Proceedings of the 34th Chinese Control Conference
July 28-30, 2015, Hangzhou, China

MeSHSim: An R/Bioconductor package for measuring semantic similarity over MeSH headings and MEDLINE documents

ZHOU Jing[1,2], SHUI Yuxuan[1,2], PENG Shengwen[1,2], LI Xuhui[3], MAMITSUKA Hiroshi[4], ZHU Shanfeng[1,2,*]

1. School of Computer Sciencee, Fudan University, Shanghai 200433, P. R. China
E-mail: zhusf@fudan.edu.cn, jingzhou12@fudan.edu.cn

2. Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai 200433, P. R. China
E-mail: zhusf@fudan.edu.cn, jingzhou12@fudan.edu.cn

3. School of Information Management, Wuhan University, Wuhan 430072, P. R. China
E-mail: lixuhui@whu.edu.cn

4. Bioinformatics Center, Institute for Chemical Research, Kyoto University, Kyoto 611-0011, Japan
E-mail: mami@kuicr.kyoto-u.ac.jp

Abstract: All recent MEDLINE documents are indexed by Medical Subject Headings (MeSH). Computing semantic similarity between two MeSH headings as well as two documents has become very important for many biomedical text mining applications. We develop an R package, MeSHSim, which can compute nine similarity measures between MeSH nodes, by which similarity between MeSH Headings as well as MEDLINE documents can be computed. In addition, MeSHSim supports querying hierarchy information of a MeSH heading and retrieving MeSH headings of a query document. It can be easily integrated into pipelines for any biomedical text analysis tasks. MeSHSim is released under GPL(General Public License), and available through Bioconductor and from Github at https://github.com/JingZhou2015/MeSHSim

Key Words: MeSH, Semantic Similarity, MEDLINE documents, R/Bioconductor Package

1 Introduction

MeSH (Medical Subject Headings) is a vocabulary thesaurus, being controlled by National Library of Medicine to index MEDLINE documents. MeSH consists of a set of description terms, which are organized in a hierarchical structure (called MeSH trees), where more general terms appear at nodes closer to the root and more specific terms appear at nodes closer to leaves [1]. Each MEDLINE document is manually annotated with a set of (usually 10-15) MeSH headings, including around three to five major headings, representing main topics of the corresponding document. Computing semantic similarities between two MeSH headings as well as two documents (one document having a set of MeSH headings) has been proved very useful to improve the performance of many biomedical text mining tasks, such as retrieval [2, 3], indexing [4] and clustering [5, 6]. As such, MeSH semantic similarity is widely used, but there have been no available tools for computing similarity between MeSH headings and also MEDLINE documents, except an online tool THE MESH SIMILARITY (http://sce.uhcl.edu/biomedsim/). This tool (last updated in 2011) cannot be used for computing similarity between two documents. Importantly, this is a web server, which cannot be a building block of a text mining software on MEDLINE documents. In this light, we provide an R package, MeSHSim, to compute semantic similarity among MeSH headings and also MEDLINE documents. MeSHSim can be easily integrated into biomedical text mining applications, which will be built by users.

* To whom correspondence should be addressed

This work has been partially supported by National Natural Science Foundation of China (61170097, 61272110), Scientific Research Starting Foundation for Returned Overseas Chinese Scholars, Ministry of Education, China and JSPS KAKENHI (#2430054).

2 Implementation

Many measures of semantic similarity for MeSH with a variety of interesting properties have been proposed. Generally we can divide these approaches into two types: path-based and information content (IC)-based measures. In our package we implement five path-based and four IC-based similarity measures.

2.1 Path-based Similarity Measure

This kind of measurement is based on spread activation theory proposed by [7], which assumes that the hierarchy of heading is organized along the lines of semantic similarity. The path-based measure computes the similarity as a function of the length of the path linking two headings.

2.1.1 SP: Shortest Path [8]

This measure is designed to find the gap between the local path length and the maximum path length, and use it as the semantic score.

$$Sim_{SP} = (MAX - L)/MAX \tag{1}$$

where MAX is the maximum path length between two headings in the hierarchy, L is the shortest path between two headings.

2.1.2 WL: Weighted Links [9]

It extended the Shortest Path measure by introducing the weighted edges in counting the path length.

$$Sim_{WL} = \frac{WMAX - WL}{WMAX} \tag{2}$$

where $WMAX = max_{i,j} WL_{ij}$ is the maximum weighted path length, and

$$WL_{ij} = \sum_{k \in path_{ij}} \frac{1}{H_k} \tag{3}$$

where H_k is the depth of node k in the hierarchy

2.1.3 WP: headingual similarity [10]

This measure is designed to find the nearest common ancestor of the two headings. The path length from this ancestor heading to the root of the ontology is scaled by the sum of path length of the two headings.

$$Sim_{WP} = \frac{2H_c}{H_1 + H_2} \tag{4}$$

where c is the nearest common ancestor of the two headings.

