The Fudan-UIUC participation in the BioASQ Challenge Task 2a: The Antinomyra system

Ke Liu[1,2], Junqiu Wu[3], Shengwen Peng[1,2], Chengxiang Zhai[4], and Shanfeng Zhu[1,2] *

[1] School of Computer Science, Fudan University, Shanghai 200433, P. R. China,
[2] Shanghai Key Lab of Intelligent Information Processing, Fudan University, Shanghai 200433, P. R. China
{antinomyra,pswgoo}@gmail.com
zhusf@fudan.edu.cn
[3] School of Information Science & Engineering, Central South University, 410083 P. R. China
oxalca@gmail.com
[4] Department of Computer Science, University of Illinois at Urbana-Champaign, IL 611801, USA
czhai@illinois.edu

Abstract. This paper describes the Antinomyra System that participated in the BioASQ Task 2a Challenge for the large-scale biomedical semantic indexing. The system can automatically annotate MeSH terms for MEDLINE citations using only title and abstract information. With respect to the official test set (batch 3, week 5), based on 1867 annotated citations out of all 4533 citations (June 6, 2014), our best submission achieved 0.6199 in flat Micro F-measure. This is 9.8% higher than the performance of official NLM solution Medical Text Indexer (MTI), which achieved 0.5647 in flat F-measure.

Keywords: MeSH Indexing; Logistic Regression; Learning to Rank; Multi-Label Classification.

1 Introduction

1.1 The Problem

Medical Subject Headings (MeSH) are used by National Library of Medicine (NLM) to index articles in MEDLINE [1]. MeSH is organized in hierarchical structure, and slightly updated every year. In 2014, there are altogether 27149 MeSH headings[5]. Usually each article is annotated by 5 to 20 MeSH headings. Many studies have been carried out to utilize MeSH for efficiently retrieving and mining biomedical documents for knowledge discovery [2, 3, 4, 5, 6, 7]. The

* Corresponding author
[5] http://www.nlm.nih.gov/mesh/introduction.html

accurate prediction of MeSH headings for each citation will greatly reduce the financial and time cost of annotating biomedical documents. The BioASQ Task2a is a large scale biomedical semantic indexing competition for automatic MeSH annotation. Each week, thousands of new MEDLINE citations are provided to the competition participants, who are required to submit predicted MeSH headings of each citation in 21 hours. Since each MeSH heading can be deemed as a class label, the MeSH annotation problem is a multi-label classification problem. For each citation, our Antinomyra system tries to assign it a certain number of MeSH headings out of all 27149 MeSH headings.

1.2 Challenges

Simple mapping does not work well Many MeSH headings do not appear directly in the title or abstract of a target citation, which means that a simple mapping does not work very well. This is why we resort to advanced machine learning methods to predict MeSH headings.

The amount of information is insufficient During the competition, we have very limited information of target citations, more specifically, only titles and abstracts are available in this task. By contrast, MeSH indexers have the full text for annotating MeSH headings. Since the main text contains some important clues, predicting MeSH headings with very limited information is a big challenge for the competition participants. Indeed, it will be very interesting if we can know the performance of professional MeSH annotators with the same information as the BioASQ Task2a participants. This may bring more insights on how to make use of the information in the title and abstract.

Table 1. The frequencies of some typical MeSH headings in our Local MEDLINE database of 12,504,999 citations.

ID	MeSH	Count	Frequency Rank
6801	Humans	8152852	1
8297	Male	4777692	2
18570	Risk Assessment	129816	100
2540	Cerebral Cortex	74513	200
12987	Soil	23178	1000
12045	Regulatory Sequences, Nucleic Acid	12503	2000
23001	Transplantation Tolerance	1532	10000
6991	Hypnosis, Anesthetic	199	20000

There are large variations in the occurrence of different MeSH headings Some MeSH headings, such as check tags Humans, Male and Female, appear very frequently, while some others are very rare. We have counted the occurrence of each MeSH heading in the whole MEDLINE. Only about 150 MeSH

headings have occurred in more than 1% of the whole MEDLINE. That is to say, a vast majority of MeSH headings do not appear very often. It is not surprising that, for many MeSH headings, we lack of enough positive examples (citations) to train accurate models. As illustrated in Table 1, MeSH heading"Hypnosis, Anestheti" only occurs in 199 out of more than 12 million citations.

