The MeSHSim package

Jing Zhou, Yuxuan Shui
School of Computer Science, Fudan University, Shanghai, China

April 16, 2015

1 INTRODUCTION

MeSH(Medical Subject Headings) is a vocabulary thesaurus, being controlled by NLM(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(Nelson et al., 2004). Each MeSH node is represented by a tree number, which indicates the postion of the node in MeSH tree. MeSH headings are term names or identifiers. 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 corresponding document. Besides, MeSH is also used for indexing the NLM-produced database including cataloging of books and audiovisuals. 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 (Rada et al., 1989; Blott et al., 2003), indexing (Névéol et al., 2006) and clustering (Zhu et al., 2009; Gu et al., 2013).

MeSHSim is an R package for semantic similarities calculation among MeSH headings and MEDLINE documents. As shown in table 1 Five path-based measures and four Information Content (IC)- based measures are implemented in MeSHSim. It also supports querying the hierarchy information of a MeSH heading and information of a given document including title, abstraction and MeSH headings. The package can be easily integrated into pipelines for other biomedical text analysis tasks. With its specific focus, to the best of our knowledge, MeSHSim is the most comprehensive software package of this kind.

Function	**Description**
`nodeSim`	Return semantic similarity between two MeSH nodes.
`mnodeSim`	Return semantic similarity matrix between two lists of MeSH nodes.
`headingSim`	Return semantic similarity between two MeSH headings.
`mheadingSim`	Return semantic similarity matrix between two lists of MeSH headings.
`headingSetSim`	Return semantic similarity between two sets of MeSH headings.
`docSim`	Return semantic similarity between two MEDLINE documents.
`nodeInfo`	Return hierarchy information of a given MeSH node.
`termInfo`	Return hierarchy information of a given MeSH heading
`docInfo`	Return information of a given MEDLINE document including title, abstract, MeSH headings.

Table 1: **Summary of functions implemented by MeSHSim.**

2 PATH-BASED SIMILARITY MEASURE

The kind of measurement is based on spread activation theory(Cohen and Kjeldsen, 1987), which assume that the hierarchy of heading is organized along the lines of semantic similarity. As all the headings of the ontology are organized in a hierarchy, where more general headings are near the root of the hierarchy, and more specific ones near at the leaves, it is convenient to measure similarity as a function of the length of the path linking the headings and on the position of the headings in the hierarchy. Most of the measures that are based on the hierarchy structure of ontology are actually based on: 1) path length (i.e., shortest path length/distance between the two heading nodes) and 2) depth of heading

nodes in the ontology.

2.1 *SP*: *Short Path*(Bulskov et al., 2002)

The measurement is motivated by the observation that the headingual distance between two nodes is often proportional to the number of edges separating the two nodes in the hierarchy. 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.

$$Sim_{SP} = \frac{MAX - L}{MAX} \tag{1}$$

where MAX is the maximum path length between two headings in the hierarchy, L is the shortest path between two headings. A measure like this could be integrated into information retrieval system which is based on indexing documents and queries into terms from a semantic hierarchy, or implemented to help rank documents for search engine(Hliaoutakis, 2005).

2.2 *WL*: *Weighted Links*(Richardson et al., 1994)

It extended the Shortest Path measure by introducing the weighted edges in counting the path length instead of the uniform edges, the distance between two headings is translated into sum of the weights of the traversed links instead of counting them.

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

where $WMAX = max_{i,j} WL_{ij}$ is the maximum weighted path length between two headings in the hierarchy, 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

2.3 *WP*: *Headingualr Similarity*(Wu and Palmer, 1994)

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 H_1 and H_2 are the depths of two headings, respectively, H_c is the depth of the nearest common ancestor of the two headings.

2.4 *LC*: *Leacock and Chodorow*(Leacock and Chodorow, 1994)

This measure is based on finding the shortest path between two headings and scaling that value by twice the maximum depth of the hierarchy, and then taking the logarithm of the resulting score.

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

where L is the shortest path between two headings, and D is the maximum depth of the heading in the ontology.

2.5 *Li*: *Li et al.*(Li et al., 2003a)

The measure, which is intuitively and empirically derived, combines the shortest path and the depth of the closest common ancestor in a non-linear function.

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

where $\alpha \geq 0$ and $\beta \geq 0$ are parameters scaling the contribution of shortest path length and depth respectively. According to (Li et al., 2003b), we set α and β to 0.2 and 0.6 respectively. The value is 1(for similar headings) and 0, L is the shortest path between two headings, H is the minimum depth of the least common nearest common ancestor. This measure is motived by the fact that information sources are infinite to some exend while humans compare word similarity with a finite interval between completely similar and nothing similar. Intuitively the transformation between an infinite interval to a finite one is non-linear(Hliaoutakis, 2005).

