Topic modeling of biomedical text

From words and topics to disease and gene links

Sarah ElShal, Mithila Mathad, Jaak Simm, Jesse Davis, and Yves Moreau
Department of Electrical Engineering (ESAT)
iMinds Future Health Department
Department of Computer Science (DTAI)
KU Leuven, Belgium
sarah.elshal@esat.kuleuven.be

Abstract— **The massive growth of biomedical text makes it very challenging for researchers to review all relevant work and generate all possible hypotheses in a reasonable amount of time. Many text mining methods have been developed to simplify this process and quickly present the researcher with a learned set of biomedical hypotheses that could be potentially validated. Previously, we have focused on the task of identifying genes that are linked with a given disease by text mining the PubMed abstracts. We applied a word-based concept profile similarity to learn patterns between disease and gene entities and hence identify links between them. In this work, we study an alternative approach based on topic modelling to learn different patterns between the disease and the gene entities and measure how well this affects the identified links. We investigated multiple input corpuses, word representations, topic parameters, and similarity measures. On one hand, our results show that when we (1) learn the topics from an input set of gene-clustered set of abstracts, and (2) apply the dot-product similarity measure, we succeed to improve our original methods and identify more correct disease-gene links. On the other hand, the results also show that the learned topics remain limited to the diseases existing in our vocabulary such that scaling the methodology to new disease queries becomes non trivial.**

Keywords- text analysis; pattern recognition; machine learning; topic modelling; disease-gene linkage

I. INTRODUCTION

Text mining PubMed abstracts is popular, especially because approximately 500,000 new citations are added to PubMed each year [1]. This huge amount of text has motivated interest in going beyond simple keyword search of PubMed to automatic extraction of knowledge or information from the abstracts. One line of work attempts to automatically generate biomedical hypotheses which can then be used to guide laboratory experiments. Examples of this include identifying links between biomedical entities of interest, such as genes and diseases or targets and drugs [2 - 9].

These methods involve techniques that rely on co-occurrence [2, 3, and 8] concept profile similarity [4, 5], classification models [6], or rule-based strategies [7]. Previously, we applied a combination of word-based co-occurrence and concept profile similarity to extract links between diseases and genes [9]. This work used all words extracted from the PubMed abstracts to generate disease and gene profiles. However, this set of words was noisy, and a more refined and compact representation for the profiles, could lead to improved performance for the disease-gene learning problem. This could be achieved by clustering similar words into grouped sets, or topics, or assigning higher weights to more important words in a profile.

Topic modelling is an unsupervised learning technique that identifies a set of unobserved topics, or variables, inside an input set of documents [10]. It can be viewed as a way of mapping documents represented by a large set of words into documents represented, or modelled, by a smaller set of topics. It is based on the idea that a document is a mixture of topics, which are in turn a probability distribution over words. Many approaches exist for learning these probabilities such as Latent Dirichlet Allocation (LDA), which estimates a posterior distribution of words and topics given an input corpus [11]. Starting from a bag-of-words representation of the documents, LDA can learn the topic distributions by applying the necessary Bayesian models. This involves for example the application of Gibbs sampling, which learns such distributions in an iterative approach. More specifically, this involves selecting a few parameters, such as the number of topics, the number of iterations, and the Dirichlet priors.

Previously topic modelling has been studied to compute similarities between documents, between words, or between documents and words [12]. Intuitively, if two documents' topic distributions are similar, then there is probably a correlation between them. Such similarity can be measured by means of the Kullback-Leibler (KL) divergence, Jensen-Shannon (JS) divergence, or simply by means of a dot product. Document or word similarity can be used in information retrieval applications where the goal is to retrieve the most relevant documents to a query. This can be achieved by decomposing the query into a set of words and then averaging the similarity over all the decomposed words with the documents in question. We illustrate this Fig.1.

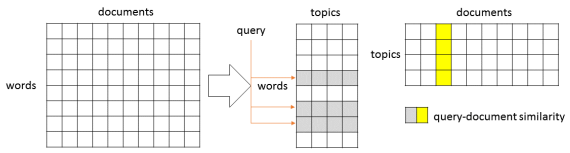

Figure 1 Topic modeling transfers the initial bag-of-words representation of documents, documentXword matrix on the left hand side, into topic representations of words and documents, wordXtopic matrix and topicXdocument matrix on the right hand side. We can use this in information retrieval to compute query-document similarities.

