday. Then, the teams can upload their results before the next 24 hours after the release. A total 15 test sets have been published, which are grouped in three different periods (batches). It should be noted that the articles used in the test datasets have not been annotated yet by human experts, and therefore, it is not possible to provide an immediate evaluation of the participant systems. This is an important inconvenience since there is no fast way to assess if a given technique or resource helps to improve the results. It should be very helpful having a development dataset. We built our own development dataset from a random sample of 1099 documents taken from the small training dataset (Training v.2016a). Thus, our development dataset only collects articles from the same set of journals used to build the test datasets of the task. As mentioned above, these articles were removed from our training set in order to avoid any potential bias.

3.2 ElasticSearch

Our approach relies on the assumption that similar documents should be classified by similar MeSH labels. While previous work has exploited a kNN approach in order to propose the MeSH labels of the relevant documents for a given query (test document), we propose to calculate document similarity by using ElasticSearch, an open source search

engine. ElasticSearch provides horizontal scalability, that is, it is able to index large collections of documents. The main advantage of ElasticSearch is its capacity to create distributed systems by specifying only the configuration of the hierarchy of nodes. Then, ElasticSearch is self-managed to maintain better fault tolerance and load distribution. The core of ElasticSearch is Lucene,[4] a free, open-source and de-facto standard retrieval software library. Lucene is based on the well-known and commonly used vector space model for information retrieval. The efficiency of Lucene is due to it searches on index instead of searching the text directly.

Another important advantage of ElasticSearch is that it does not require very high computing power and a high storage capacity to index large collections. In this study, ElasticSearch (version 2.2) was installed on a server Ubuntu Server 14.0f4 with 24GB of RAM and 500GB of disk space. We create an index (that is like a database in a relational database) built from the training dataset. By default, each index in Elasticsearch is configured with five shards, lucene instances. One of the most important advantages of ElasticSearch is that the shards can be distributed amongst all nodes in the cluster, and can be moved from one node to another in the case of node failure. Each shard has a backup copy.

As it has already been mentioned before, our approach is to index the training dataset and represent each test document as a query. In particular, we define two different types of index, one using the large training dataset (Training v.2016a) and the second one using the small training set (Training v.2016b), that only contains articles from the journals used for testing. Both collections are indexed using bag-of-word model. To translate the test documents to the queries, each document is also represented as bag of words. Then, each query is fired against the index, returning the most relevant documents (relevance scoring is calculated using TF/IDF). Figure 1 shows the basic architecture of our system.

Finally, the MeSH categories from the relevant documents are collected. The simplest approach would be to return the whole set of MeSH labels for all retrieved documents. However, we define a metric to rank each MeSH category for a given test document based on the total number of occurrences of the label in the whole index as well as the similarity of the relevant document containing this category with the test document (query). Our scoring system is based on the hypothesis that similar documents should have similar MeSH categories, and that the most used MeSH categories should achieve higher scores. The following formula describes this metric:

[4]https://lucene.apache.org/core/

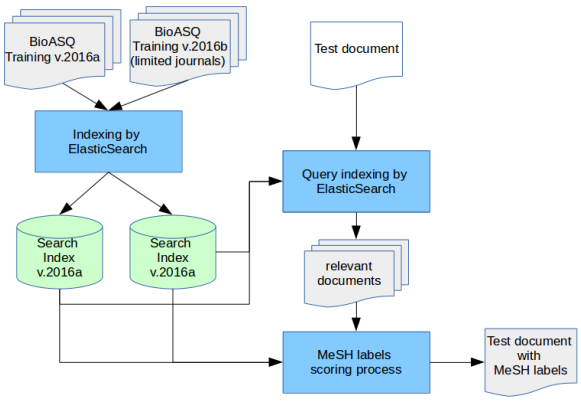

Figure 1: Architecture of our system.

$$score(l,q) = (tf(l) - R) \sum_{d:l \in d} score(d,q) \qquad (1)$$

where $tf(l)$ refers to the total number of occurrences of the label in the whole index, and R is a discrete parameter that indicates the minimum number (minus one) of times a label has to appear in the relevant documents in order to be considered as a candidate label for the test document. R takes only three values: 0, 1 and 2. Finally, $score(d,q)$ represents the scoring of a document d, containing the MeSH label l, for a given query q (a test document).