2 Related Work

In the past few years, especially for the last BioASQ challenge (task 1a) [8], many studies have been carried out to improve the prediction accuracy of MeSH heading suggestions [9, 10, 11]. In addition to NLM's official solution MTI [12], there are two other methods most closely related to our method [9, 10, 11]. The first one is MetaLabeler, which was proposed in [13] and utilized by Tsoumakas et al. in BioASQ Task1a for MeSH prediction [9]. In this method, firstly, for each MeSH heading, a binary classification model was trained using linear SVM. Secondly, a regression model was trained to predict the number of MeSH headings for each citation. Finally, given a target citation, different MeSH headings were ranked according to the SVM prediction score of each classifier, and the top K MeSH headings were returned as the suggested MeSH headings, where K is the number of predicted MeSH headings by the model. The second one is the learning to rank (LTR) method, which was widely used in information retrieval [14] and utilized by Lu et al. for automatic MeSH annotation [10, 11]. In this method, each citation was deemed as a query and each MeSH headings as a document. LTR method was utilized to rank candidate MeSH headings with respect to target citation. The candidate MeSH headings came from similar citations (nearest neighbors). In our system, we also made use of the LTR framework for predicting MeSH headings. However, in addition to the information from similar citations, we also use the prediction scores from individual MeSH classifier to improve the prediction accuracy.

3 Method

3.1 Data processing

We downloaded the whole PubMed database in an XML format. The citations without abstract or MeSH annotations were then filtered. Finally we obtained 12,504,999 citations. Some citations have subsections, such as backgrounds, methods and results, but we didn't treat these subsections separately. Based on target journals in BioASQ Task2a, we kept the latest 20,000 citations for validation and testing, and 1,000,000 additional latest citations for training.

3.2 Tokenization

BioTokenizer[6][15] was used in our system for the tokenization task. We also tried CoreNLP[7] and doc2mat[8], but the performance of BioTokenizer was the best. After the processing of BioTokenizer, the text of a citation was tokenized and the stemming task was carried out simultaneously. Finally, based on the tokenization result, we compiled a dictionary for converting the text of each citation into a vector.

3.3 Primary Classifiers

Similar to MetaLabeler[9], we trained a binary classifier for each label (MeSH heading). We call these binary classifiers as Primary Classifiers in our framework. For efficiency, we used logistic regression instead of SVM to train these classifiers. For a target citation, each Primary Classifier can predict the annotation probability of the corresponding label.

3.4 Nearest Neighbors

Given a target citation, we used NCBI efetch[9] to find its similar (neighbor) citations. The MeSH headings from these neighbors were deemed as promising candidates to annotate the target citation. It is generally believed that candidate MeSH headings from most similar citations are more important than those from less similar citations. To measure the importance of each candidate MeSH heading, we added up the similarity scores between the target citation and its neighbor citations that contain this candidate MeSH heading. These similarity scores could be also retrieved by NCBI efetch.

3.5 Learning to Rank Framework

Features

We used a learning to rank framework to integrate multiple types of information, such as the information from Primary Classifiers and nearest neighbors. For a target citation, firstly we used the Primary Classifiers to calculate the annotation probability (score) of every MeSH heading. Then we retrieved similar citations for the neighbor scores. Finally, these two scores were considered as features in the LTR framework. In the BioASQ task2a challenge, the default results of NLM official solution MTI were also considered as a feature in the LTR framework.

Candidates

[6] See http://sifaka.cs.uiuc.edu/jiang4/software/BioTokenizer.pl
[7] See http://nlp.stanford.edu/software/corenlp.shtml
[8] See http://glaros.dtc.umn.edu/gkhome/files/fs/sw/cluto/doc2mat.html
[9] See http://www.ncbi.nlm.nih.gov/books/NBK25499/

For a target citation, a MeSH label could be a candidate MeSH of this citation if and only if it satisfied any of the two following requirements.

1. The label appeared in the similar citations of the target citation;
2. The labels Primary Classifier score was in the top 100 of all MeSH labels.