3 INFORMATION-BASED SIMILARITY MEASURE

The notion of information content of the heading practically has to do with the frequency of the heading in a given document collection. Given a corpus C, $p(c)$ is the probability of encountering an instance of heading c. The heading probability is defined as $p(c) = freq(c)/N$, where N is the total number of headings that appear in C, $freq(c)$ corresponds to the frequency of heading c. Additionally, the frequency counts of every heading includes the frequency counts of subsumed headings in an IS-A hierarchy. It implies that if c_1 is a sub-heading of c_2 in the MeSH tree, then $p(c_1) \leq p(c_2)$, which intuitively means that the more general the concept c is, the higher its associated probability. Then, the information content of c can be computed: $I(c) = -logp(c)$, which means that as probability increases, informationtiveness decreases, os the more abstract a MeSH heading, the lower its information content.

3.1 *Lord*: *Lord et al.*(Lord et al., 2003)

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. The measure implies that similarity judgements might be sensitive to frequency of a heading rather than its information content.

3.2 *Resnik*: *Resnik*(Resnik, 1999)

This measure signifies that the more information two headings share in common, the more similar they are, and the information shared by two headings is indicated by the information content of the heading that subsume them in the ontology.

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

where c is the nearest common ancestor of heading c_1 and c_2. As $p(c)$ varies between 0 and 1, this measure varies infinitiy(for very similar terms) to 0. In order to keep consistency of the range of simialrity, we normalized the Resnik measure by divided it by the max information content in the hierarchy.

3.3 *Lin*: *Lin*(Lin, 1993)

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

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

where c is the nearest common ancestor of heading c_1 and c_2. As the Resnik measure relies only on information content of the nearest common ancestor, there are only as many discrete socre as there are ontology terms. Lin's measure utilize information content of both compared terms and their nearest common ancestor, which means the amount of discrete scores is quadratic in the number of terms in the ontology. Thus, this measure can obtain a bettern ranking of similarity that Resnik' measure.

3.4 *JC*: *Jiang and Conrath*(Jiang and Conrath, 1998)

The measure defined a distance function as follows,

$$Dist_{JC} = I(c_1) + I(c_2) - 2 * I(c) \tag{10}$$

The authors use an exponential function to transform the distance into a similarity with constant λ, which adjusts the steepness of the exponential curve. A large λ will yield a high similarity value even for weakly related headings.

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

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

4 SEMANTIC SIMILARITY BETWEEN MESH HEADINGS

Althourgh the structure of MeSH is a hierarchical tree, a MeSH heading can appear in different subtrees at the same time. There are 15 tree hierarchies(subtree) in the MeSH ontology. 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 and heading-based framework. On the one hand, node-based framework implies that we project the two MeSH main headings onto the tree structure and calculate the similarity between the two projected node sets(Zhu et al., 2009). On the other hand, heading-based framework tries to build a relation structure among main headings through the MeSH tree structure and then only consider the nearest path between them. Please note that the two frameworks differ from each other in computing the information content.

4.1 Node-based Framework

Node-based framework uses the Average Maximum Match method(Wang et al., 2007). 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':

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

Computing IC: As each MeSH main heading corresponds to one or several MeSH tree nodes, accordingly the frequency of these related tree nodes and their ancestor nodes should be updated simultaneously. Then, the total number N is defined as the frequency of global root node. Finally, the information content of each node is computed as previous mentioned in section 2.

4.2 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 maximum similarity score is chosen as the semantic score between the two headings.

Computing IC: As each MeSH main heading could be mapped to the tree structure, the frequency of it and it's ancestor main headings (through it's projection onto the tree structure) should accumulate simultaneously. And the total number N is denoted as the sum of the frequencies of all the main headings. Therefore, the information content of each main heading is computed as previous mentioned in section 2.

5 SIMILARITY BETWEEN TWO DOCUMENTS (MESH 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 sets, which relate to the two documents. Sematic similarity between two MeSH sets are calculated by the idea of Average Maximum Match(AMM)(Zhu et al., 2009).

$$Sim(S, S') = \frac{\sum_{M \in S} max_{M' \in S'} Sim(M, M') + \sum_{M' \in S'} max_{M \in S} Sim(M, M')}{|S| + |S'|} \tag{13}$$

where S and S' are the two MeSH sets corresponding to two MEDLINE documents, and M and M' are MeSH heading that belong to related MeSH sets.

6 Usage of MeSHSim

6.1 Similarity Measurements Statements

SP: Shortest Path method
WL: Weighted Link method
WP: Wu and Palmer's method
LC: Leacock and Chodorow's method
Li: Li's method
Lord: Lord's method

Resnik: Resnik's method
Lin: Lin's method
JC: Jiang and Conrath's method

Detailed information are listed previous sections.

