Identifying Similar Non-Lattice Subgraphs in Gene Ontology based on Structural Isomorphism and Semantic Similarity of Concept Labels

Rashmie Abeysinghe[1], Xufeng Qu, MS[1], Licong Cui, PhD[1,2,⋆]
[1]Department of Computer Science, University of Kentucky, Lexington, KY
[2]Institute for Biomedical Informatics, University of Kentucky, Lexington, KY

Abstract

Non-Lattice Subgraphs (NLSs) are graph fragments of a terminology which violates the lattice property, a desirable property for a well-formed terminology. They have been proven to be useful in identifying inconsistencies in biomedical terminologies. Similar NLSs may denote similar inconsistencies that may suggest possibly similar remediations. Therefore, we investigate a structural-semantic-based approach to identify similar NLSs in the Gene Ontology (GO). For an input NLS, we first obtain all its isomorphic NLSs. Then, we compare each concept of the input NLS with the corresponding concept in an isomorphic NLS and then compute a similarity score for the two NLSs. Applying this approach to 10 different structures of NLSs in GO, we found that 38.43% (910/2368) of NLSs have at least one similar NLS. We also observed some interesting lexical patterns frequently existing in similar NLSs. Our approach may be applicable to other biomedical terminologies for identifying similar NLSs.

1 Introduction

Biomedical terminologies serve as the knowledge sources for a variety of applications in biomedical research, including data integration and management, semantic interoperability, and decision support and reasoning[1]. Existence of inconsistencies in biomedical terminologies is unavoidable due to the large size and complexity of these terminologies. Therefore, Terminology Quality Assurance (TQA) has become an important aspect of the terminology management lifecycle. However, manual review to uncover inconsistencies is tedious and labor-intensive. Hence, automating TQA has become an active area of research.

Non-Lattice Subgraphs (NLSs) are graph fragments of a terminology which violates the lattice property, a desirable structural property for a well formed terminology[2]. Recent work has shown that combining NLSs and lexical features of concepts is an effective way to identify inconsistencies in biomedical terminologies[3–5], including missing hierarchical relations, missing concepts, and incorrect hierarchical relations.

Since similar NLSs may contain similar inconsistencies, exhaustive examination of all NLSs would involve redundant work. For example, Figure 1 (above) shows two NLSs X and Y in the Gene Ontology (GO). X and Y have identical graph structures and similar concept labels in corresponding positions in the structures. Not only do they appear similar, from Figure 2 it can also be seen that their definitions are very similar as well. According to Cui et al.'s work[3], both of these two NLSs fall into a lexical pattern called *union* (e.g., in NLS X, the union of the set of words of X_1 and X_2 is equal to the set of words of X_4, which may indicate a missing subtype relation between X_3 and X_4). Thus, automatic remediations can be suggested to fix them (see Figure 1, below). It can be seen that the remediations for NLSs X and Y are similar: a missing IS-A relation between the two bottom concepts (bottom left IS-A bottom right). Figure 3 contains another two similar NLSs: A and B. Their definitions shown in Figure 4 appear similar as well. It should be noted that a remediation has not yet been found to fix either of these NLSs. Hence, when a remediation is found to fix one of them, it is highly possible that the same remediation would work for the other one as well. Therefore, identifying such similar NLSs may remove the need for redundant analysis which would lessen a great deal of manual work.

In this paper, we introduce a novel structural-semantic method to identify similar NLSs for an input NLS. Given an input NLS, we leverage graph isomorphism to obtain its structurally identical NLSs. We compute a similarity score between the input NLS and each of its structurally identical NLSs based on semantic similarity between their corresponding concept labels. To compute the similarity between concept labels, we convert labels to vectors using Doc2Vec model and then calculate the cosine similarity of the two vectors. All the structurally identical NLSs with a