LTR method

Each citation was treated as a query, and the candidate MeSH headings as documents. Then LambdaMART[16] was used as the ranking method in the learning to rank framework. The LTR training data contained about 30,000 citations from BioASQ task1a test set. 1000 decision trees were used in the LambdaMART model without overfitting. After getting the LTR score of each candidate MeSH, we used the best threshold which was tuned from the validation set to determine how many labels we should return.

4 Experimental Results

4.1 Implementation

The whole project was coded in C++. We used some third part libraries in our solution: Liblinear[10][17] for Logistic Regression, RankLib[11] for LambdaMART algorithm and JsonCpp[12] for Input/Output json files. We also used OpenMP[13] to make our task parallel.

4.2 Computational performance

The server we used for the challenge has 4 * Intel XEON E5-4650 2.7GHzs CPU and 128GB RAM. The most computational expensive part is the training of Primary Classifiers, which took 5 days. All other training tasks took about 1 day. However, the time cost for prediction is low. For annotating 10,000 citations, it only took 2 hours.

4.3 Label based & Example based Performance

As shown in Table 2, we compared the performance of five different methods, MTIFL, MTIDEF, directly mapping, MetaLabeler, and LTR. These methods were evaluated on a test set of 9040 citations, which were published in BioASQ task2a target journal between 2012 and 2013. By integrating multiple types of information, LTR achieved the highest MiF of 0.61, followed by MTIDEF

[10] See http://www.csie.ntu.edu.tw/ cjlin/liblinear/
[11] See http://sourceforge.net/p/lemur/wiki/RankLib/
[12] See http://jsoncpp.sourceforge.net/
[13] See http://openmp.org/wp/

(0.572), MTIFL (0.564), Metalabeler (0.56) and directly mapping (0.27). Based on this framework, in the last week of Batch 3 of BioASQ Task2a (annotated articles: 1867/4533 on 6 June)[14], our system achieved the highest performance in terms of both flat F-Measure (0.6199) and hierarchical F-Measure (0.5145)

Table 2. The performance comparsions of 5 methods. These methods are evaluated on an offline testing set that has 9040 citations published between 2012 and 2013.

Method	MiP	MiR	MiF	EBP	EBR	EBF	MaP	MaR	MaF
MTI FirstLine Inedx (MTIFL)	0.614	0.522	0.564	0.619	0.539	0.555	0.516	0.492	0.471
Default MTI (MTIDEF)	0.574	0.571	0.572	0.578	0.591	0.564	0.513	0.537	0.494
Directly Mapping	0.236	0.314	0.270	0.250	0.329	0.268	0.374	0.415	0.349
MetaLabeler	0.558	0.561	0.560	0.556	0.577	0.550	0.460	0.462	0.436
LTR Without MTI Features	0.612	0.593	0.603	0.606	0.613	0.594	0.507	0.485	0.470
LTR Ensemble	0.621	0.601	0.610	0.616	0.623	0.603	0.523	0.507	0.490

5 Discussions and Conclusions

Although our system performed very well in the competition, the system could be further improved in several aspects. Firstly, for the local offline evaluation, we used only flat measures (MiF) to tune our method, which leads to the underperformance of our system in hierarchical measure (LCA-F). Considering the significant difference between MiF and LCA-F, we could further improve the performance of our system by using LCA-F to tune the model. Secondly, we can consider some special MeSH headings separately. It is noticed that, for some most frequent MeSH headings, such as check tags, direct prediction may improve the annotation accuracy of these MeSH headings[18]. Finally, we have not used any indexing rules in our system. Incorporating this kind of human knowledge into the system would be a very promising direction for significantly increasing the accuracy of MeSH heading recommendations.

As a general framework, the LTR we used for MeSH heading recommendations can also be applied to some other types of tasks. Moreover, the performance could be further improved if more information is integrated. As such, it raises an interesting question as to what the upper bound of our system will be in the presence of more information integrated in the LTR framework.