6.2 MeSH Nodes Similarities

In this section we illustrate how to caluculate semantic similarity between MeSH nodes using MeSHSim package through the use of the `nodeSim`, `mnodeSim`. Function `nodeSim` is to calculate similarity between two MeSH nodes, whose value is between 0 and 1, function `mnodeSim` is to calculate similarity matrix between two lists of MeSH nodes.

6.2.1 `nodeSim`

Usage of nodeSim
nodeSim(node1, node2, method="SP", frame="node", env=NULL)
Function: to calculate similarity between two MeSH nodes.
Parameters:
node1,node2: two MeSH nodes
method: similarity measurement, options are presented at Section Similarity Measurements Statements
frame: framework for calculating the similarity. as MeSH node similarity can be only calculated by node-based, so the default value is set to "node".
env: the dataset to use. As the dataset has been pre-calculated in MeSHSim package, there is no need to set "env" defaultly.
Value:
return similarity between two MeSH nodes
Examples of nodeSim
Let us examine the similarity of the MeSH nodes for nodes "B03.440.400.425.340" and "B03.440.400.425.117.800", which stand for MeSH heading "Francisella" and "Taylorella", respectively.

```
> library(RCurl)
> library(XML)
> library(MeSHSim)
> nodeSim("B03.440.400.425.340", "B03.440.400.425.117.800")

[1] 0.8571429
```

The type of similarity measurement is set by specifying the *method* argument one to of `"SP"`, `"WL"`, `"WP"`, `"LC"`, `"Li"`, `"Lord"`, `"Resnik"`, `"Lin"`, and `"JC"`. Detailed information are listed in Section Similarity Measurements Statements. The default type is `"SP"`, which is Shortest Path.

```
> nodeSim("B03.440.400.425.340", "B03.440.400.425.117.800", method="LC")

[1] 0.5693234

> nodeSim("B03.440.400.425.340", "B03.440.400.425.117.800", method="Lin")

[1] 0.5398654
```

6.2.2 `mnodeSim`

Usage of mnodeSim
mnodeSim(nodeList1, nodeList2, method="SP", frame="node", env=NULL)
Function: to calculate similarity matrix between two lists of MeSH nodes.
Parameters:
nodeList1,nodeList2: two lists of MeSH nodes
method: similarity measurement, options are presented at Section Similarity Measurements Statements
frame: framework for calculating the similarity. as MeSH node similarity can be only calculated by node-based, so the default value is set to "node"
env: the dataset to use. As the dataset has been pre-calculated in MeSHSim package, there is no need to set "env" defaultly.

Value:

return similarity matrix between two lists of MeSH nodes

Examples of mnodeSim

Let us examine the similarity matrix of two MeSH node lists, which are ["B03.440.450.425.800.200", "B03.440.450.900.859.225"] and ["B03.440.400.425.340", "B03.440.400.425.117.800", "B03.440.400.425.127.100"], respectively.

```
> nodeList1<-c("B03.440.450.425.800.200", "B03.440.450.900.859.225")
> nodeList2<-c("B03.440.400.425.340", "B03.440.400.425.117.800", "B03.440.400.425.127.100")
> mnodeSim(nodeList1,nodeList2)

          [,1]      [,2]      [,3]
[1,] 0.6666667 0.6190476 0.6190476
[2,] 0.6666667 0.6190476 0.6190476
```

The type of similarity measurement is set by specifying the *method* argument to one of `"SP"`, `"WL"`, `"WP"`, `"LC"`, `"Li"`, `"Lord"`, `"Resnik"`, `"Lin"`, and `"JC"`. Detailed information are listed in Section Similarity Measurements Statements. The default type is `"SP"`, which is Shortest Path.

```
> nodeList1<-c("B03.440.450.425.800.200", "B03.440.450.900.859.225")
> nodeList2<-c("B03.440.400.425.340", "B03.440.400.425.117.800", "B03.440.400.425.127.100")
> mnodeSim(nodeList1,nodeList2,method="Lord")

          [,1]      [,2]      [,3]
[1,] 0.9962991 0.9962991 0.9962991
[2,] 0.9962991 0.9962991 0.9962991

> mnodeSim(nodeList1,nodeList2,method="Resnik")

          [,1]      [,2]      [,3]
[1,] 0.2921566 0.2921566 0.2921566
[2,] 0.2921566 0.2921566 0.2921566
```

6.3 MeSH Headings Similarities

In this section we illustrate how to caluculate semantic similarity between MeSH headingsusing MeSHSim package through the use of the `headingSim`, `mheadingSim` and `headingSetSim`. Function `headingSim` is to calculate similarity between two MeSH headings, whose value is between 0 and 1, function `mheadingSim` is to calculate similarity matrix between two lists of MeSH headings, and function `headingSetSim` is to calculate similarity between two sets of MeSH headings.

6.3.1 `headingSim`

Usage of headingSim

headingSim(heading1, heading2, method="SP", frame="node", env=NULL)

Function: to calculate similarity between two MeSH headings.