⋆Corresponding author. Email: licong.cui@uky.edu

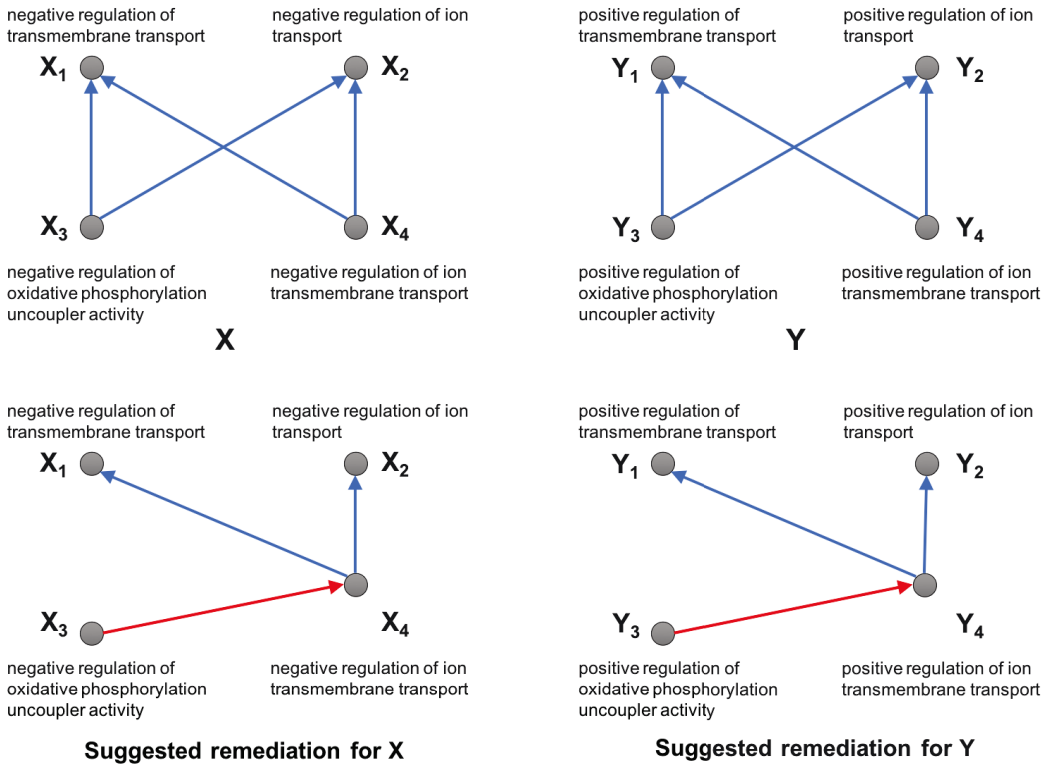

Figure 1: Above: two similar NLSs X and Y with the same structure and similar concept labels. **Below**: similar remediations for X and Y by adding a missing IS-A relation (in red) between the bottom left concept and the bottom right concept.

