Ranking Medical Subject Headings using a factor graph model

Wei Wei[1], Dina Demner-Fushman[2], Shuang Wang[1], Xiaoqian Jiang[1], Lucila Ohno-Machado[1]

[1]Division of Biomedical Informatics, University of California, San Diego, La Jolla, CA 92093 USA,
Email: {w2wei, shw070, x1jiang, lohnomachado}@ucsd.edu
[2]National Center for Biomedical Communications, National Library of Medicine, Bethesda, MD, 20894
Email: ddemner@mail.nih.gov

Abstract

Automatically assigning MeSH (Medical Subject Headings) to articles is an active research topic. Recent work demonstrated the feasibility of improving the existing automated Medical Text Indexer (MTI) system, developed at the National Library of Medicine (NLM). Encouraged by this work, we propose a novel data-driven approach that uses semantic distances in the MeSH ontology for automated MeSH assignment. Specifically, we developed a graphical model to propagate belief through a citation network to provide robust MeSH main heading (MH) recommendation. Our preliminary results indicate that this approach can reach high Mean Average Precision (MAP) in some scenarios.

INTRODUCTION

MeSH is a controlled vocabulary thesaurus used at the NLM for indexing biomedical literature. MeSH indexing improves literature retrieval and it is widely used in biomedical text mining (1). The NLM indexers generally assign 5 to 15 MH to every article using their domain knowledge (2). This process is assisted by the automated MTI system developed at the NLM (3,4). Currently, around 65% of MHs are suggested by MTI.

Automatically assigning MHs was recently a task in the international BioASQ challenge (5). The two winning teams were able to improve the strong baseline provided by MTI using supervised machine learning methods; particularly, learning to re-rank the original MTI results (6). We were therefore motivated to explore a novel machine learning method for finding relevant MHs and providing more accurate suggestions. We investigated a factor graph based approach that uses the hierarchical structure of the MeSH ontology and semantic distance metrics in a case study. The performance of our preliminary model is close to MTI, which considers more attributes such as the abstract and the title.

BACKGROUND

MTI generates a ranked list of MHs, Subheadings and CheckTags as a final result from the title and the abstract of every article, using a combination of two indexing methods: PubMed Related Citations and MetaMap indexing (7). PubMed Related Citations, an implementation of the *k*-Nearest Neighbor (*k*-NN) algorithm, produces a list of related articles; MetaMap maps text from the titles and abstracts of the articles to the UMLS Metathesaurus. The results of the two methods are then clustered and ranked through a post-processing phase (3). After that, the indexers review MTI suggestions and select the appropriate main headings from the pool. The *k*-NN algorithm on its own has outperformed several other approaches in the experiments on 1000 randomly selected MEDLINE citations (8). There are re-ranking algorithms applied to the MTI output (6) to adjust suggestions. Similar ideas have been also applied to the results of the PubMed Related Citations algorithm (9), as well as the multi-label ensemble method consisting of the SVM classifier and the Latent Dirichlet Allocation (LDA) model (10). These algorithms can outperform the strong MTI baseline by about 10%.

Graphic models, which generally include directed graphs, undirected graphs, and factor graphs, are powerful tools for representing probabilistic models. Factor graphs, which can represent both directed and undirected graphs, have gained popularity as they offer great flexibility for problem solving. Although both directed and undirected graphs allow representation of a global function with multiple variables as a product of factors over subsets of variables, a factor graph provides an explicit way to factorize the global function by introducing variable and factor *nodes*. The introduction of factor nodes allows the optimization algorithms (e.g., belief propagation) to be derived in a simple and general form, because the factor graph unifies the directed and undirected graph with the same representation. A factor graph is a bipartite graph that consists of two types of nodes, i.e., factor and variable nodes. Each factor node needs to connect at least one variable node and vice versa. A variable node represents a hidden variable or an observation in an inference problem. A factor node captures the factor function of the variables in the connected variable nodes. For an acyclic graph, a factor function corresponds to a factor of the decomposed joint probability. For example, Figure 1 depicts the factor graph representation of a decomposed joint probability $P(x, y) = P(x \mid y)P(y)$, where factor nodes and variable nodes are denoted by squares and circles, respectively. The factor nodes F_1 and F_2 capture the prior probability $P(y)$ and the conditional probability $P(x \mid y)$, respectively. The variable nodes V_1 and V_2 represent the hidden variable x and the observation y, respectively.