Topic modelling has been applied in other applications beyond documents and words. For example it has been used to classify genomic sequences, where DNA sequences are mapped to documents, and fixed-sized windows of sequences are mapped to words [13]. It has also been used to classify drugs according to safety and therapeutic use, which relies on an initial ambiguous drug labelling representation [14].

In this work we investigate the application of topic modelling in finding links between genes and diseases. Opposed to existing word-based approaches which measure disease-gene similarity based on their bag-of-words representations [15], this paper proposes measuring the similarity based on the disease and gene representations in the topic space. Measuring the similarity in the topic space is interesting since the genes and diseases are mapped to better clustered and less dimensioned representations, which imply simpler and less noisy computations compared to word-based similarities.

We developed this work in two phases. In the first phase we investigated the application of topic modeling in general inside our problem. This involved testing different LDA parameters and multiple similarity measures. In the second phase we additionally investigated varying the input corpus such that it incorporates different sets of gene-related abstracts. We also varied the similarity measure and observed the impact on the learned similarities. Based on our experiments we find that every factor we investigated to generate the topics and measure the similarity plays an important role in the learning process such that only when we arrange the right settings we are able to improve the original methods.

II. MATERIAL AND METHODS

A. *The data sets*

We use PubMed as our source of biomedical text. Given a set of disease and gene entities, we extract all the abstracts that are linked to each of them. We use GeneRIF [16] to identify abstracts linked to each gene and we use the PubMed search engine to identify abstracts linked to each disease. In the first phase of this work, we relied on data downloaded in May 2012. This corresponds to 282,460 GeneRIF abstracts and 16,493 genes. In the second phase, we relied on data downloaded in March 2015. This corresponds to 349,274 GeneRIF abstracts and 17,116 genes. For each abstract, we generated a bag-of-words representation using MetaMap [17] in the first phase or EXTRACT annotations of PubMed [18] in the second phase. We only consider words present in the GeneRIF corpus, which results in 66,884 words in the first phase and 73,027 words in the second phase. We use OMIM [19] to identify a list of experimentally validated disease-gene entities. This corresponds to 314 diseases, 2,055 genes, and 2654 disease-gene pairs in the first phase, and 330 diseases, 2,214 genes, and 2,789 disease-gene pairs in the second phase. We summarize our data sets in Table 1.

TABLE 1 SUMMARY OF THE DATA SETS DOWNLOADED FROM GENERIF AND OMIM

First phase	Number of entities	Downloaded in
GeneRIF abstracts	282,460	05/2012
GeneRIF genes	16,493	05/2012
GeneRIF words	66,883	05/2012
OMIM diseases	314	07/2013
OMIM genes	2,055	07/2013
OMIM disease-gene pairs	2,654	07/2013
Second Phase	Number of entities	Downloaded in
GeneRIF abstracts	349,274	03/2015
GeneRIF genes	17,116	03/2015
GeneRIF words	73,027	03/2015
OMIM diseases	330	05/2015
OMIM genes	2,214	05/2015
OMIM disease-gene pairs	2,789	05/2015

B. *LDA and choosing the parameters for our problem*

In this work, we employ the LDA model described using the plate notation as shown in Fig. 2. We observe that this model relies on two Dirichlet priors, α and β. The α parameter controls the topic distributions θ for each document d in D. The β parameter controls the topic-word distributions φ. For more details about the model, we refer the reader to [12].

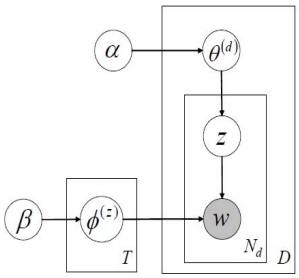

Figure 2 LDA model described using plate notation

In the first phase of this work, we employed the Java implementation of LDA (JGibbLDA) [20] to learn the topic and topic-word distributions and ultimately generate the topics profiles for our disease and gene entities. This implementation relies on Gibbs sampling to learn the distributions, which requires the following parameters. K: number of topics, β and α, the Dirichlet priors, and N, number of iterations. In this phase we tested wide ranges of each parameter (e.g. 300<K<5000 and 1000<N<7000). Here we also tested different similarity measures (e.g. KL and JS).