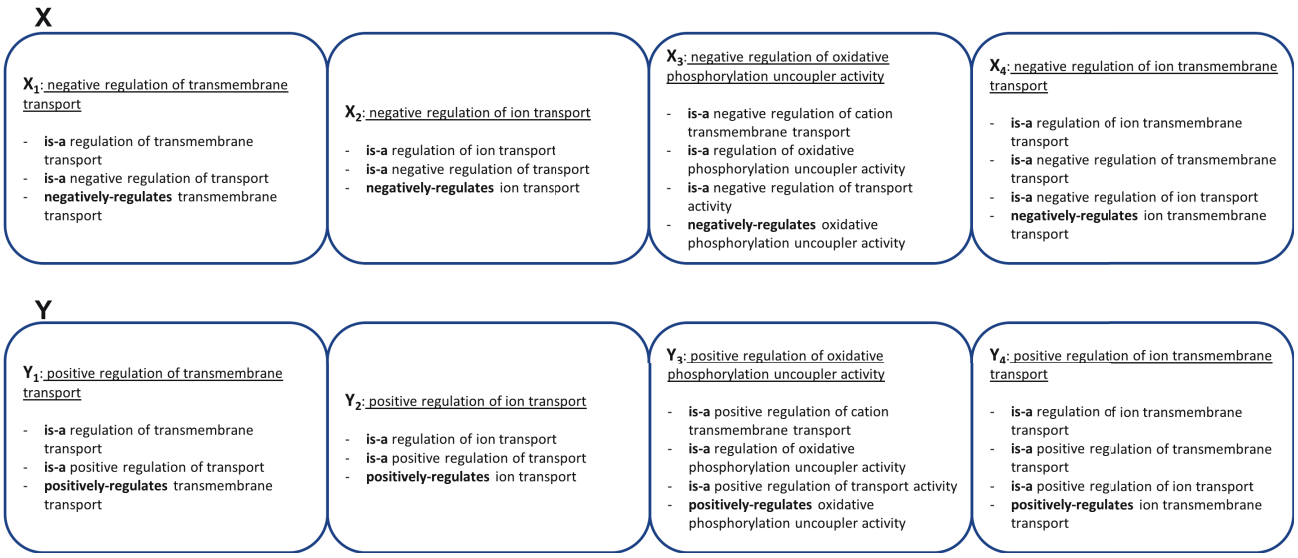

Figure 2: Definitions of the concepts of NLSs X and Y in Figure 1.

similarity score above 0.85 is considered to be similar to the input NLS. We apply this method to 10 different structures of NLSs in GO and found that 38.43% of these NLSs have at least one similar NLS.

2 Background

2.1 Terminology Quality Assurance

Various approaches have been proposed to audit biomedical terminologies such as GO and SNOMED CT[6]. Abstraction networks which are high-level summary graphs, have been widely employed to find inconsistencies in biomedical terminologies[7–10]. Different types of abstraction networks such as area, partial area taxonomies and subject-based, tribal-based abstractions have been introduced. Mougin[11] has introduced a method to identify redundant and missing relations in GO. Verspoor et al.[12] have developed an automated transformation-based clustering methodology to detect univocality violations in GO. Agrawal et al.[13] have introduced Positional Similarity Sets to uncover inconsistent modeling in SNOMED CT. Abeysinghe et al.[14] have proposed a lexical-based inference approach to detect subtype inconsistencies in GO. Jiang et al.[15] have proposed a formal concept analysis-based method to audit the semantic completeness of SNOMED CT. Zhang et al.[2] have introduced a lattice-based structural auditing approach to extract non-lattice pairs which are pairs of concepts of a terminology violating the lattice property. Cui et al.[16] have implemented a big data approach using MapReduce to exhaustively detect non-lattice pairs in biomedical terminologies.

2.2 Non-Lattice Subgraphs (NLSs)

A terminology where concepts may have multiple parents can be structured as a rooted directed acyclic graph (DAG) with respect to the IS-A taxonomic (or subtype) relationship[2,16]. Given a concept pair c_1 and c_2 in such a rooted DAG, the maximal common descendant of a concept pair c_1 and c_2 is defined as the maximal concept c^* that is both c_1 and c_2's descendant, i.e., there is no other concept c such that c is both c_1 and c_2's descendant and c^* is c's descendant. Similarly, the minimal common ancestor of a concept pair c_1 and c_2 is defined as the minimal concept c_* that is both c_1 and c_2's ancestor, i.e., there is no other concept c such that c is both c_1 and c_2's ancestor and c_* is c's ancestor. A concept pair having more than one maximal common descendant is known as a non-lattice pair[2,16]. For example, in Figure 3A, the concepts A_1 *(intrinsic component of organelle membrane)* and A_2 *(nuclear membrane part)* form a non-lattice pair, since they share two maximal common descendants A_3 *(intrinsic component of nuclear inner membrane)* and A_4 *(intrinsic component of nuclear outer membrane)*.

An NLS determined by a non-lattice pair (c_1, c_2) can be obtained by first reversely computing the minimal common ancestors of the maximal common descendants of the non-lattice pair: $mca(mcd(c_1, c_2))$ and then aggregating all the concepts and edges between (including) $mca(mcd(c_1, c_2))$ and $mcd(c_1, c_2)$[3]. Here, $mca(mcd(c_1, c_2))$ is named as the upper bounds and $mcd(c_1, c_2)$ is named as the lower bounds of the NLS. The size of the NLS is the number of concepts it contains. For example, Figure 3A contains a size-4 NLS with upper bounds A_1 and A_2 as well as lower bounds A_3 and A_4.