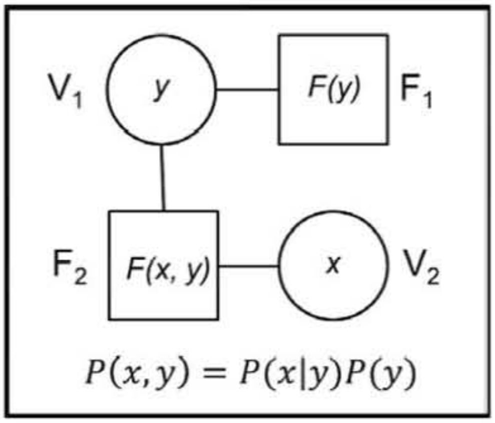

Figure 1. An example of a factor graph representation of a joint probability p(x, y), where factor nodes and variable nodes are denoted by squares and circles, respectively. In this pilot study, we are exploring a factor graph model to represent our knowledge about a corpus of articles related to Kawasaki disease (Kawasaki corpus).

METHODS

Dataset preparation

To evaluate the performance of the factor graph model, we created a Kawasaki disease corpus of 770 PMC articles with automatically extracted references, authors, journal titles, and PMID information. From this corpus, we randomly selected 20 original research articles and manually verified information of MHs, references, and MHs of reference articles. In this model, every article and its reference articles form a factor graph; two graphs are independent from each other. Thus, we can run multiple models in parallel to deal with larger datasets.

Knowledge representation

We used variable nodes to represent PubMed articles; every node has pre-defined attributes to store the knowledge about that article, restricted to the belief distribution on MHs in this report. Variable nodes were connected via intermediary factor nodes according to their citation relations. These intermediary factor nodes summarize beliefs of MHs from adjacent variable nodes and then pass this information to neighbors and update their beliefs on the same MHs. A leaf factor node provided the prior belief distribution on MHs of the connected variable node; MHs that appeared in the corresponding article were assigned greater probabilities while the other MHs received small but non-zero prior probabilities. It is common to see loops in citation networks. In this study, we focused on two-layer tree-structured factor graphs, so we did not have to consider loops. However, it is worth mentioning that the proposed model is able to deal with loops in factor graphs, which refer to the loopy belief propagation algorithm. Although loopy BP will introduce approximation in the results, many existing studies (11) show that loopy BP shows good converge performance. Considering the computation complexity, an intermediary factor node only connected two variable nodes, and a leaf factor node connected one variable node. All cited articles contribute equally to the citing article, given the graph structure and the inference algorithm introduced below.

Inference

We used the sum-product algorithm (a.k.a, belief propagation algorithm) (12,13) to infer the marginal probability on every MH. The sum-product algorithm is an efficient method to compute the exact marginal probability of each variable in an acyclic graph. The sum-product algorithm converges efficiently with acyclic graphs; for graph with cycles, it also provides a good performance in many applications such as image processing. In general, the sum-product algorithm includes three steps.

Step 1: Belief update about each variable: The belief about each variable is the estimated marginal probability of the given variable. In the case of a tree structured graph, the belief is identical to the exact marginal probability once the algorithm converges. The belief update follows equation [1].

$$b(x_i) = \prod_{F_j \in \mathcal{N}(V_i)} m_{F_j \to V_i}(x_i) \qquad [1]$$

where i, j stand for the ith and jth nodes in the factor graph; x_i is the variable represented by the variable node V_i; $\mathcal{N}(V_i)$ is a set of all neighboring factor nodes of V_i; $m_{F_j \to V_i}(x_i)$ is the message sent from adjacent factor node F_j to variable node V_i .