While some documents present a large number of MeSH labels, others only contain a small set. In order to reduce this variability, the scoring for a label is normalized using the following equation:

$$score(l,q)_n = (tf(l) - R) \sum_{d:l \in d} \frac{score(d,q)}{max_{a:l \in a} score(a,q)} \qquad (2)$$

Finally, we choose those MeSH categories with a score higher than a threshold (which was set empirically upon our development dataset). It should be noted that if the threshold is set to 0 then the whole set of MeSH categories for all retrieved documents is returned.

4 Experimental results

Task 4.a began on 8th of February, 2016, however we enrolled almost two months later. Our first submission was on the fourth week of the second batch (March 14-April 11). Unfortunately, there is no results for our systems at the time of writing this paper and we cannot offer any official definitive results. For this reason, we show the results of our settings on own development dataset.

The performance of the participating systems is evaluated using standard IR measures (e.g., precision, recall, accuracy), as well as hierarchical variants of them, such as Lowest Common Ancestor F-measure (LCA-F). The HEMKit tool[5] (Kosmopoulos et al., 2015) was used to evaluate our different settings on our development set.

We experimented with different settings such as the index used to retrieve the documents, the number of relevant documents (10, 20, 30 and 40), the option of including MeSH labels without repetitions, and the threshold to select the MeSH labels. We also provided baseline results based on the use of the MTI system (Mork et al., 2013). Table 1 shows the results. Our best result among all approaches is highlighted in bold. The different settings are described bellow:

- **MTI**: our baseline system using MTI.
- **Elastic-2016V-X-R-T**: V refers to the index used: a for the index built from the large training dataset (Training v.2016a) or b for index built from the small training dataset (Training v.2016b). X refers to the number of relevant documents retrieved by ElasticSearch. R is a discrete parameter that indicates the minimum number (minus one) of times a label has to appear in the relevant documents in order to be considered as a candidate label for the test document. R takes only three values: 0, 1 and 2. T refers to the minimum threshold in equation 2 for selecting the MeSH labels.

We experimented with different settings such as the index type, the number of relevant documents or the threshold used to select the MeSH categories. Results for some of these settings are shown in Table 1 (we do not show all results for lack of space). Experiments showed that the increase in the number of relevant documents achieved to improve precision and recall values. Finally, the number of documents was set to 30 because this value achieved the best results while keeping the processing time low (less than 15 seconds per document).

The simplest approach by using ElasticSearch (that is, returning the whole set of MeSH labels for all retrieved documents) provides a very high recall (93%) but with a very low precision (15-16%). We tried with different values for the threshold T (minimum score to select the MeSH categories) and decided that 1.5 was a good value balancing precision with recall as higher values returned.

Regardless of the other parameters, the index type, that is, the use of the large training dataset versus the small training dataset, does not seem to obtain a significant difference. The results obtained with the small index are slightly better than those obtained with the large index.

As could be expected, the fact of including the MeSH categories with frequencies lower than 2 achieves better recall value, but has worse precision. On the contrary, if we require that the MeSH category has to occur at least twice in the set of the relevant documents in order to be selected, the precision increases but the recall decreases.

When comparing the experimental results of the current study with those from the MTI baseline, we can observe that our approach outperforms this baseline at recall, but with a significant decrease in precision. Therefore, we need to further research for techniques to improve precision. On the other hand, it should be noted that our system based on ElasticSearch gives a much better time response than the MTI system.

Finally, we also combined the MTI baseline with our approach based on ElasticSearch by outputing all MeSH labels proposed by MTI as well as those proposed by ElasticSearch. In this case, the best value for the threshold T was 3. This setting provided the best results (see two last rows in Table 1).