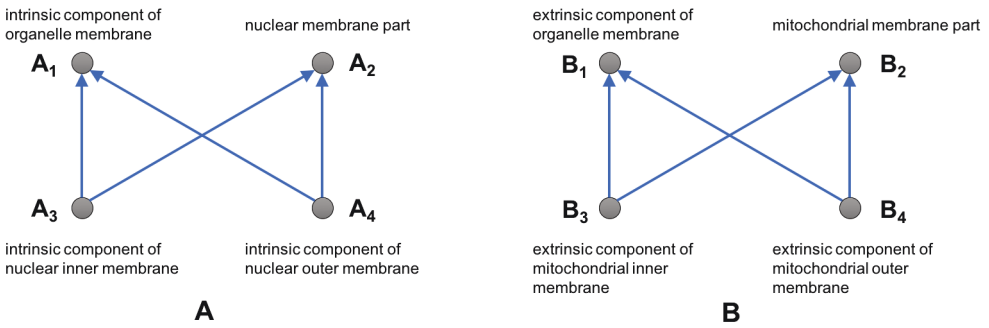

Figure 3: Two similar NLSs with the same structure and similar concept labels. Remediations for these NLSs are yet unknown.

NLSs have been shown to be effective in identifying inconsistencies in biomedical terminologies. Cui et al.[3] have identified four lexical patterns in NLSs to identify missing hierarchical relations and concepts in SNOMED CT (with a precision of 59%). Abeysinghe et al.[4] have applied that approach to the NCI Thesaurus and further introduced two new lexical patterns (with a precision of 66%). Cui et al.[5] have proposed a novel approach combining NLSs and enriched lexical attributes of concepts to detect missing hierarchical relations in SNOMED CT (with a precision of 82.96%) and uncover incorrect hierarchical relations. Although much progress has been made on mining NLSs

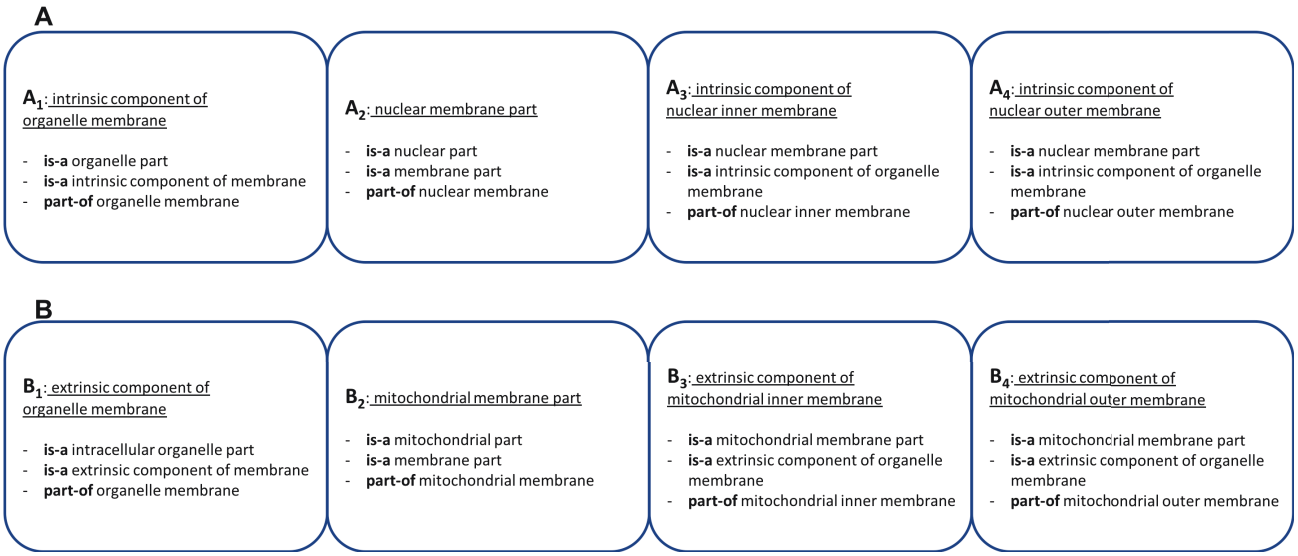

Figure 4: Definitions of the concepts of NLSs A and B in Figure 3.

for TQA, the existing work only covered a limited portion of NLSs (e.g., 7.4% of all NLSs in SNOMED CT[5]). We expect that more approaches will be explored to identify different types of inconsistencies in NLSs. In such a scenario it would be valuable to identify similar NLSs to a particular NLS whose inconsistency is found, so that redundant analysis could be avoided.