Step 2: Variable node update: The message that will be sent from variable node V_i to an adjacent factor node F_j can be calculated as [2]

$$m_{V_i \to F_j}(x_i) = \frac{b(x_i)}{m_{F_j \to V_i}(x_i)} \qquad [2]$$

Step 3: Factor node update: The message that needs to be sent from factor node F_j to it neighbor variable node V_i can be evaluated as [3]

$$m_{F_j \to V_i}(x_i) = \sum_{V_k \in \mathcal{N}(F_j) \setminus V_i} f(\boldsymbol{x}_s) \prod_{V_k \in \mathcal{N}(F_j) \setminus V_i} m_{V_k \to F_j(x_i)} \qquad [3]$$

Where $\mathcal{N}(F_j)$ is a set of all neighboring variable nodes of F_j; $\boldsymbol{x}_s$ is a set of variables represented by the variable nodes in $\mathcal{N}(F_j)$; $f(\boldsymbol{x}_s)$ is the factor function; $\mathcal{N}(F_j) \setminus V_i$ denotes a set of all the neighboring nodes excluding node V_i.

These three steps are repeated until the beliefs converge.

Design of the factor function

The performance of the model largely depends on the design of factor functions. Here, we only consider the MeSH ontology based semantic correlations for estimating the final marginal probabilities. The factor function is a monotonically decreasing function of the semantic correlations. We experimented with different functions from five families: exponential, tangent, arctangent, logarithm, and linear.

Evaluation

We compared the prediction to gold standards in terms of precision, recall, and Mean Average Precision (MAP), which was the mean of the precision scores obtained after each relevant document was retrieved (14–16). The five metrics are defined as below:

$$\text{Precision}=\frac{\sum_D c(N,D,H_1^N)}{\sum_D N}, \ \text{Recall}=\frac{\sum_D c(N,D,H_1^N)}{\sum_D AN(D)}, \ \text{F-score}=\frac{2*precision*recall}{precision+recall}$$

$$\text{AP(D)}=\frac{1}{AN(D)}\sum_r I(h_r)*\frac{c(r,D,H_1^r)}{r}, \ \text{MAP}(\Omega)=\frac{1}{|\Omega|}\sum_{D\in\Omega} AP(D)$$

H_1^N is a ranked list of top *N* MHs from factor graphs; $c(N,D,H_1^N)$ is the number of correct predictions among the top *N* MHs in document *D*; $AN(D)$ is the number of MHs assigned to *D* in gold standards; $AP(D)$ is the average precision; $I(h_r)$ is an indicator function, which returns 1 if rth MH in the prediction is in the gold standards and return 0 otherwise; Ω is the corpus of articles (15).

We implemented the above evaluation metrics. In addition, we used TREC_EVAL package version 9.0 (16) to calculate MAP. In this study, we had no plan to learn an optimal factor function, because we focused on the possibility of applying factor graph models to MH assignment.

RESULTS

Our two-layer factor graph model is illustrated in Figure 2.

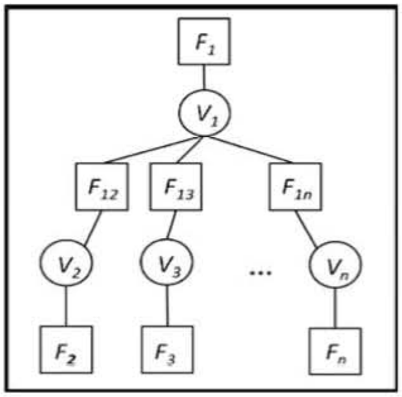

Figure 2. An example of a factor graph with two layers of variable nodes. V_1 is a variable node which represents the citing article. V_2 to V_n are the variable nodes for cited articles. F_1 to F_n are leaf factor nodes and they provide the prior probability distributions on MHs for their adjacent variable nodes. F_{12} to F_{1n} are intermediary factor nodes.

To determine a factor function in this study, we explored five factor functions (Table 1) from four different families and evaluated their MAP scores on five manually verified articles.