Parameters:

heading1,heading2: two MeSH headings

method: similarity measurement, options are presented at Section Similarity Measurements Statements

frame: framework for calculating the similarity. One of "node" and "heading"

env: the dataset to use. As the dataset has been pre-calculated in MeSHSim package, there is no need to set "env" defaultly.

Value:

return similarity between two MeSH headings

Examples of headingSim

Let us examine the similarity of the MeSH headings for headings "Lumbosacral Region" and "Body Regions".

```
> headingSim("Lumbosacral Region", "Body Regions")

[1] 0.8636364
```

The type of similarity measurement is set by specifying the *method* argument to one of "SP", "WL", "WP", "LC", "Li", "Lord", "Resnik", "Lin", and "JC". Detailed information are listed in Section Similarity Measurements Statements. The default type is "SP", which is Shortest Path.

```
> headingSim("Lumbosacral Region", "Body Regions", method="LC")

[1] 0.5693234

> headingSim("Lumbosacral Region", "Body Regions", method="Lin")

[1] 0.7236669
```

The type of framework is set by specifying the *frame* argument to of "node" and "heading", which stand for "node-based" and "heading-based" similarity framework, repectively, described in prevous sections. The default setting is "node" using node-based framework.

```
> headingSim("Lumbosacral Region", "Body Regions", method="JC", frame="node")

[1] 0.2351395

> headingSim("Lumbosacral Region", "Body Regions", method="JC", frame="heading")

[1] 0.2351395
```

6.3.2 mheadingSim

Usage of mheadingSim
mheadingSim(headingList1, headingList2, method="SP", frame="node", env=NULL)
Function: to calculate similarity matrix between two lists of MeSH headings.
Parameters:
headingList1,headingList2: two lists of MeSH headings
method: similarity measurement, options are presented at Section Similarity Measurements Statements
frame: framework for calculating the similarity. One of "node" and "heading"
env: the dataset to use. As the dataset has been pre-calculated in MeSHSim package, there is no need to set "env" defaultly.
Value:
return similarity matrix between two lists of MeSH headings

Examples of mheadingSim
Let us examine the similarity matrix of two MeSH heading lists, which are ["Body Regions", "Abdomen", "Abdominal Cavity"] and ["Lumbosacral Region", "Body Regions"], respectively.

```
> headingList1<-c("Body Regions", "Abdomen", "Abdominal Cavity")
> headingList2<-c("Lumbosacral Region", "Body Regions")
> mheadingSim(headingList1, headingList2)

          [,1]      [,2]
[1,] 0.8636364 1.0000000
[2,] 0.8636364 0.9090909
[3,] 0.8181818 0.8636364
```

The type of similarity measurement is set by specifying the *method* argument to one of "SP", "WL", "WP", "LC", "Li", "Lord", "Resnik", "Lin", and "JC". Detailed information are listed in Section Similarity Measurements Statements. The default type is "SP", which is Shortest Path.

```
> headingList1<-c("Body Regions", "Abdomen", "Abdominal Cavity")
> headingList2<-c("Lumbosacral Region", "Body Regions")
> mheadingSim(headingList1, headingList2, method="Lord")

          [,1]      [,2]
[1,] 0.9966083 0.9966083
[2,] 0.9992751 0.9966083
[3,] 0.9992751 0.9966083
```

```
> mheadingSim(headingList1, headingList2, method="Resnik")

          [,1]      [,2]
[1,] 0.2967087 0.2967087
[2,] 0.3772228 0.2967087
[3,] 0.3772228 0.2967087
```

The type of framework is set by specifying the *frame* argument to of `"node"` and `"heading"`, which stand for "node-based" and "heading-based" similarity framework, repectively, described in prevous sections. The default setting is "node" using node-based framework.

```
> headingList1<-c("Body Regions", "Abdomen", "Abdominal Cavity")
> headingList2<-c("Lumbosacral Region", "Body Regions")
> mheadingSim(headingList1, headingList2, method="JC", frame="node")

          [,1]      [,2]
[1,] 0.2351395 1.0000000
[2,] 0.3277106 0.4982023
[3,] 0.2636690 0.4008431

> mheadingSim(headingList1, headingList2, method="JC", frame="heading")

          [,1]      [,2]
[1,] 0.2351395 1.0000000
[2,] 0.3277106 0.4982023
[3,] 0.2636690 0.4008431
```

6.3.3 `headingSetSim`

Usage of headingSetSim
headingSetSim(headingSet1, headingSet2, method"SP", frame="node", env=NULL)
Function: to calculate similarity matrix between two sets of MeSH headings.
Parameters:

headingSet1,headingSet2: two sets of MeSH headings
method: similarity measurement, options are presented at Section Similarity Measurements Statements
frame: framework for calculating the similarity. One of "node" and "heading"
env: the dataset to use. As the dataset has been pre-calculated in MeSHSim package, there is no need to set "env" defaultly.

