NCBI at the 2013 BioASQ challenge task: Learning to rank for automatic MeSH indexing

Yuqing Mao[1], Zhiyong Lu[1,*]

[1] National Center for Biotechnology Information (NCBI), 8600 Rockville Pike, Bethesda, MD 20894, USA

{yuqing.mao,zhiyong.lu}@nih.gov

Abstract. In this paper we describe our statistical approach for the 2013 BioASQ biomedical semantic indexing task where participating teams are provided with PubMed articles and asked to return relevant MeSH terms. Our overall approach builds on our previous learning-to-rank research with features extracted from PubMed articles and MeSH terms collected from neighbor documents. More specifically, candidate MeSH terms were ranked using the MART algorithm. Furthermore, we make multiple adjustments to our previous method including dynamically selecting both the number of neighboring documents and the number of MeSH terms returned. In addition, we also incorporate new features from domain knowledge. Our best results on the largest official test set (Batch 2, Week 5) are 0.4617 and 0.5471 in flat and hierarchical F-measures, respectively. We find that our results compare favorably to the state of the art and that the newly proposed features result in improved performance.

Keywords: MeSH terms; Indexing; Text categorization; Learning-to-rank; Evaluation.

1 Introduction

MeSH® indexing refers to the task of manually assigning relevant MeSH terms to new publications in MEDLINE by the human indexers at the US National Library of Medicine (NLM). As a result, MeSH terms can then be used implicitly or explicitly when searching the biomedical literature in PubMed [1]. MeSH terms also play a role in many other scientific investigations [2, 3].

Like many other manual annotation projects, MeSH indexing is labor-intensive and time-consuming. Hence, many (semi-)automated systems for assisting MeSH indexing have been proposed [4-6]. The NLM Medical Text Indexer (MTI) and its newer version, Medical Text Indexer First Line (MTIFL) are both used in NLM production pipelines to assist human annotators with indexing MeSH main headings, and more recently, main heading/subheading pairs [7].

Automatic MeSH indexing can be cast as a multi-class text classification problem where each MeSH term represents a distinct class label. In this regard, it is closely related to automatic ontological annotation problems such as [8], where the number of class labels is typically very large and hence the corresponding classification accuracy remains modest. To advance the state of the art on such problems, a community-wide challenge event known as BioASQ was organized in 2013 (http://www.bioasq.org/), specifically aiming for a solution to the information access problem of biomedical information seekers. BioASQ 2013 consists of two sub-tasks including the large-scale online biomedical semantic indexing (Task 1a) and the introductory biomedical semantic QA (Task 1b).

We participated in the 2013 BioASQ semantic indexing task where teams are provided with PubMed articles and asked to return relevant MeSH terms accordingly. Our overall approach builds on our previous research that reformulates the MeSH prediction task as a ranking problem [4]: we first obtain an initial list of MeSH terms from the *k*-nearest neighbor articles as candidates for each target article. Next, we apply a learning-to-rank algorithm to sort the candidate MeSH terms based on the learned associations between the document text and each MeSH term.

In order to improve MeSH prediction results, we made multiple extensions to our previous approach: First, we experiment with several additional learning-to-rank algorithms [11], including MART, LambdaMART, and AdaRank. Second, we choose the number of neighbor documents dynamically, according to the document similarity score given each target article. Third, we design an automatic cutoff measure to return only the most relevant MeSH terms for each target article, as opposed to returning a fixed number of MeSH terms. Finally, we propose incorporating the MTIFL results as a new feature in our learning algorithm.

A brief description of our learning-to-rank approach is presented in Section 2, along with a description of different strategies used to improve performance in the BioASQ challenge. In Section 3 the results of different approach settings are compared on the BioASQ dataset. In sections 4 and 5 we present a brief discussion of the results and conclusions of our participation in this challenge.

2 Methods

2.1 *K*-nearest neighbors

We first adapt the PubMed Related Articles algorithm [9] to retrieve *k*-nearest neighbors for each new PubMed article from the MEDLINE database. In this work, we rebuilt the index of PubMed documents by removing all associated MeSH terms, i.e. modifying the document length and the inverse document frequency such that MeSH terms are not used in the computation of the neighbor documents. In other words, the similarity between two documents is solely based on the words they have in common. The parameter *k* was fixed in our previous work (k=20), which means the same number of neighbors will be included for all target articles. However, some articles may only have a few very similar documents. We therefore adjust the parameter *k* dynamically according to the similarity scores of the neighbors: the smaller the average similarity score of the neighbors, the fewer neighbors will be used.