Factor Functions
$f(x) = \exp(-x^2)$, $f(x) = \exp(-x^3)$,
$f(x) = -\ln(x)$, $f(x) = -\frac{1}{x}$, $f(x) = \arctan(x)$

Table 1. Candidate factor functions. They are basic functions selected from four families. Variable x is the semantic distance between two MHs in the MeSH ontology.

Five articles were selected from the Kawasaki corpus as shown in Table 2. We manually verified all the selected articles and corrected metadata and reference information collected from automated extraction. On every article, we built factor graph models with five factor functions and evaluated the MAP scores using Trec_eval 9.0 package. Since the exponential functions generally performed better than other functions in this testing, we selected $\exp(-x^2)$ with considerations of further extension. In future work we will consider learning factor functions from data, such as using an iterative log-linear regression method.

	Citing Article PMID	MAP*	Selected Factor Function
1	11953819	0.6266	$\exp(-x^3)$
2	15611788	0.2487	$\exp(-x^2)$
3	11875736	0.3571	$\exp(-x^2)$
4	16202147	0.8889	$\exp(-x^2)$
5	9874566	1	$\arctan(x)$
6	9874566	1	$-\ln(x)$

***These MAP scores were calculated using Trec_eval 9.0 package. Each input file contains only one article with all its available MHs.**

Table 2. Model performance with different factor graphs. The last column is the factor function used, among six candidates.

On a set of 20 verified articles, we obtained precision, recall, F score, and average precision (AP) in Table 3.

PMID	Precision	Recall	F score	AP
9874566	0.44	0.52	0.48	0.05
11875736	0.20	0.29	0.24	0.29
11953819	0.56	0.78	0.65	0.28
12556969	0.24	0.33	0.28	0.22
12671708	0.36	0.75	0.49	0.33
12823849	0.12	0.60	0.20	0.40
14676801	0.40	0.63	0.49	0.25
15611788	0.20	0.63	0.30	0.63
15928668	0.44	0.53	0.48	0.24
16202147	0.56	0.58	0.58	0.33
16404364	0.52	0.65	0.58	0.55
16594731	0.20	0.38	0.26	0.23
16965625	0.56	0.78	0.65	0.44
17640353	0.24	0.32	0.28	0.21
18070342	0.40	0.56	0.47	0.11
18171482	0.52	0.73	0.60	0.50
18387181	0.44	0.69	0.54	0.38
18782781	0.60	0.79	0.68	0.05
19065999	0.16	0.29	0.21	0.07
19264792	0.52	0.76	0.62	0.47

Table 3. Outcomes from 20 articles.

Based on Table 3, we evaluated the mean Precision, mean Recall, mean F score and MAP as shown in Table 4, following the formula in the evaluation section.

Precision	Recall	F score	MAP
0.38	0.58	0.46	0.30

Table 4. Mean values of precision, recall, F score and average precision.

DISCUSSION

Huang et al. (15) reported a precision of 0.302, recall of 0.583, F score of 0.398, and MAP of 0.462 of MTI on a dataset of 1000 randomly selected MEDLINE documents. We learned from the NLM that the estimated MAP of MTI is 0.35. Considering that MTI adopted multiple attributes such as nearest neighbors and text from titles and abstracts, the factor graph model has shown the potential for providing better ranked MH suggestions.

The performance of the factor graph model depends on multiple factors, including the design of the factor function, the attributes, the type of article, number of references, and the MeSH vocabulary of a particular corpus. In future studies, we will extend the model and incorporate attributes of journal and author, because a journal has a strong association with the topics of its articles and authors usually have very specific research fields. Other attributes are also in consideration, as long as it can better represent the relations between articles and improve the model performance.

Some MHs do not occur in the cited articles, but they could be derived from the text of the citing article. We will use MetaMap to map text in the citing articles to UMLS terms, and identify potential MHs from these UMLS terms, a solution similar to the one used in MTI. This also shed a light on resolving a limitation of this study. Currently, the factor graph models are with articles in PMC with complete references. However, not all articles are indexed by PMC and references may be incomplete. In the case that reference articles containing desired MHs are missed from data, it is possible to recover these MHs using the above natural language processing techniques.