2.1.4 LC: Leacock and Chodorow [11]

This measure is to scale the shortest path by twice the maximum depth of the hierarchy.

$$Sim_{LC} = 1 - \frac{log(1+L)}{1+2D} \tag{5}$$

where D is the maximum depth of the heading.

2.1.5 Li: Li et al [12]

The measure combines the shortest path and the depth of the closest common ancestor in a non-linear function.

$$Sim_{Li} = e^{-\alpha L}\frac{e^{\beta H} - e^{-\beta H}}{e^{\beta H} + e^{-\beta H}} \tag{6}$$

where α and β are parameters scaling the contribution of shortest path length and depth respectively. H is the minimum depth of the nearest common ancestor

2.2 Information-Content based Similarity Measure

The IC-based measure uses a corpus, i.e. a collection of documents, with MeSH trees. That is, IC of MeSH heading c, i.e. $I(c)$, can be computed as $I(v) = -\log p(c)$, where $p(c) = \frac{freq(c)}{N}$, $freq(c)$ is the number of appearances of c in a given corpus and N is the number of documents in a given corpus.

2.2.1 Lord: Lord [13]

The first way to compare two headings is by using a measure that simply uses the probability of nearest common ancestor.

$$Sim_{lord} = 1 - p(c) \tag{7}$$

where c is the nearest common ancestor of heading c_1 and c_2.

2.2.2 Resnik: Resnik [14]

This measure signifies that the more information two headings share in common, the more similar they are.

$$Sim_{Resnik} = I(c) \tag{8}$$

2.2.3 Lin: Lin [15]

This measure is the same as WP, except that the information content is used, instead of node depth.

$$Sim_{Lin} = \frac{2 * I(c)}{I(c_1) + I(c_2)} \tag{9}$$

2.2.4 JC: Jiang and Conrath [16]

The measure defined a distance function as follows,

$$Dist_{JC} = I(c_1) + I(c) - 2 * I(c) \tag{10}$$

We use an exponential function to transform the distance into a similarity with constant λ. A large λ will yield a high similarity value even for weakly related headings.

$$Sim_{JC} = e^{-\frac{Dist_{JC}(c_1,c_2)}{\lambda}} \tag{11}$$

2.3 Semantic Similarity between MeSH Headings

As not all of the MeSH headings are represented by only one tree node, two frameworks have been proposed to compute the semantic similarity between two MeSH headings: node-based framework first proposed by [5] and heading-based framework.

2.3.1 Node-based Framework

Node-based framework uses the Average Maximum Match(AMM) method proposed by [17]. Considering a general case in which each MeSH main heading has one or multiple tree nodes, for each MeSH nodes v in main heading M, the maximum similarity between v and any MeSH nodes in M' is used to represent its contribution to the similarity between M and M':

$$\text{Sim}(M, M') = \frac{\sum_{v\in M}\max_{v'\in M'}\text{Sim}(v,v') + \sum_{v'\in M'}\max_{v\in M}\text{Sim}(v,v')}{|M| + |M'|}, \tag{12}$$

where $|M|$ indicates the number of MeSH node in M.

2.3.2 Heading-based Framework

Heading-based framework treat each MeSH main heading as a basic computational element, however many headings could be mapped to not a single position on the tree structure; so when projected to the tree structure, there might be several position-position relationship for a MeSH heading pair and we can calculate several candidate similarity scores. Typically, the heading-based similarity is computed by a simpler idea given as follows [4]:

$$\text{Sim}(M, M') = \max_{v\in M, v'\in M'}\text{Sim}(v, v'). \tag{13}$$

2.4 Similarity between Two Documents (MeSH Heading Sets)

As each MEDLINE article is marked by a set of MeSH headings, the similarity between two documents can be measured by the similarity between two MeSH heading sets, which relate to the two documents. Given two documents, D and D', the similarity between two MeSH headings can be calculated by again the AMM as follows:

$$\text{Sim}(D, D') = \frac{\sum_{M\in D}\max_{M'\in D'}\text{Sim}(M,M') + \sum_{M'\in D'}\max_{M\in D}\text{Sim}(M,M')}{|D| + |D'|}, \tag{14}$$

where $|D|$ indicates the number of headings in document D.