Once those *k*-nearest neighbor documents are retrieved, we collect all the unique MeSH terms associated with those neighbor documents. Since Task 1a uses the MeSH hierarchy for classifying PubMed documents and evaluates the results with hierarchical metrics, we only considered the main headings and removed subheadings attached to the main headings.

Next, each MeSH term in the initial list was assigned a score by the ranking algorithm described below. In our previous work, the top *N* ranked MeSH terms were considered relevant to the target article; we set the number *N* to be 25. We found, however, that the average number of MeSH terms per article in the BioASQ training data was only 12.6. In this work we used an automatic cut-off method to further prune the results from the top 25 ranked MeSH terms as follows:

$$S_{i+1} / S_i < i / (i + 1 + \alpha) \qquad \mathbf{(1)}$$

where S_i is the score of the predicted MeSH term at position *i* in the top 25 ranking list. The rationale for Formula (1) is that if the (i+1)th MeSH term was assigned with a score much smaller than the *i*th MeSH term, the MeSH terms ranked lower than *i* would be not be considered relevant to the target article. Formula (1) also takes account of the factor that the difference between lower-ranked MeSH terms is subtler than that between higher-ranked MeSH terms. The parameter α was empirically set to be 1.2 in this work.

2.2 Learning to rank

Following our previous work, we approached the task of MeSH term indexing as a ranking problem. First, we obtained a training set consisting of biomedical articles with human assigned MeSH terms from MEDLINE. For each article, we obtain an initial list of MeSH terms from its neighbor documents. Each main heading is then represented as a feature vector. For the list of MeSH term from its neighbor documents, denoted by $\{M_1, M_2, \ldots, M_N\}$, where *N* is the number of feature vectors (the number of MeSH terms, each MeSH term is represented by a vector of features), M_i is the *i*th feature vector, we obtain a corresponding list $\{y_1, y_2, ..., y_N\}$, where $y_i \in \{0,1\}$ is the *i*th class label. y_i=1 if the MeSH term was manually assigned to the target article by expert annotators of the NLM, otherwise y_i=0.

In our previous work, we learned the ranking function with ListNet [10], which sorts the results based on a list of scores. In this work we evaluated several other learning-to-rank algorithms [11] on the BioASQ test dataset, including MART, RankNet, Coordinate Ascent, AdaRank, and LambdaMART, which are available in RankLib v2.2[1], and found that MART achieved the best performance. MART and its derivation algorithms can be viewed as generalizations of logistic regression [12].

[1] http://sourceforge.net/p/lemur/wiki/RankLib/

They model $p_{i,1}$, the probability that a MeSH term x_i can be placed at the first position of a ranked list, as:

$$p_{i,1} = \Pr(y_i = 1 \mid x_i) = \frac{e^{F_{i,1}(x_i)}}{e^{F_{i,0}(x_i)} + e^{F_{i,1}(x_i)}} \quad (2)$$

For the ranking function, MART adopts the flexible "additive model," which is a function of M terms:

$$F^{(M)}(x) = \sum_{m=1}^{M} \rho_{\mathrm{m}} h(x; a_m) \quad (3)$$

where the base learner $h(x; a_m)$ is typically a regression tree. The parameters ρ_m and a_m are learned from the data by maximum likelihood, which is equivalent to minimizing the negative log-likelihood loss:

$$L = \sum_{i=1}^{N} L_i, L_i = -\sum_{k=0}^{1} r_{i,k} \log p_{i,k} \quad (4)$$

where $r_{i,k}$= 1 if y_i=k otherwise $r_{i,k}$ = 0.

The learning process then becomes an optimization to maximally align the machine predictions with the gold standard.

2.3 Features

In our previous work we developed various novel features which can be categorized as: neighborhood features, word unigram/bigram overlap features, translation probability features, query-likelihood features, and synonym features. In this work we use the results of MTIFL as a domain-specific knowledge feature. MTIFL is used as one of the baselines in the BioASQ Task 1a, and MTIFL uses MetaMap indexing, mapping the phrases in the text to UMLS (Unified Medical Language System) concepts. We used a binary feature indicating whether an entry term can be observed in the MTIFL results.

We also replace the 13,999 documents used in our previous work to compute the average length of documents and the document frequency for each word with a larger set of 58,088 documents from the MEDLINE database. The translation model and the background language model were rebuilt through training with this new data set accordingly.

2.4 Additional modifications

A complete list of the differences is detailed in Table 1. In addition to the differences mentioned above, the table includes several other notable modifications: we updated our lexicon with MeSH 2013 version; we updated training documents according to the select BioASQ journals and used a larger set of documents for training our algorithms and building statistical models; finally we developed a heuristic rule based on the error analysis of our results on foreign journals.

Table 1: Differences between this work and our previous work Huang et al., 2011.