Value:

return similarity matrix between two sets of MeSH headings

Examples of headingSetSim
Let us examine the similarity of two MeSH heading sets using the method introduced in section 5, which are ["Body Regions", "Abdomen", "Abdominal Cavity"] and ["Lumbosacral Region", "Body Regions"], respectively.

```
> headingSet1<-c("Body Regions", "Abdomen", "Abdominal Cavity")
> headingSet2<-c("Lumbosacral Region", "Body Regions")
> headingSetSim(headingSet1, headingSet2)

[1] 0.9272727
```

The type of similarity measurement is set by specifying the *method* argument to one of `"SP"`, `"WL"`, `"WP"`, `"LC"`, `"Li"`, `"Lord"`, `"Resnik"`, `"Lin"`, and `"JC"`. Detailed information are listed in Section Similarity Measurements Statements. The default type is `"SP"`, which is Shortest Path.

```
> headingSet1<-c("Body Regions", "Abdomen", "Abdominal Cavity")
> headingSet2<-c("Lumbosacral Region", "Body Regions")
> headingSetSim(headingSet1, headingSet2, method="Lord")

[1] 0.9982084

> headingSetSim(headingSet1, headingSet2, method="Resnik")
```

```
[1] 0.3450171
```

The type of framework is set by specifying the *frame* argument to of `"node"` and `"heading"`, which stand for "node-based" and "heading-based" similarity framework, repectively, described in prevous sections. The default setting is "node" using node-based framework.

```
> headingSet1<-c("Body Regions", "Abdomen", "Abdominal Cavity")
> headingSet2<-c("Lumbosacral Region", "Body Regions")
> headingSetSim(headingSet1, headingSet2, method="JC", frame="node")

[1] 0.6453512

> headingSetSim(headingSet1, headingSet2, method="JC", frame="heading")

[1] 0.6453512
```

6.4 MEDLINE Documents Similarities

In this section we illustrate how to caluculate semantic similarity between MEDLINE articles through the use of the `docSim`. Function `docSim` is to calculate similarity between two MEDLINE articles, whose value is between 0 and 1.

6.5 `docSim`

Usage of docSim
docSim(pmid1, pmid2, method="SP", frame="node", major=FALSE, env=NULL)
Function: to calculate similarity between two MEDLINE documents.
Parameters:
pmid1, pmid2: PMIDs(PubMed IDs) of two articles whose similarity is needed to be calculated
method: similarity measurement, options are presented at Section Similarity Measurements Statements
frame: framework for calculating the similarity. One of "node" and "heading"
major: use only major MeSH headings to calculate documents similarity
env: the dataset to use. As the dataset has been pre-calculated in MeSHSim package, there is no need to set "env" defaultly.
Value:
return similarity between two MEDLINE documents

Examples of docSim
Let us examine the similarity of two MEDLINE documents, whose PMID(PubMed ID) is "1111113"" and "1111111" representing document "Growth hormone: independent release of big and small forms from rat pituitary in vitro" and document "Evaporative water loss in box turtles: effects of rostral brainstem and other temperatures.", respectively.

```
> docSim("1111113","1111111")

[1] 0.4628407
```

The type of similarity measurement is set by specifying the *method* argument to one of `"SP"`, `"WL"`, `"WP"`, `"LC"`, `"Li"`, `"Lord"`, `"Resnik"`, `"Lin"`, and `"JC"`. Detailed information are listed in Section Similarity Measurements Statements. The default type is `"SP"`, which is Shortest Path.

```
> docSim("1111113","1111111", method="LC")

[1] 0.3123662

> docSim("1111113","1111111", method="Lin")

[1] 0.3165832
```

The type of framework is set by specifying the *frame* argument to of `"node"` and `"heading"`, which stand for "node-based" and "heading-based" similarity framework, repectively, described in prevous sections. The default setting is "node" using node-based framework.

```
> docSim("1111113","1111111", method="JC", frame="node")
```

```
[1] 0.1897405

> docSim("1111113","1111111", method="JC", frame="heading")

[1] 0.238017
```

Users are able to choose to use only major MeSH headings to caculate documents similarity by setting *major* argument to "TRUE". The default setting is "FALSE", which means use all the MeSH headings to calculate documents similarity.

```
> docSim("1111113","1111111", method="JC", frame="node", major=TRUE)

[1] 0.03399591

> docSim("1111113","1111111", method="JC", frame="node", major=FALSE)

[1] 0.1897405
```

6.6 MeSH and MEDLINE Information Retrieval

In this section, we illustrate how to retrieve MeSH node, MeSH heading and MEDLINE information using MeSHSim package through the use of `nodeInfo`, `termInfo` and `docInfo`. Function `nodeInfo` and `termInfo` are to retrieve hierarchical information of MeSH node and MeSH heading, and function `docInfo` is to retrieve MEDLINE article information including title, abstract and related MeSH headings.