2.5 Required packages

MeSHSim needs three R packages: bitops, XML and RCurl, where bitops, used by RCurl, supports bitwise operations of integer vectors, XML supports reading XML documents and RCurl [18] is to fetch document information from PubMed. They are freely available at CRAN (Comprehensive R Archive Network).

Table 1: Nine functions in MeSHSim

name	input	output
nodeSim	two MeSH nodes	similarity
headingSim	two MeSH headings	similarity
headingSetSim	two MeSH heading sets	similarity
docSim	two MEDLINE documents	similarity
mnodeSim	multiple MeSH nodes	similarities
mheadingSim	multiple MeSH headings	similarities
nodeInfo	MeSH node	tree information
termInfo	MeSH heading	tree information
docInfo	MEDLINE document	document information

3 Functions and examples

Table 1 shows nine functions implemented in MeSHSim. The first four functions compute pairwise similarities, which takes a value between zero and one, higher values being more similar. For example, *nodeSim* (the default parameter of "method" is SP, standing for Shortest Path), *headingSim* and *headingSetSim* (the default parameter of "frame" is "node", standing for node-based framework) are executed as follows:

```
> nodeSim("C01.252.400","C01.539.757",method="SP")
[1] 0.8

> headingSim("Hip","Hand",method="WL",frame="node")
[1] 0.763113


> sa<-c("Body Regions", "Abdominal Cavity")
> sb<-c("Lumbosacral Region", "Body Regions")
> headingSetSim(sa, sb, method="JC", frame="node")
[1] 0.666128
```

docSim shows the similarity between two documents (PMID).

```
> docSim("2189633", "18974831",frame="heading")
[1] 0.1
```

The next two functions compute the similarities of all pairs of multiple inputs at once. Examples are as follows:

```
> la<-c("B03.440.450.425.800.200",
    "B01.650.940.800.575")
> lb<-c("B03.440.400.425.340",
    "B03.440.400.425.117.800",
    "B03.440.400.425.127.100")
> mnodeSim(la,lb,method="Lord")
          [,1]      [,2]      [,3]
[1,] 0.9962991 0.9962991 0.9962991
[2,] 0.8354435 0.8354435 0.8354435


> la<-c("Lumbosacral Region", "Body Regions")
> lb<-c("Body Regions","Abdomen")
> mheadingSim(la,lb,method="Resnik", frame="node")
          [,1]      [,2]
[1,] 0.2967087 0.3772228
[2,] 0.2967087 0.2967087
```

The rest three functions show information on queries. *nodeInfo* and *termInfo* are to query MeSH tree information of a given MeSH node and MeSH heading, respectively. The default setting of "brief" is "TRUE", which is to retrieve the whole MeSH tree information including the path to root node and all the child nodes of the given MeSH node or heading. *docInfo* outputs the title, abstract and MeSH headings of a query document, while the default setting of "verbose" is "TRUE" that is to output all MeSH heading without the title and abstract. Also "major" can be set at "TRUE" to output major MeSH headings only.

```
> nodeInfo("B03.440", brief=TRUE)
$B03
[1] "Bacteria"
$B03.440
[1] "Gram-Negative Bacteria"

> termInfo("Americas", brief=TRUE)
[[1]]
[[1]]$Z01
[1] "Geographic Locations"
[[1]]$Z01.107
[1] "Americas"

> docInfo("111123",verbose=TRUE,major=TRUE)
[1] "Title: Antibiotic accountability."
[1] "Abstract: NA"
[1] "MeSH Headings:"
[1] "Anti-Bacterial Agents" "Drug Utilization"
```

4 Application on Biomedical Document Clustering

As each MEDLINE document corresponds to an MeSH heading set, it has been demonstrated that MeSH plays a critical role in MEDLINE document clustering as one of the most informative field [19, 20]. Here, by integrating MeSH semantic similarity into MEDLINE documents clustering, we demonstrated a typical application of the MeSHSim. We choose two popular methods to incorporate MeSH semantic similarity. One of them is proposed by [5] using linear combination (LCM), the other one, called Semi-supervised Normalized Cut (SSNCut)[6].

In the LCM, we first normalize content similarity matrix S^{con} of MEDLINE documents as S^{con}_{nor}, and semantic similarity matrix S^{sem} of MeSH headings of articles as S^{sem}_{nor}, which calculated by function *headingSetSim* or *docSim* in the MeSHSim package. Besides, the contents (title and abstract) of documents can be retrieved by function *docInfo*. Then, we combine the similarity matrixes linearly by using weight as follows,

$$S_{LCM} = (1-\omega)S^{con}_{nor} + \omega S^{sem}_{nor} \tag{15}$$

where S_{LCM} is the integrated similarity matrix over which we cluster documents using Normalized Cut (NCut).