3 Methods

We use the 02-12-2018 release of GO in this work to extract NLSs. For two NLSs to be considered similar, our approach measures similarity in terms of two fronts: structural and semantic. If two NLSs are isomorphic to each other, then they satisfy the structural requirement. If the corresponding concepts between the two isomorphic NLSs are semantically similar, then they satisfy the semantic requirement.

3.1 NLS Isomorphism

Two graphs are said to be isomorphic if: (1) they contain the same number of vertices and edges, (2) the edge connectivity between the two graphs is identical. Formally, two graphs G and H with vertices $V_n = \{1, 2, 3, ..., n\}$ are isomorphic if there exists a permutation p of V_n such that u, v is in the set of graph edges $E(G)$ *iff* $p(u), p(v)$ is in the set of graph edges $E(H)$[17]. For example, Figure 3A and Figure 3B are isomorphic since they have the same number of vertices (four concepts each) and edges (for relations each) and also the concepts connected by relations in both the graphs are the same (e.g. A_3 to A_1 relation in A is similar to B_3 to B_1 relation in B). We used the VF2 algorithm[18] to compute isomorphic NLSs. VF2 is an algorithm for graph isomorphism and subgraph isomorphism which is capable of dealing with large graphs.

There could be multiple vertex mappings between two isomorphic graphs. For example, isomorphic NLSs in Figure 3A and Figure 3B have the following four different mappings between their vertices.

- A_1:B_1, A_2:B_2, A_3:B_3, A_4:B_4
- A_1:B_2, A_2:B_1, A_3:B_3, A_4:B_4
- A_1:B_1, A_2:B_2, A_3:B_4, A_4:B_3
- A_1:B_2, A_2:B_1, A_3:B_4, A_4:B_3

Therefore, for vertex A_1 in Figure 3A, the corresponding vertex in Figure 3B can be either B_1 or B_2. We take into account all the possible mappings between two isomorphic NLSs when calculating the semantic similarity.

3.2 Semantic Similarity of NLSs

For two isomorphic NLSs to be semantically similar, their concepts must be semantically similar. For example, in Figure 3, concept A_1 *(intrinsic component of organelle membrane)* in NLS A and concept B_1 *(extrinsic component of organelle membrane)* in NLS B appear similar with a word difference: *intrinsic* versus *extrinsic*.

To measure the similarity between two concepts, Doc2Vec model is used. Doc2Vec or paragraph vector is an unsupervised framework that learns continuous distributed vector representations for pieces of texts[19]. This method is applicable for variable length texts from phrases, sentences to even documents. The idea is similar to Word2Vec model which can be used to compute continuous vector representations of words[20].

First we train a Doc2Vec model considering all the concept labels in GO as inputs using the open source library Deeplearning4j[21]. Then a vector representation could be obtained for any concept label as needed. When comparing two concept labels, we simply obtain their vector representation and compute the cosine similarity between the two vectors. Because it is trained using all the concept labels in GO, the Doc2Vec model will allow to make more meaningful comparisons rather than simply comparing concepts based on the words in their labels.

To compute the similarity score between the two NLSs, we take an average of the cosine similarity scores between all the corresponding concepts of the two NLSs. Since two NLSs may have multiple mappings, we perform similarity scores for all possible mappings of the two NLSs and get the maximum score. For example, the cosine similarity scores for the mapping A_1:B_1, A_2:B_2, A_3:B_3, A_4:B_4 between the two NLSs in Figure 3A and Figure 3B are as follows:

$CosineSimilarity(A_1, B_1) = 0.9129$, $CosineSimilarity(A_2, B_2) = 0.8184$,

$CosineSimilarity(A_3, B_3) = 0.8369$, $CosineSimilarity(A_4, B_4) = 0.8384$.