Conclusion

In this pilot study, we experimented with the factor graph model and sum-product algorithm to infer MHs on a Kawasaki disease corpus from PubMed. The results warrant the further investigation using this technique to improve the prediction performance.

ACKNOLEDGEMENT

This work is supported by NIH grant (U54HL108460), NLM (R00LM011392, R21LM012060) and NHGRI (K99HG008175), in part, by the Medical Informatics Training Program at the Lister Hill National Center for Biomedical Communications, NLM.

REFERENCE

1. Zweigenbaum P, Demner-Fushman D, Yu H, Cohen KB. Frontiers of biomedical text mining: current progress. Brief Bioinform. 2007 Sep 1;8(5):358–75.

2. Understanding the Vocabulary. U.S. National Library of Medicine. Available from: http://www.nlm.nih.gov/bsd/disted/pubmedtutorial/015_010.html

3. Mork JG, Yepes AJJ, Aronson AR. The NLM Medical Text Indexer System for Indexing Biomedical Literature. Proceedings of the first Workshop on Bio-Medical Semantic Indexing and Question Answering, a Post-Conference Workshop of Conference and Labs of the Evaluation Forum 2013. Valencia, Spain; 2013.

4. Mork JG, Demner-Fushman D, Schmidt SC, Aronson AR. Recent enhancements to the NLM medical text indexer. Working Notes for CLEF 2014 Conference. Sheffield, UK; 2014.

5. What BioASQ Is About [Internet]. Available from: http://www.bioasq.org/project/about

6. Mao Y, Wei C-H, Lu Z. NCBI at the 2014 BioASQ challenge task: large-scale biomedical semantic indexing and question answering. Proceedings of Question Answering Lab at CLEF. 2014.

7. Aronson AR, Bodenreider O, Chang HF, Humphrey SM, Mork JG, Nelson SJ, et al. The NLM Indexing Initiative. Proceedings of the AMIA Symposium American Medical Informatics Association. 2000. p. 17–21.

8. Trieschnigg D, Pezik P, Lee V, de Jong F, Kraaij W, Rebholz-Schuhmann D. MeSH Up: effective MeSH text classification for improved document retrieval. Bioinformatics. 2009 Jun 1;25(11):1412–8.

9. Liu K, Wu J, Peng S, Zhai C, Zhu S. The Fudan-UIUC Participation in the BioASQ Challenge Task 2a: The Antinomyra system. CLEF (Working Notes). 2014. p. 1311–8.

10. Papanikolaou Y, Dimitriadis D, Tsoumakas G, Laliotis M, Markantonatos N, Vlahavas IP. Ensemble Approaches for Large-Scale Multi-Label Classification and Question Answering in Biomedicine. CLEF (Working Notes). 2014. p. 1348–60.

11. Ihler AT, Fisher JWI, Willsky AS. Loopy Belief Propagation: Convergence and Effects of Message Errors. Journal of Machine Learning Research. 2005. p. 905–36.

12. Kschischang FR, Frey BJ, Loeliger H-A. Factor graphs and the sum-product algorithm. IEEE Trans Inf Theory [Internet]. IEEE Press; 2001;47(2):498–519.

13. Yedidia JS, Freeman WT, Weiss Y. Understanding belief propagation and its generalizations. Morgan Kaufmann Publishers Inc.; 2003 Jan 1;239–69.

14. Buckley C, Voorhees EM. Retrieval evaluation with incomplete information. Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04 [Internet]. New York, New York, USA: ACM Press; 2004. p. 25.

15. Huang M, Névéol A, Lu Z. Recommending MeSH terms for annotating biomedical articles. J Am Med Inform Assoc;18(5):660–7.

16. Trec_eval09 package [Internet]. [cited 2014 Sep 3]. Available from: http://trec.nist.gov/trec_eval/