In the SSNCut, it has been demonstrated that MeSH semantic similarity can improve performance of MEDLINE documents clustering using semi-supervised learning, which takes advantage of a small amount of prior knowledge to guide clustering process and boost clustering performance.

SSNCut consists of the following four steps.

(1) Using function *docSim* or *headingSetSim* to calculate the MeSH semantic similarity $Sim^{sem}(d_i, d_j)$ for all pairs of documents (i.e. $d_i, d_j \in D$) in data set.

(2) Using cut-off trick on Sim^{sem} to generate prior constraints for semi-supervised clustering algorithm. For simplicity, we denote ML as must-link set and CL as cannot-link set.
(3) Calculate content similarity $Sim^{con}(d_i, d_j)$ for all pairs of documents (i.e. $d_i, d_j \in D$,) in document data set.
(4) Perform semi-supervised clustering using SSCNut over S^{con}, ML and CL.

4.1 Data set

In order to demonstrate the usefulness of the MeSHSim, we collected and generated a MEDLINE document data set TREC2005 extracted from TREC genomics track 2005. The organisation provides 50 topics to query relevant documents from the TREC Genomics 2005 corpus containing 4,591,008 documents [21]. In our experiments, we regard these 50 topics as true clusters of relevant documents. Then, we remove the topics having only nine or fewer documents to avoid very small clusters, and further remove documents that are relevant to more than one topic. We then obtain a basic data set of 2,317 documents in 24 topics.

4.2 Evaluation criteria

To evaluate clustering performance, we choose the Normalized Mutual Information (NMI) which is a popular and well-accepted criteria in clustering domain. Since there are multiple version of NMI, we use the squared version proposed by [22] defined as follows,

$$NMI(P;C) = \frac{I(P;C)}{\sqrt{H(P) * H(C)}} \tag{16}$$

where P and C are the predicted labels and true labels, respectively. $I(P;C) = H(P) - H(P|C)$ is the mutual information between P and C. $H(.)$ stands for entropy.

4.3 Results

Table 2 reports the clustering results using the LCM. The first column shows result using NCut over S^{con}_{nor} without MeSH similarity. The other four columns presents results based on S_{LCM}, and the numbers in the titles imply the weights ω of MeSH semantic similarity S^{sem}_{nor}.

It is obvious that the LCM outperforms the NCut using only content similarity. For example, the NMI of LCM is improved to 0.8210 at least, and 0.8518 at most, while the NMI of NCut is only 0.8022. It is demonstrated that MeSHSim package is able to help analyze MELDINE documents.

Table 2: LCM results of TREC2005.

NCut	$LCM_{0.3}$	$LCM_{0.4}$	$LCM_{0.5}$	$LCM_{0.6}$
0.8022	0.8302	**0.8518**	0.8286	0.8210

Table 3: SSNCut results of TREC2005.

NCut	$SSNCut_{0.5\%}$	$SSNCut_{1\%}$	$SSNCut_{2\%}$	$SSNCut_{5\%}$
0.8022	0.8719	**0.8770**	0.8756	0.8671

Table 3 shows the clustering results using the SSNCut. The last four columns list results combining MeSH similarity. The percentage numbers are the ratio of generated constraints using cut-off trick. For example, $SSNCut_{1\%}$ means the pairs of documents, whose MeSH similarities are within top 1%, are connected with must-links, while those within bottom 1% are used to generate cannot-links. Note that we use the "JC" method and "node" based framework in the function *headingSetSim* to calculate MeSH-based similarities over MEDLINE articles.

As shown in the Table 3, the SSNCut outperforms NCut at all the four percentages of constraints. For instance, the NMI of SSNCut is boosted to 0.8671 at least, and 0.8770 at most, while the NMI of NCut is only 0.8022. Thus, we can see that our MeSHSim package is able to provide convenience for researchers to do MEDLINE documents clustering.

5 Conclusion

The measures of the semantic similarities for MeSH ontology facilitate users to compare MEDLINE documents, and therefore have became an significant prior knowledge in many text mining approaches in biomedical domain. The MeSHSim package implemented nine typical MeSH ontology-based semantic similarity measures in the powerful R system. Compared with the few existing related tools, like the online server THE MESH SIMILARITY, the MeSHSim can be easily integrated into pipelines for other biomedical text analysis task to improve their performance, such as information retrieval, biomedical document clustering and citation searching process. Other utilities, such as functions for querying MeSH heading information and retrieving MEDLINE documents, should offer a straightforward way to study MeSH tree and MEDLINE documents.