6.6.1 `docInfo`

Usage of docInfo
docInfo(pmid, verbose=FALSE, major=FALSE)
Function: to retrieve information of a given article from PubMed
Parameters:
pmid: pmid of the desired article.
verbose: whether the title and abstract of the article should be print out.
major: whether only major MeSH headings should be returned.
Value:
return information of a given article from PubMed including title, abstract, MeSH headings

Examples of docInfo

```
> library(RCurl)
> library(XML)
> library(MeSHSim)
> docInfo("1111111")

[1] "Animals"                     "Body Temperature Regulation"
[3] "Brain Stem"                  "Hot Temperature"
[5] "Hypothalamus"                "Turtles"
[7] "Water Loss, Insensible"
```

Whether the title and abstract of the article should be print out could be set to *verbose*. If *verbose* is set to "TRUE", `docInfo` will fetch title and abstract of the given document. If *verbose* is set to "FALSE", it will only retrieve MeSH headings of the given document. The default setting is "FALSE".

```
> docInfo("1111111", verbose=TRUE)

[1] "Title: Evaporative water loss in box turtles: effects of rostral brainstem and other temperatures."
[1] "Abstract: Box turtles were implanted with thermodes astride the preoptic tissue of the brainstem. The ra
[1] "MeSH Headings:"
[1] "Animals"                     "Body Temperature Regulation"
[3] "Brain Stem"                  "Hot Temperature"
[5] "Hypothalamus"                "Turtles"
[7] "Water Loss, Insensible"
```

```
> docInfo("1111111", verbose=FALSE)

[1] "Animals"                       "Body Temperature Regulation"
[3] "Brain Stem"                    "Hot Temperature"
[5] "Hypothalamus"                  "Turtles"
[7] "Water Loss, Insensible"
```

Users are able to choose to fetch major MeSH headings or all the MeSH heading by setting *major*. The default setting is retrieve all MeSH headings, whose value "FALSE"

```
> docInfo("1111111", verbose=FALSE, major=TRUE)

[1] "Brain Stem"             "Hot Temperature"
[3] "Turtles"                "Water Loss, Insensible"

> docInfo("1111111", verbose=FALSE, major=FALSE)

[1] "Animals"                       "Body Temperature Regulation"
[3] "Brain Stem"                    "Hot Temperature"
[5] "Hypothalamus"                  "Turtles"
[7] "Water Loss, Insensible"
```

6.6.2 nodeInfo

Usage of nodeInfo
nodeInfo(node1, env=NULL)
Function: to retrieve hierarchy information of a given MeSH node
Parameters:
node: a MeSH node name
brief: whether to retrive breif tree information of MeSH node
env: the dataset to use. As the dataset has been pre-calculated in MeSHSim package, there is no need to set "env" defaultly.
Value:
return hierarchy information of a given MeSH node