In the second phase we applied the Matlab Topic Modeling toolbox 1.4 (GibbsSamplerLDA) [21]. This toolbox also relies on Gibbs sampling and requires similar parameters (T: number of topics, β and α, and N). Here, we apply the parameters shown in Table 2. We choose T=5000 and N=1000 since they were the most convenient according to the first phase performance and computation wise (see Results). We choose β=200/W, where W denotes the number of words, and α=50/T, since this was recommended by the toolbox given previous experiments.

TABLE 2 PARAMETERS FOR GIBBSSAMPLERLDA

T	5000
β	200/73,027
α	50/5000
N	1000

C. *Experimental setup*

We use the gene-to-abstract links according to GeneRIF, the disease-to-abstract links according to MEDLINE, and the abstract-to-word links according to EXTRACT to generate the gene-to-word and the disease-to-word links, which correspond to the bag-of-word profiles for each gene and disease entity. In our previous work, we used the Term Frequency – Inverse Document Frequency (TF-IDF) transformation to numerically represent each bag-of-word profile. Then we used the cosine-similarity to score how similar a given gene is to the disease in question. Finally, we ranked all the genes for each disease in question and measured the True Positive Rate (TPR). We focused on early discovery where we measured the TPR in the top 10, 25, 50, and 100 ranking genes, which corresponds top 0.6% of the ranking genes. For more details on our previous work we refer the reader to our earlier publications [9, 15]. We are mainly interested in early discovery since it has been proven that users rarely go to second page results, especially for web queries [22], which implies that users rarely check more than 10 results given any search query. Hence we want to maximize the number of correct results in the early ranks, which the users are mostly interested in.

In the first phase of this work we were mainly investigating the application of topic modeling in our problem. As shown on the left of Fig.3, we started from the bag-of-word profiles of genes as our input corpus, numerically represented by TF-IDF values. Note that here we rely on MetaMap to generate the bag-of-word profiles. We ran JGibbLDA to learn the topic distributions as shown on the right of Fig. 3. We mapped the diseases to words in our corpus and computed the disease-gene similarities in the topic space. As briefly introduced, in this phase we tested multiple ranges of K and N, and compared the similarities using KL, JS, and the cosine similarity. Note that all the similarities are based on the dot product, which can be computed in the topic space as in (1):

$$P(d \mid g_i) = \sum_{j=1}^{K} P(w_d \mid z = j) P(z = j \mid g_i)$$

Equation 1: where d corresponds to the disease, g_i to a given gene i, w_d to the word mapped to the disease, K to the number of topics, and z to a given topic j

For more details about each similarity measure, we refer the reader to [12].

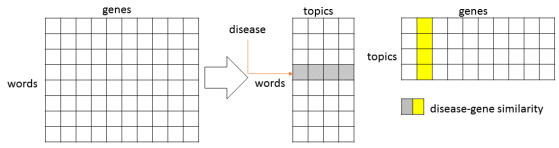

Figure 3 Topic modeling setup in the first phase - starting from the genes bag-of-word profiles, wordXgene matrix

In the second phase, we applied the best parameters resulting from the first phase, and investigated different input corpuses. We arranged two setups incorporating different input corpuses. In the first one we started from the bag-of-word profiles of genes, like in the first phase. This can be seen as a gene-clustered version of the PubMed abstracts. In the second one, as we show in Fig. 4, we directly started from the bag-of-word profiles of abstracts. Note that both corpuses are numerically represented by TF values, as required by GibbsSamplerLDA. Also note that here we rely on EXTRACT to generate the bag-of-word profiles as it proved better in previous experiments [15]. In the first setup, we proceeded like in the first phase, where we mapped the diseases to words in our corpus and computed the disease-gene similarities in the topic space. In the second setup, we mapped both the genes and diseases to our words, and then computed the disease-gene similarities. In both setups we measured the similarity using cosine similarity and dot product.

In both phases, based on the similarity scores, we ranked the genes for each disease, and we measured the TPR in the top ranking genes. We compared the TPR results to their counterparts in our previous work for the same disease entities.