Notable Differences	Huang et al., 2011	This work
Learning-to-rank algorithm	ListNet	MART
# of neighbor documents	Top 20 most related documents	Dynamically adjusted according to the document similarity scores.
# of returned MeSH terms	Top 25 ranked MeSH terms	Automatic cutoff method to select most relevant MeSH terms from the top 25.
Features used in the learning-to-rank algorithm	Neighborhood features, word unigram/bigram overlap features, etc.	Features in previous work plus the MTIFL results as a knowledge feature.
MeSH version	MeSH 2010	MeSH 2013
Training data for the learning-to-rank algorithm	200 MEDLINE documents (2002-2009)	1000 documents from select BioASQ Journal List (2013)
Training data for the translation background language models	13,999 MEDLINE documents	58,088 MEDLINE documents
Additional rules	None	“Annual Reports as Topic” et al. for specific European journals

3 Results

3.1 Datasets

The 2013 BioASQ Task 1a ran for three consecutive periods (batches) of 6 weeks each. For each week, the BioASQ organizers distributed new unclassified PubMed documents, and participants have a limited response time (less than one day) to return their predicted MeSH terms. As new manual annotations become available, they were used to evaluate the classification performance of participating systems.

BioASQ provided approximately 11 million PubMed documents as the training set data. Since all existing PubMed documents can be used as training data, we randomly selected a set of 1000 MEDLINE documents from the list of the journals provided by BioASQ. We did not use the training set Small200 in our previous work because the corresponding MeSH terms in the new training set were assigned more recently (in 2013), and the articles in the BioASQ datasets were published in that select list of journals. We also reprocessed the documents in the datasets using MeSH 2013 (vs MeSH 2010 in our previous work).

3.2 Evaluation Metrics

The BioASQ Task 1a assessed the performance of the participating systems based on two measures. One is the flat measure “label-based micro F-measure” and the other is the hierarchical measure “Lowest Common Ancestor F-measure (LCA-F)”.

For reasons of completeness, BioASQ also provided the evaluation results using Micro Precision (MiP), Micro Recall (MiR), Lowest Common Ancestor Precision (LCA-P), Lowest Common Ancestor Recall (LCA-R), and some other metrics.

3.3 Comparison of different methods

Here we present the results on the BioASQ Task 1a Batch 2 Week 5 dataset. This dataset contains the largest number of curator-annotated articles (2,556) out of all 6 test sets of Batch 2, as of October 11, 2013. We submitted the results in five different runs. For the “MeSH Indexing Add” run, the list of MeSH terms was truncated using the automatic cut-off method described above, while the number of neighbors was set to 40 which performed best in our previous work. For the “MeSH Indexing Pre” run, the number of neighbors was adjusted to be 50. For the “MeSH Indexing New” run, the number of neighbors was dynamically adjusted according to the similarity scores between a target document and its neighbor documents. For the “MeSH Indexing” run, we used the results of MTIFL as the feature of domain-specific knowledge. For the “MeSH Indexing Ref” run, the learning-to-rank algorithm used is MART. Official results for our systems and the top 10 best runs from 11 different participating systems are shown in Table 2. We also added the results based on our previous work as “Huang et al., 2011” for comparison.

Table 2. Official results for the top 10 best runs on Batch 2 Week 5 test set plus the results for our previous approach (Huang et al., 2011). Our five runs are presented as underlined text.

Systems	MiF	MiP	MiR	LCA-F	LCA-P	LCA-R
system1	0.5677	0.5684	0.5670	0.4759	0.4946	0.4866
<u>MeSH Indexing Ref</u>	0.5471	0.5530	0.5412	0.4617	0.5029	0.4596
MTI First Line Index	0.5449	0.5992	0.4997	0.4582	0.5239	0.4379
<u>MeSH Indexing</u>	0.5298	0.5052	0.5568	0.4595	0.4821	0.4782
<u>MeSH Indexing Add</u>	0.5223	0.4733	0.5826	0.4563	0.4428	0.5070
<u>MeSH Indexing Pre</u>	0.5218	0.4729	0.5821	0.4570	0.4464	0.5035
<u>MeSH Indexing New</u>	0.5217	0.4772	0.5754	0.4568	0.4609	0.4947
Wishart-S2	0.4993	0.5319	0.4127	0.4207	0.5103	0.3809
Wishart-S5	0.4976	0.6277	0.4122	0.4204	0.5080	0.3814
Wishart-S4	0.4840	0.5886	0.4109	0.4222	0.5098	0.3833
Huang et al., 2011	0.4312	0.3248	0.6408	0.3996	0.3123	0.5546

As shown in Table 2, all our five runs ranked in the top 10 results with our best run "MeSH Indexing Ref" ranked the second. Our five runs also show substantial improvement of precision and F-measure over our previous approach (Huang et al., 2011), while the choice of top 25 MeSH terms in our previous work resulted in higher recall. Although including fewer neighbors trades recall for precision, considering fewer neighbors for articles without very similar neighbors can improve the performance in terms of Lowest Common Ancestor F-measure (LCA-F 0.4568 over LCA-F 0.4563).