Averaging these cosine similarity scores yields a similarity score of 0.8517 for the mapping. However, there are three more concept mappings for these two NLSs and their similarity scores have to be computed as well to select the maximum one.

Table 1: Different concept mappings and similarity scores for NLSs in Figure 3A and Figure 3B.

Concept mapping	Similarity score
A_1:B_1, A_2:B_2, A_3:B_3, A_4:B_4	0.8517
A_1:B_2, A_2:B_1, A_3:B_3, A_4:B_4	0.6804
A_1:B_1, A_2:B_2, A_3:B_4, A_4:B_3	0.6514
A_1:B_2, A_2:B_1, A_3:B_4, A_4:B_3	0.8226

Table 1 presents similarity scores for the four mappings that the two NLSs have. It can be seen that the mapping A_1:B_1, A_2:B_2, A_3:B_3, A_4:B_4 has the highest similarity score 0.8517. Therefore, the similarity score of the two NLSs is set to this maximum value and only this mapping with maximum score will be considered for further analysis.

In summary, for a given input NLS, our method first identifies all the isomorphic NLSs for the input NLS, and then iterates through them to find the similarity score between each of them and the input NLS. Based on our observation on a sample set, isomorphic NLSs with a similarity score greater than or equal to 0.85 is considered to be similar to the input NLS in this work.

4 Results

4.1 Summary Results

A total of 24,517 NLSs were extracted from the 02-12-2018 release of GO. We applied our method to obtain similar subgraphs for 2,368 NLSs belonging to 10 different structures shown in Figure 5. These 10 structures were the ones having the highest number of NLSs. Table 2 presents the number of NLS for each structure and the number of NLSs that have at least one similar NLS for each structure. For example, structure (i) in Figure 5 is of size 4 and has 594 NLSs. 218 of those were found to be having at least one similar NLS. Overall, 910 (38.43%) out of the 2,368 NLSs were found to be having at least one similar NLS. Figure 6 contains 8 similar NLSs obtained for an input NLS and

their similarity scores.

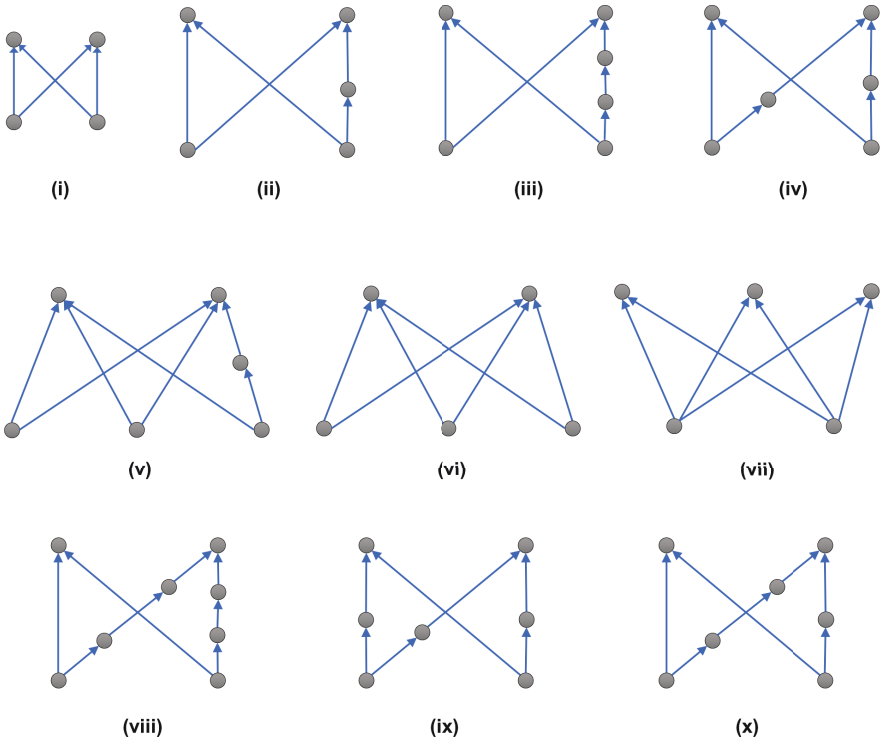

Figure 5: 10 different structures of NLSs.