References

[1] S. Nelson, M. Schopen, A. Savage, J. Schulman, and N. Arluk, "The mesh translation maintenance system: structure, interface design, and implementation." *Studies in health technology and informatics*, vol. 107, no. Pt 1, pp. 67–69, 2004.

[2] R. Rada, H. Mili, E. Bichnell, and M. Blettner, "Development and application of a metric on semantic nets." *IEEE Trans. Syst., Man, Cybern*, vol. 9, pp. 17–30, 1989.

[3] S. Blott, C. Gurrin, G. Jones, A. Smeaton, and T. Sodring, "On the use of mesh headings to improve retrieval effectiveness." *Text REtrieval Conference*, pp. 215–224, 2003.

[4] A. Névéol, K. Zeng, and O. Bodenreider, "Besides precision & recall: exploring alternative approaches to evaluating an automatic indexing tool for MEDLINE," *AMIA Annu Symp Proc.*, pp. 589–593, 2006.

[5] S. Zhu, J. Zeng, and H. Mamitsuka, "Enhancing MEDLINE document clustering by incorporating MeSH semantic similarity." *Bioinformatics*, vol. 25, no. 15, pp. 1944–1951, 2009.

[6] J. Gu, W. Feng, J. Zeng, H. Mamitsuka, and S. Zhu, "Efficient semisupervised medline document clustering with mesh-semantic and global-content constraints," *IEEE Trans. Cybernetics*, pp. 1265–1276, 2013.

[7] P. R. Cohen and R. Kjeldsen, "Information retrieval by constrained spreading activation in semantic networks," *Information processing & management*, vol. 23, no. 4, pp. 255–268, 1987.

[8] H. Bulskov, R. Knappe, and T. Andreasen, "On measuring similarity for conceptual querying." *Proceedings of the 5th International Conference on Flexible Query Answering Systems (FQAS)*, vol. 2522, pp. 100–111, 2002.

[9] R. Richardson, A. Smeaton, and J. Murphy, "Using wordnet as a knowledge base for measuring semantic similarity between words," 1994.

http://ir.ohsu.edu/genomics

[10] Z. Wu and M. Palmer, "Verbs semantics and lexical selection." *In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics (ACL'94)*, pp. 133–138, 1994.

[11] C. Leacock and M. Chodorow, "Filling in a sparse training space forword sense identification." *In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics(ACL94)*, 1994.

[12] Y. Li, Z. A. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources." *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 871–882, 2003.

[13] P. Lord, R. Stevens, A. Brass, and C. Goble, "Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation." *Bioinformatics*, vol. 19, no. 10, pp. 1275–1283, 2003.

[14] O. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity and natural language." *Journal of Artificial Intelligence Research*, vol. 19, pp. 95–1130, 1999.

[15] D. Lin, "Principle-based parsing without overgeneration," in *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, ser. ACL '93. Stroudsburg, PA, USA: Association for Computational Linguistics, 1993, pp. 112–120. [Online]. Available: http://dx.doi.org/10.3115/981574.981590

[16] J. Jiang and D. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy." *In Proceedings of the International Conference on Research in Computational Linguistic, Taiwan*, 1998.

[17] J. Z. Wang, Z. Du, R. Payattakool, P. S. Yu, and C.-F. Chen, "A new method to measure the semantic similarity of GO terms," *Bioinformatics*, vol. 23, no. 10, pp. 1274–1281, 2007.

[18] D. Lang, "R as a web client-the rcurl package." *Journal of Statistical Software*, 2007.

[19] S. Zhu, I. Takigawa, J. Zeng, and H. Mamitsuka, "Field independent probabilistic model for clustering multi-field documents," *Information Processing & Management*, vol. 45, no. 5, pp. 555–570, 2009.

[20] X. Huang, X. Zheng, W. Yuan, F. Wang, and S. Zhu, "Enhanced clustering of biomedical documents using ensemble non-negative matrix factorization," *Information Sciences*, vol. 181, no. 11, pp. 2293–2302, 2011.

[21] W. Hersh, A. Cohen, J. Yang, R. T. Bhupatiraju, P. Roberts, and M. Hearst, "Trec 2005 genomics track overview," in *In TREC 2005 notebook*, 2005, pp. 14–25.

[22] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.