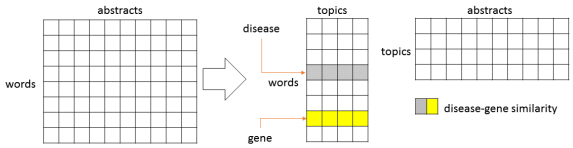

Figure 4 The extra setup in the second phase – starting from abstracts bag-of-words, wordXabstract matrix

III. RESULTS

We present the results of the first phase in Tables 3, 4, 5. We observe that in general increasing the number of topics and the number of iterations slightly enhances the performance. This corresponds to a TPR of 30% and 56% in the top 10 and the top 100 ranking genes when K=5000, and 28% and 60% when N=7000. Note that this slight enhancement comes with a major increase in the computation time, especially with a higher N. We also observe that the three similarity measures: KL, JS, and cosine similarity, achieve similar performance and only in the top 100 the TPR increases when applying the cosine similarity measure.

TABLE 3 TPR RESULTS FIRST PHASE – 300<K<5000

	top 10	top 25	top 50	top 100
Topic profiles K=300, cos-sim	28%	39%	47%	56%
Topic profiles K=400, cos-sim	29%	40%	49%	57%
Topic profiles K=500, cos-sim	29%	39%	46%	54%
Topic profiles K=5000, cos-sim	30%	40%	48%	56%

TABLE 4 TPR RESULTS FIRST PHASE – 1000<N<7000

	top 10	top 25	top 50	top 100
Topic profiles N=1000, cos-sim	28%	40%	49%	56%
Topic profiles N=2000, cos-sim	27%	39%	48%	56%
Topic profiles N=3000, cos-sim	27%	40%	48%	57%
Topic profiles N=7000, cos-sim	28%	40%	49%	**60%**

TABLE 5 TPR RESULTS FIRST PHASE – COMPARING KL, JS, AND COSINE SIMILARITY

	top 10	top 25	top 50	top 100
Topic profiles cos-sim	26%	37%	46%	**54%**
Topic profiles KL	28%	39%	44%	50%
Topic profiles JS	28%	37%	42%	44%

We present the TPR results of the first setup in the second phase, which starts from the word-by-abstract representation, in Fig.5. We observe that computing the similarities in the topics space by means of dot product resulted in a TPR of 36% and 42% in the top 10 and the top 100 ranking genes. These results are worse than using the bag-of-words space, which resulted in a TPR of 42% and 71% at the same ranking thresholds.

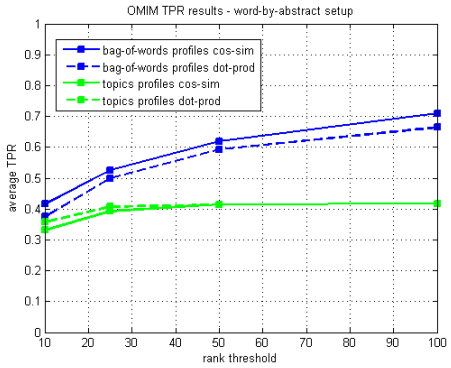

Figure 5 TPR results when we start from the bag-of-words of abstracts. We observe *worse* performance in the top 100 ranking genes when applying topic modeling inside the wordXabstract setup

We present the TPR results of the second setup in the second phase, which starts from the word-by-gene representation, in Fig.6. We observe that computing the similarities in the topics space using a dot product resulted in a TPR of 44% and 74% in the top 10 and the top 100 ranking genes. This improves over the bag-of-words representation.

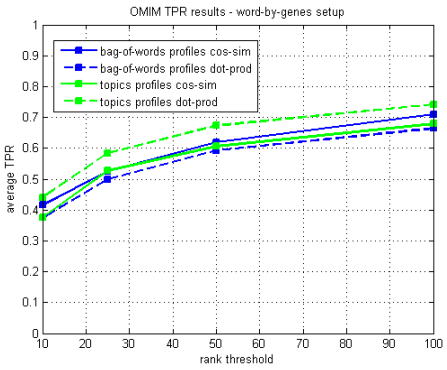

Figure 6 TPR results when we start from the bag-of-words of genes. We observe *better* performance in the top 100 ranking genes when applying topic modeling inside the wordXgene setup

We present a summary of the results of the second phase in Table 6. Compared to the best results from our previous methods, by measuring the cosine similarity on the bag-of-word profiles, we succeed to improve the recall by measuring the dot product on the topic profiles generated inside the word-by-gene setup.