Table 2: The number of NLSs in each structure in Figure 5 as well as the number of NLSs having at least one similar NLS for each structure.

NLS structure	Size	Number of NLSs	NLSs with at least one similar NLS
(i)	4	594	218
(ii)	5	432	156
(iii)	6	120	46
(iv)	6	406	165
(v)	6	107	36
(vi)	5	227	99
(vii)	5	135	63
(viii)	8	103	55
(ix)	7	105	28
(x)	7	139	44

4.2 Patterns Among Similar NLSs

We also noted some interesting patterns among similar NLSs. Figure 7 contains two similar NLSs obtained for an input NLS. Note that all the concepts between the input NLS and the similar NLS C differs in one word: *catabolic* versus *biosynthetic*. Likewise for the similar NLS D, the difference is *catabolic* versus *metabolic*. From 811 such pairs of similar NLSs, we obtained a list of high frequent patterns which can be found in Table 3. For instance, the $\{$*positive* $\Leftrightarrow$ *negative*$\}$ pattern is observed in 136 similar NLS pairs, while $\{$*negative* $\Leftrightarrow$ -$\}$ is observed in 119. The latter is obtained by concept pairs such as *negative regulation of fatty acid transport* and *regulation of fatty acid transport* where all the words of one concept is contained in another concept. Such frequently observed patterns may help understand common problems that occur in different areas in a terminology.

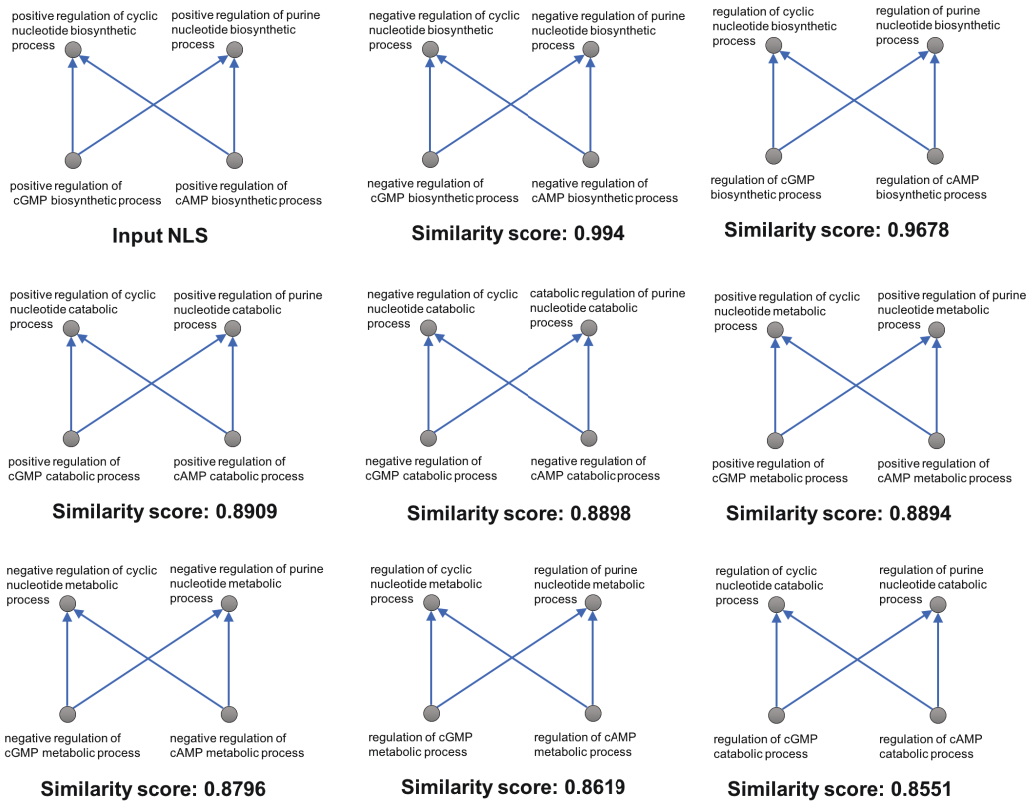

Figure 6: Similar NLSs and similarity scores for an input NLS.