Table 2 shows our results on a very small sample (302 articles) from the test batch 3-week 5, and thereby, no conclusion can be drawn yet. The setting used for this submission was only based on providing the labels from the top 30 articles retrieved by ElasticSearch from the small index (Training v.2016b). This set was also extended with the MeSH labels proposed by MTI.

[5] http://nlp.cs.aueb.gr/software

Systems	F	R	P	LCA-F	LCA-R	LCA-P
MTI	0.7065	0.6741	0.6881	0.4165	0.4217	0.454
Elastic-2016a-30-0-0	0.2734	0.9394	0.1647	0.2004	0.6792	0.1206
Elastic-2016b-30-0-0	0.2626	0.9364	0.1571	0.1933	0.6752	0.1156
Elastic-2016a-30-1-1.5	0.5150	0.8303	0.3926	0.3345	0.5589	0.2510
Elastic-2016b-30-1-1.5	0.5188	0.8474	0.3925	0.3377	0.5717	0.2519
Elastic-2016a-30-2-1.5	0.5592	0.7944	0.4537	0.3580	0.5282	0.2861
Elastic-2016b-30-2-1.5	0.5632	0.8066	0.4543	0.3625	0.5364	0.2889
MTI + Elastic-2016a-30-2-3	**0.6266**	0.8168	0.5330	0.4034	0.5420	0.3396
MTI + Elastic-2016b-30-2-3	**0.6207**	0.8039	0.5297	0.3982	0.5345	0.3357

Table 1: Experimental results on our development dataset.

Systems	F	P	R	LCA-F	LCA-P	LCA-R
MTI	0.6373	0.6650	0.6674	0.3949	0.4085	0.4168
MTI + Elastic-2016b-30-2-3	0.4408	0.3295	0.6910	0.3890	0.4774	0.7928

Table 2: Experimental results on the test batch 3, week 5 (Annotated articles:302/3130).

5 Conclusions

Several works have already applied a k-Nearest-Neighbors (kNN) approach for semantic indexing (Névéol et al., 2007; Mao et al., 2014; Dramé et al., 2014). This approach relies on the assumption that similar documents should be classified by similar MeSH labels. We make the same assumption, but our work is the first that explores the document similarity using ElasticSerach instead of kNN. Our approach achieves similar results to those reported in previous editions of BioASQ, while keeping the processing time much lower than that reported by the MTI baseline (30-45 seconds per document). Our approach yields high recall (80-84%), but with a low precision (45-53%). Therefore, we plan to study alternatives that aim to improve precision. As future steps, we also plan to determine semantic similarity between documents using word embeddings (Mikolov et al., 2013), instead of the well-known and commonly used vector space model for information retrieval.

Acknowledgments

This work was supported by eGovernAbility-Access project (TIN2014-52665-C2-2-R).

References

George Balikas, Ioannis Partalas, Axel-Cyrille Ngonga Ngomo, Anastasia Krithara, Eric Gaussier, and Georgios Paliouras. 2014. Results of the BioASQ tasks of the Question Answering Lab at CLEF 2014. In *Proceedings of CLEF 2014*.

Georgios Balikas, Aris Kosmopoulos, Anastasia Krithara, Georgios Paliouras, and Ioannis Kakadiaris. 2015. Results of the BioASQ tasks of the Question Answering Lab at CLEF 2015. In *Proceedings of CLEF 2015*.

Khadim Dramé, Fleur Mougin, and Gayo Diallo. 2014. A k-nearest neighbor based method for improving large scale biomedical document indexing. In *Proceedings of the 6th International Symposium on Semantic Mining in Biomedicine (SMBM)*, pages 19–26.

William Hersh, Chris Buckley, TJ Leone, and David Hickam. 1994. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of SIGIR94*, pages 192–201.