TABLE 6 SUMMARY OF THE TPR RESULTS – SECOND PHASE

	top 10	top 25	top 50	top 100
Bag-of-words profiles, cos-sim	*0.4157*	*0.5251*	*0.6187*	*0.7092*
Bag-of-words profiles, dot-prod	*0.3746*	*0.4988*	*0.5927*	*0.6638*
Topic profiles, cos-sim, word-by-abstract setup	0.3307	0.3920	0.4144	0.4169
Topic profiles, dot-prod, word-by-abstract setup	0.3577	0.4072	0.4139	0.4175
Topic profiles, cos-sim, word-by-gene setup	0.3751	0.5275	0.6062	0.6794
Topic profiles, dot-prod, word-by-gene setup	**0.4407**	**0.5836**	**0.6739**	**0.7424**

IV. DISCUSSION

The application of topic modeling of biomedical text is interesting in linking genes with diseases. When applying the right parameter settings and similarity measure, we succeed to identify more correct disease-gene similarities. We achieve this using less number of dimensions, in the topic space, compared to the original high dimension similarities, computed in the bag-of-words space.

In our experiments in the first phase we varied multiple LDA parameters and observed that we slightly improve the performance when increasing the number of topics and the number of iterations. We also observed that the three similarity measures: KL, JS, and cosine similarity produce very similar TPR results. In our experiments in the second phase, we find improved disease-gene linkage when applying topic modelling on the bag-of-word profiles representing our genes, which corresponds to gene-clustered version of the abstracts. However, we fail to improve the linkage when we directly apply topic modelling on the bag-of-words profiles representing our abstracts. We believe the gene-clustered version is better as an input corpus since we somehow direct the learning to better model the gene entities, which are the focus of interest in our problem. This

is unlike starting from the individual abstracts representation where the gene entities are hidden inside the words. In our experiments in this phase we also observe that we only achieved similar results to our original methods when we used the cosine similarity measure. However, we could significantly improve them when using the dot product. We are not exactly sure why the dot product is better here but we believe further analysis can be helpful to formulate a reasonable hypothesis.

Nevertheless, we notice one major limitation in our setting for topic modelling in the broad context of disease-gene linkage. Since we map our diseases into the words inside the bag-of-word profiles, we are not able to apply the same model on more general disease queries that do not exist in our vocabulary. We can consider two options to overcome this. In the first one, we decompose our disease query into words that exist in our vocabulary, then we compute the average similarities. We already tried this, by starting from the TF-IDF profiles of a disease query, selecting the top 1 and 5 words that exist in our vocabulary and measuring the average similarity. We show the results in Fig.7. We observe that using the top word for a disease query approximates the performance of our original setting where we map the query to the exact word in the vocabulary, however when using the top 5 words the performance drops significantly. Although using the top word is a nice way out, it is still somehow limiting when the query is already composed of a combination of diseases. We see that using the top word here will probably be misleading if we consider only one of the disease words to compute the similarities against the genes.

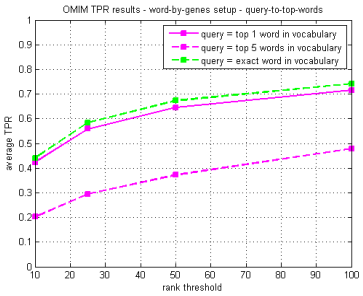

Figure 7 query-to-topWords TPR results. We observe worse performance when mapping the query into the top representative words in our vocabulary compared to using the exact match

In the second option, we include the disease queries in the learning process. Hence, we eventually have our queries and genes represented by topics, which we can directly use to compute the similarities. However, this is still limited by having to learn the topics distributions each time a new query is issued. This is undesired when we expect many new queries arriving daily and learning a model takes few days.