Moreover, the Micro F-measure of the MART algorithm used in "MeSH Indexing Ref" (0.5471) represents a 4.85% improvement over that of the ListNet algorithm used in our previous work and "Mesh Index Pre" (0.5218). We also note that using the MTIFL's output as a feature in "Mesh Indexing" slightly outperformed the results from those without using this feature (Micro F-measure 0.5298 over Micro F-measure 0.5218).

4 Discussion

In spite of the prevalence that common terms are often found in similar documents, the NLM indexers may have their own rules for handling special cases. For instance, as we observed in the human results, all the articles in some European journals are annotated with "Annual Reports as Topic", "Europe", "Humans", "Publications" regardless of the language and content of the article. The nearest neighbor methods should address these special cases with pre-defined rules.

It is also important to note that we observed many redundant or contradictory Check tags (e.g., Human, Male, Female) in our returned results. This is because that the Check tags can appear in the neighbor documents frequently, e.g., an article describing about a disease in males might have many similar documents discussing about the same disease in females, which will result attaching the undesirable Check tag "Female" to that article. However on the other hand, it is improper to simply exclude the tag "Female" if "Male" already exists, because there are about 4 million articles in PubMed that have both "Male" and "Female" as MeSH terms. We believe there is a potential to further improve our precision by filtering out those incorrect Check tags accordingly.

5 Conclusion

We described a ranking-based approach for the BioASQ 2013 challenge on large-scale online biomedical semantic indexing. Our results were among the top-performing teams in both hierarchical and flat measures. These results were obtained using learning-to-rank algorithms for MeSH terms obtained from neighbor articles. We find that the approach with automatic truncation outperforms our previous approach. Additionally, we find that the use of domain knowledge features based on the MTI output provided better results. Finally, the learning-to-rank algorithm MART, appeared to be the best tree-based boosting algorithm for classification on the Bio-

ASQ dataset. In conclusion, our learning-to-rank method used in combination with *k*-nearest neighbor and domain-specific knowledge is a competitive approach for the automatic MeSH indexing task.

Acknowledgements

We would like to thank the authors of MTIFL, and in particular Jim Mork, for helpful discussion. We also thank Robert Leaman for proofreading the manuscript. This research is supported by the NIH Intramural Research Program, National Library of Medicine.

References

1. Lu, Z., Kim, W., Wilbur, W.J.: Evaluation of query expansion using MeSH in PubMed. Information retrieval 12, 69-80 (2009)
2. Névéol, A., Doğan, R.I., Lu, Z.: Author keywords in biomedical journal articles. In: AMIA Annual Symposium Proceedings, pp. 537. American Medical Informatics Association, (2010)
3. Doğan, R.I., Lu, Z.: Click-words: learning to predict document keywords from a user perspective. Bioinformatics 26, 2767-2775 (2010)
4. Huang, M., Névéol, A., Lu, Z.: Recommending MeSH terms for annotating biomedical articles. Journal of the American Medical Informatics Association 18, 660-667 (2011)
5. Kim, W., Aronson, A.R., Wilbur, W.J.: Automatic MeSH term assignment and quality assessment. In: Proceedings of the AMIA Symposium, pp. 319. American Medical Informatics Association, (2011)
6. Huang, M., Lu, Z.: Learning to annotate scientific publications. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 463-471. Association for Computational Linguistics, (2010)
7. Névéol, A., Shooshan, S.E., Humphrey, S.M., Mork, J.G., Aronson, A.R.: A recent advance in the automatic indexing of the biomedical literature. Journal of biomedical informatics 42, 814-823 (2009)
8. Yuqing Mao, K.V.A., Donghui Li, Cecilia N. Arighi, Zhiyong Lu: The Gene Ontology Task at BioCreative IV. In: Proceedings of the BioCreative IV workshop. (2013)
9. Lin, J., Wilbur, W.J.: PubMed related articles: a probabilistic topic-based model for content similarity. BMC bioinformatics 8, 423 (2007)
10. Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the 24th international conference on Machine learning, pp. 129-136. ACM, (2007)
11. Liu, T.-Y.: Learning to rank for information retrieval. Foundations and Trends in Information Retrieval 3, 225-331 (2009)
12. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of Statistics 1189-1232 (2001)