Minlie Huang, Aurélie Névéol, and Zhiyong Lu. 2011. Recommending MeSH terms for annotating biomedical articles. *Journal of the American Medical Informatics Association*, 18(5):660–667.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Jun'ichi Tsujii. 2011. Overview of BioNLP shared task 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 1–6.

Aris Kosmopoulos, Ioannis Partalas, Eric Gaussier, Georgios Paliouras, and Ion Androutsopoulos. 2015. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29(3):820–865.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu,

Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. 2015. The CHEMDNER corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7(S1):1–17.

Ke Liu, Junqiu Wu, Shengwen Peng, Chengxiang Zhai, and Shanfeng Zhu. 2014. The FUDAN-UIUC participation in the BioASQ challenge Task 2a: The antinomyra system. In *CLEF2014 (Working Notes)*, volume 129816, page 100.

Ke Liu, Shengwen Peng, Junqiu Wu, Chengxiang Zhai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2015. MeSHLabeler: improving the accuracy of large-scale MeSH indexing by integrating diverse evidence. *Bioinformatics*, 31(12):i339–i347.

Yuqing Mao, Chih-Hsuan Wei, and Zhiyong Lu. 2014. NCBI at the 2014 BioASQ Challenge Task: Large-scale Biomedical Semantic Indexing and Question Answering. In *CLEF2014 (Working Notes)*, pages 1319–1327.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

James G Mork, Antonio Jimeno-Yepes, and Alan R Aronson. 2013. The NLM Medical Text Indexer System for Indexing Biomedical Literature. In *Proceedings of BioASQ CLEF 2013*.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of BioNLP shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7.

Aurélie Névéol, James G Mork, and Alan R Aronson. 2007. Automatic indexing of specialized documents: using generic vs. domain-specific document representations. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 183–190.

Ioannis Partalas, Éric Gaussier, and Axel-Cyrille Ngonga Ngomo. 2013. Results of the First BioASQ Workshop. In *Proceedings of BioASQ CLEF 2013*, pages 1–8.

Rafal Rak, Lukasz A Kurgan, and Marek Reformat. 2007. Multilabel associative classification categorization of MEDLINE articles into MeSH keywords. *IEEE engineering in medicine and biology magazine*, 26(2):47.

Francisco J Ribadas, Luis M De Campos, Vıctor M Darriba, and Alfonso E Romero. 2014. CoLe and UTAI participation at the 2014 BioASQ semantic indexing challenge. In *Proceedings of the CLEF BioASQ 2014 Workshop*, pages 1361–1374.

Miguel E Ruiz and Padmini Srinivasan. 2002. Hierarchical text categorization using neural networks. *Information Retrieval*, 5(1):87–118.

Isabel Segura-Bedmar, Paloma Martinez, and Daniel Sanchez-Cisneros. 2011. The 1st DDIExtraction-2011 Challenge Task: Extraction of Drug-Drug Interactions from Biomedical Texts. In *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction 2011*, pages 1–9.

Isabel Segura Bedmar, Paloma Martínez, and María Herrero Zazo. 2013. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013), page 341350.

Amber Stubbs, Christopher Kotfila, Hua Xu, and Özlem Uzuner. 2015. Identifying risk factors for heart disease over time: overview of 2014 i2b2/UTHealth shared task Track 2. *Journal of Biomedical Informatics*, 58:S67–S77.

Lei Tang, Suju Rajan, and Vijay K Narayanan. 2009. Large scale multi-label classification via metalabeler. In *Proceedings of the 18th international conference on World Wide Web*, pages 211–220.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the BioASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(1):1.

Grigorios Tsoumakas, Manos Laliotis, Nikos Markantonatos, and Ioannis Vlahavas. 2013. Large-scale semantic indexing of biomedical publications at bioasq. In *Proceedings of BioASQ CLEF 2013*.

Antonio Jose Jimeno Yepes, Laura Plaza, Jorge Carrillo-de Albornoz, James G Mork, and Alan R Aronson. 2015. Feature engineering for MEDLINE citation categorization with MeSH. *BMC Bioinformatics*, 16(1):1.