ACKNOWLEDGMENT

This work was supported by the Research Council KU Leuven [CoE PFV/10/016 SymBioSys, OT/11/051] to Y.M. and J.D.; the government agency for Innovation by Science and Technology to Y.M.; Industrial Research fund to Y.M.; Hercules Stichting to Y.M.; iMinds Medical Information Technologies [SBO 2015] to Y.M.; EU FP7 Marie Curie Career Integration Grant [\#294068] to J.D.; and FWO-Vlaanderen [G.0356.12] to J.D.

REFERENCES

[1] U.S. National Library of Medicine, "Yearly Citation Totals from 2015 MEDLINE/PubMed Baseline," 2016

[2] Fleuren WW, et al., "CoPub update: CoPub 5.0 a text mining system to answer biological questions," Nucl. Acids Res. 2011, 39

[3] Piñero J, et al., "DisGeNET: a discovery platform for the dynamical exploration of human diseases and their genes," Database 2015.

[4] Jelier R, Schuemie MJ, Roes PJ, van Mulligen EM, Kors JA, "Literature-based concept profiles for gene annotation: the issue of weighting." Int J Med Inform 2008, 77, 354-362.

[5] Cheung WA, Ouellette BF, Wasserman WW, "Inferring novel gene-disease associations using medical subject heading over-representation profiles," Genome Med. 2012, 4(9), 75

[6] Fontaine,J.F., Priller,F., Barbosa-Silva,A., Andrade-Navarro,M.A, "Genie: Literature-based gene prioritization at multi genomic scale," Nucl. Acids Res 2011, 39:W455-W461

[7] Hristovski,D., Friedman,C., Rindflesch,T.C., Peterlin,B., "Exploiting semantic relations for literature-based discovery," AMIA Annu Symp Proc. 2006, 349-53

[8] Wu Y., Liu M., Zheng W. J., Zhao Z., and Xu H., "Ranking gene-drug relationships in biomedical literature using Latent Dirichlet allocation," chapter 41, pages 422–433

[9] ElShal S, Tranchevent LC, Sifrim A, Ardeshirdavani A, Davis J, Moreau Y, "Beegle: from literature mining to disease-gene discovery," Nucl. Acids Res. 2016, 44(2).

[10] Steyvers M, Grifiths TL, "Finding scientific topics," PNAS 2004.

[11] Blei DM, Ng AY, Jordan MI, "Latent Dirichlet Allocation," Machine Learning 2003, 3, 993-1022.

[12] Steyvers M, "Probabilistic Topic Models. Latent Semantic Analysis: A Road to Meaning," 2007.

[13] La Rosa M, Fiannaca A, Rizzo R, Urso A, "Probabilistic topic modeling for the analysis and classification of genomic sequences," BMC Bioinformatics 2015, 16.

[14] Bisgin H, Liu Z, Fang H, Xu X, Tong W, "Mining FDA drug labels using an unsupervised learning technique - topic modeling," BMC Bioinformatics 2011, 12.

[15] ElShal S, Simm J, Arany A, Zakeri P, Davis J, Moreau Y, "A comprehensive comparison of two MEDLINE annotators for disease and gene linkage: sometimes less is more", LNBI: Bioinformatics and Biomedical Engineering 2016, 9656.

[16] Mitchell JA, Aronson AR, Mork JG, Folk LC, Humphrey SM, Ward JM, "Gene indexing: Characterization and analysis of NLM's GeneRIFs," AMIA Annual Symposium Proceedings 2003, 460–4

[17] Aronson,A.R., Lang,F.M., "An overview of MetaMap: historical perspective and recent advances," J Am Med Inform Assoc. 2010, 17(3), 229-36

[18] Pafilis E, et al., "EXTRACT: Interactive extraction of environment metadata and term suggestion for metagenomics sample annotation," Database 2016.

[19] Amberger J, Bocchini C, Hamosh A, "A new face and new challenges for Online Mendelian Inheritance in Man (OMIM®)," Hum Mutat 2011, 32, 564-7.

[20] Phan XH, Nguyen CT, "JGibbLDA," 2008.

[21] Steyvers M, "Matlab Topic Modeling Toolbox 1.4.," 2011.

[22] Jorge R. Herskovic, Len Y. Tanaka, William Hersh, and Elmer V. Bernstam, "A Day in the Life of PubMed: Analysis of a Typical Day's Query Log," J Am Med Inform Assoc. 2007, 14(2): 212–220.