Acknowledgments. This work has been partially supported by National Natural Science Foundation of China (61170097), and Scientific Research Starting Foundation for Returned Overseas Chinese Scholars, Ministry of Education, China. Shanfeng Zhu would like to thank the China Scholarship Council for the

[14] See http://bioasq.lip6.fr/results/2a/

financial support on his visit at University of Illinois at Urbana-Champaign. We would like to thank Hongning Wang and Mingjie Qian in UIUC for their helpful suggestions and insightful discussion, thank Jieyao Deng in Fudan University and Tianyi Peng in Tsinghua University for their help in coding works during the competition.

References

[1] Nelson, S. J., Schopen, M., Savage, A. G., Schulman, J. L., & Arluk, N.: The MeSH translation maintenance system: structure, interface design, and implementation. Medinfo, 11(Pt 1), 67–69. (2004)
[2] Gu, J., Feng, W., Zeng, J., Mamitsuka, H., Zhu, S.: Efficient Semi-supervised MEDLINE document clustering with MeSH semantic and global content constraints. IEEE Transactions on Cybernetics, 43 (4), 1265–1276, (2013)
[3] Zhu, S., Zeng, J., Mamitsuka, H.: Enhancing MEDLINE document clustering by incorporating MeSH semantic similarity. Bioinformatics 25(15): 1944–1951 (2009)
[4] Zhu, S., Takigawa, I., Zeng, J., Mamitsuka, H.: Field independent probabilistic model for clustering multi-field documents. Information Processing & Management. 45(5): 555–570 (2009)
[5] Huang, X., Zheng, X., Yuan, W., Wang, F., Zhu, S.: Enhanced clustering of biomedical documents using ensemble non-negative matrix factorization. Information Science 181(11): 2293–2302 (2011)
[6] Rajpal, D. K., Qu, X. A., Freudenberg, J. M., Kumar, V. D.: Mining emerging biomedical literature for understanding disease associations in drug discovery. Methods Mol Biol. 1159:171–206 (2014)
[7] Theodosiou, T., Vizirianakis, I. S., Angelis, L., Tsaftaris, A., Darzentas, N.: MeSHy: Mining unanticipated PubMed information using frequencies of occurrences and concurrences of MeSH terms. J Biomed Inform. 44(6):919–26. (2011)
[8] Partalas, I., Gaussier, E., & Ngomo, A. C. N.: Results of the First BioASQ Workshop. In: BioASQ@ CLEF (pp. 1-8). (2013)
[9] Tsoumakas, G., Laliotis, M., Markantonatos, N., & Vlahavas, I. P.: Large-Scale Semantic Indexing of Biomedical Publications. In BioASQ@ CLEF. (2013)
[10] Huang, M., Neveol, A., Lu, Z.: Recommending MeSH terms for annotating biomedical articles. Journal of the American Medical Informatics Association, 18(5): 660–667. (2011)
[11] Mao, Y., Lu, Z.: NCBI at the 2013 BioASQ challenge task: Learning to rank for automatic MeSH indexing[R]. Technical report. (2013)
[12] Mork, J. G., Jimeno-Yepes, A., & Aronson, A. R.: The NLM Medical Text Indexer System for Indexing Biomedical Literature. In BioASQ@ CLEF. (2013)
[13] Tang, L., Rajan, S., Narayanan, V.K.: Large scale multi-label classification via metalabeler. Proceedings of the 18th international conference on World Wide Web, 211–220 ACM (2009)
[14] Cao, Z., Qin, T., Liu, T. Y., Tsai, M. F., Li, H.: Learning to rank: from pairwise approach to listwise approach. Proceedings of the 24th international conference on Machine learning.129–136. ACM. (2007)
[15] Jiang, J., Zhai, C.: An empirical study of tokenization strategies for biomedical information retrieval. Information Retrieval, 10(4-5): 341–363. (2007)
[16] Burges, C. J.: From RankNet to LambdaRank to LambdaMART: An overview. Technical Report MSR-TR-2010-82, Microsoft Research. (2010)

[17] Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., Lin, C. J.: LIBLINEAR: A library for large linear classification. The Journal of Machine Learning Research, 9: 1871–1874. (2008)
[18] Yepes, A. J., Mork, J. G., Demner-Fushman, D., Aronson, A. R.: Comparison and combination of several MeSH indexing approaches. AMIA Annual Symposium Proceedings. 2013: 709–718. American Medical Informatics Association. (2013)