Examples of nodeInfo
Users are able to choose to fetch brief information of a given MeSH node by setting *brief*. The default setting is retrieve all MeSH headings, whose value "TRUE"

```
> nodeInfo("B03.440.400.425.127.100")

$B03
[1] "Bacteria"

$B03.440
[1] "Gram-Negative Bacteria"

$B03.440.400
[1] "Gram-Negative Aerobic Bacteria"

$B03.440.400.425
[1] "Gram-Negative Aerobic Rods and Cocci"

$B03.440.400.425.127
[1] "Azorhizobium"

$B03.440.400.425.127.100
[1] "Azorhizobium caulinodans"

> nodeInfo("B03.440.400",brief=FALSE)
```

```
$B03
$B03$term
[1] "Bacteria"

$B03$B03.026
[1] "B03.026"

$B03$B03.054
[1] "B03.054"

$B03$B03.110
[1] "B03.110"

$B03$B03.120
[1] "B03.120"

$B03$B03.130
[1] "B03.130"

$B03$B03.140
[1] "B03.140"

$B03$B03.165
[1] "B03.165"

$B03$B03.250
[1] "B03.250"

$B03$B03.275
[1] "B03.275"

$B03$B03.280
[1] "B03.280"

$B03$B03.300
[1] "B03.300"

$B03$B03.335
[1] "B03.335"

$B03$B03.370
[1] "B03.370"

$B03$B03.440
$B03$B03.440$term
[1] "Gram-Negative Bacteria"

$B03$B03.440$B03.440.040
[1] "B03.440.040"

$B03$B03.440$B03.440.050
[1] "B03.440.050"

$B03$B03.440$B03.440.090
[1] "B03.440.090"

$B03$B03.440$B03.440.097
[1] "B03.440.097"
```

```
$B03$B03.440$B03.440.100
[1] "B03.440.100"

$B03$B03.440$B03.440.180
[1] "B03.440.180"

$B03$B03.440$B03.440.190
[1] "B03.440.190"

$B03$B03.440$B03.440.210
[1] "B03.440.210"

$B03$B03.440$B03.440.400
$B03$B03.440$B03.440.400$term
[1] "Gram-Negative Aerobic Bacteria"

$B03$B03.440$B03.440.400$B03.440.400.050
[1] "B03.440.400.050"

$B03$B03.440$B03.440.400$B03.440.400.280
[1] "B03.440.400.280"

$B03$B03.440$B03.440.400$B03.440.400.400
[1] "B03.440.400.400"

$B03$B03.440$B03.440.400$B03.440.400.425
[1] "B03.440.400.425"

$B03$B03.440$B03.440.400$B03.440.400.450
[1] "B03.440.400.450"

$B03$B03.440$B03.440.400$B03.440.400.645
[1] "B03.440.400.645"

$B03$B03.440$B03.440.400$B03.440.400.840
[1] "B03.440.400.840"


$B03$B03.440$B03.440.425
[1] "B03.440.425"

$B03$B03.440$B03.440.450
[1] "B03.440.450"

$B03$B03.440$B03.440.475
[1] "B03.440.475"

$B03$B03.440$B03.440.500
[1] "B03.440.500"

$B03$B03.440$B03.440.520
[1] "B03.440.520"

$B03$B03.440$B03.440.540
[1] "B03.440.540"

$B03$B03.440$B03.440.595
```

```
[1] "B03.440.595"

$B03$B03.440$B03.440.602
[1] "B03.440.602"

$B03$B03.440$B03.440.612
[1] "B03.440.612"

$B03$B03.440$B03.440.614
[1] "B03.440.614"

$B03$B03.440$B03.440.623
[1] "B03.440.623"

$B03$B03.440$B03.440.645
[1] "B03.440.645"

$B03$B03.440$B03.440.646
[1] "B03.440.646"

$B03$B03.440$B03.440.647
[1] "B03.440.647"

$B03$B03.440$B03.440.680
[1] "B03.440.680"

$B03$B03.440$B03.440.840
[1] "B03.440.840"

$B03$B03.440$B03.440.860
[1] "B03.440.860"

$B03$B03.440$B03.440.930
[1] "B03.440.930"


$B03$B03.510
[1] "B03.510"

$B03$B03.650
[1] "B03.650"

$B03$B03.660
[1] "B03.660"

$B03$B03.851
[1] "B03.851"

$B03$B03.900
[1] "B03.900"

$B03$B03.950
[1] "B03.950"
```

6.6.3 `termInfo`

Usage of termInfo
termInfo(heading1, env=NULL)

Function: to retrieve hierarchy information of a given MeSH heading
Parameters:

heading: a MeSH heading name

brief: whether to retrive breif tree information of MeSH term

env: the dataset to use. As the dataset has been pre-calculated in MeSHSim package, there is no need to set "env" defaultly.

Value:

return hierarchy information of a given MeSH heading

Examples of `termInfo`

Users are able to choose to fetch brief information of a given MeSH term by setting *brief*. The default setting is retrieve all MeSH headings, whose value "TRUE"

```
> termInfo("Rhode Island")

[[1]]
[[1]]$Z01
[1] "Geographic Locations"

[[1]]$Z01.107
[1] "Americas"

[[1]]$Z01.107.567
[1] "North America"

[[1]]$Z01.107.567.875
[1] "United States"

[[1]]$Z01.107.567.875.550
[1] "New England"

[[1]]$Z01.107.567.875.550.680
[1] "Rhode Island"

> termInfo("Rhode Island",brief=FALSE)

[[1]]
[[1]]$Z01
[[1]]$Z01$term
[1] "Geographic Locations"

[[1]]$Z01$Z01.058
[1] "Z01.058"

[[1]]$Z01$Z01.107
[[1]]$Z01$Z01.107$term
[1] "Americas"

[[1]]$Z01$Z01.107$Z01.107.084
[1] "Z01.107.084"

[[1]]$Z01$Z01.107$Z01.107.169
[1] "Z01.107.169"

[[1]]$Z01$Z01.107$Z01.107.296
[1] "Z01.107.296"

[[1]]$Z01$Z01.107$Z01.107.424
[1] "Z01.107.424"
```

```
[[1]]$Z01$Z01.107$Z01.107.567
[[1]]$Z01$Z01.107$Z01.107.567$term
[1] "North America"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.176
[1] "Z01.107.567.176"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.403
[1] "Z01.107.567.403"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.589
[1] "Z01.107.567.589"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875
[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$term
[1] "United States"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.075
[1] "Z01.107.567.875.075"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.350
[1] "Z01.107.567.875.350"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.500
[1] "Z01.107.567.875.500"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.510
[1] "Z01.107.567.875.510"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.550
[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.550$term
[1] "New England"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.550$Z01.107.567.875.550.200
[1] "Z01.107.567.875.550.200"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.550$Z01.107.567.875.550.500
[1] "Z01.107.567.875.550.500"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.550$Z01.107.567.875.550.510
[1] "Z01.107.567.875.550.510"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.550$Z01.107.567.875.550.580
[1] "Z01.107.567.875.550.580"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.550$Z01.107.567.875.550.680
[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.550$Z01.107.567.875.550.680$term
[1] "Rhode Island"


[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.550$Z01.107.567.875.550.880
[1] "Z01.107.567.875.550.880"


[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.560
[1] "Z01.107.567.875.560"
```

```
[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.580
[1] "Z01.107.567.875.580"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.750
[1] "Z01.107.567.875.750"

[[1]]$Z01$Z01.107$Z01.107.567$Z01.107.567.875$Z01.107.567.875.760
[1] "Z01.107.567.875.760"



[[1]]$Z01$Z01.107$Z01.107.757
[1] "Z01.107.757"


[[1]]$Z01$Z01.158
[1] "Z01.158"

[[1]]$Z01$Z01.208
[1] "Z01.208"

[[1]]$Z01$Z01.252
[1] "Z01.252"

[[1]]$Z01$Z01.338
[1] "Z01.338"

[[1]]$Z01$Z01.433
[1] "Z01.433"

[[1]]$Z01$Z01.542
[1] "Z01.542"

[[1]]$Z01$Z01.586
[1] "Z01.586"

[[1]]$Z01$Z01.639
[1] "Z01.639"

[[1]]$Z01$Z01.678
[1] "Z01.678"

[[1]]$Z01$Z01.756
[1] "Z01.756"
```

References

Blott, S., Gurrin, C., Jones, G., Smeaton, A., and Sodring, T. (2003). On the use of mesh headings to improve retrieval effectiveness. *Text REtrieval Conference(TREC2003)*, pages 215–224.

Bulskov, H., Knappe, R., and Andreasen, T. (2002). On measuring similarity for conceptual querying. *Proceedings of the 5th International Conference on Flexible Query Answering Systems (FQASÂąÂŕ02)*, 2522:100–111.

Cohen, P. R. and Kjeldsen, R. (1987). Information retrieval by constrained spreading activation in semantic networks. *Information processing & management*, 23(4):255–268.

Gu, J., Feng, W., Zeng, J., Mamitsuka, H., and Zhu, S. (2013). Efficient semisupervised medline document clustering with mesh-semantic and global-content constraints. *IEEE T. Cybernetics*, pages 1265–1276.

Hliaoutakis, A. (2005). Semantic similarity measures in mesh ontology and their application to information retrieval on medline. *Master's thesis*.

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

Leacock, C. and Chodorow, M. (1994). Filling in a sparse training space forword sense identification. *In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics(ACLÂąÂŕ94)*.

Li, Y., Bandar, Z. A., and McLean, D. (2003a). An approach for measuring semantic similarity between words using multiple information sources. *IEEETransactions on Knowledge and Data Engineering*, 15(4):871–882.

Li, Y., Bandar, Z. A., and McLean, D. (2003b). An approach for measuring semantic similarity between words using multiple information sources. *Knowledge and Data Engineering, IEEE Transactions on*, 15(4):871–882.

Lin, D. (1993). Principle-based parsing without overgeneration. *In Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACLÂąÂŕ93)*, pages 112–120.

Lord, P., Stevens, R., Brass, A., and Goble, C. (2003). Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–1283.

Nelson, S., Schopen, M., AG, S., S.JL, and A.N (2004). The mesh translation maintenance system: Structure, interface design, and implementation. *In Proceedings of MEDINFO*.

Névéol, A. et al. (2006). Besides precision & recall: exploring alternative approaches to evaluating an automatic indexing tool for medline. *Proceedings, Washington, DC. USA, American Medical Informatics Association, Bethesda, MD, USA,*, pages 589–593.

Rada, R., Mili, H., Bichnell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Trans. Syst., Man, Cybern*, 9:17–30.

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

Richardson, R., Smeaton, A., and Murphy, J. (1994). Using wordnet as a knowledge base for measuring semantic similarity between words. *School of Computer Applications, Dublin City University*.

Wang, J. Z., Du, Z., Payattakool, R., Philip, S. Y., and Chen, C.-F. (2007). A new method to measure the semantic similarity of go terms. *Bioinformatics*, 23(10):1274–1281.

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

Zhu, S., Zeng, J., and Mamitsuka, H. (2009). Enhancing medline document clustering by incorporating mesh semantic similarity. *Bioinformatics*, 25(15):1944–1951.