Table 3: The patterns observed between similar NLSs and their frequencies.

Pattern	Frequency
positive ⇔ negative	136
negative ⇔ -	119
positive ⇔ -	105
metabolic ⇔ biosynthetic	56
catabolic ⇔ biosynthetic	37
catabolic ⇔ metabolic	33
cellular ⇔ -	10
negative regulation of ⇔ -	8
positive regulation of ⇔ -	6

5 Discussion

In this paper, we investigated an approach to identify similar NLSs to a given input NLS. We first identified isomorphic NLSs to the input NLS. Then, we computed a similarity score to measure the similarity between the input NLS and an isomorphic NLS based on their concept labels. We set a threshold (0.85) for the similarity score for two NLSs to be considered similar.

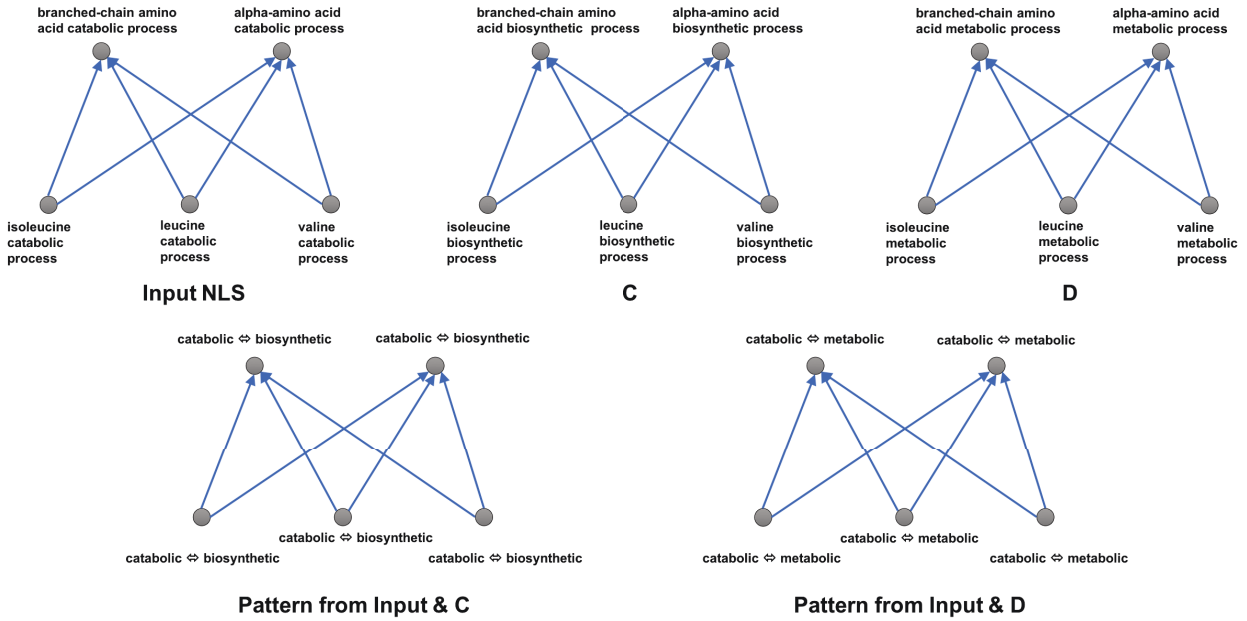

Figure 7: Similar NLSs for an input NLS and the patterns observed between them.

5.1 Analysis of Failure Cases

In this work, our goal was to identify similar NLSs with possibly similar inconsistencies so that redundant analysis to fix them could be avoided. However, this approach sometimes incorrectly identifies some NLSs to be similar to an input NLS. For instance, Figure 8 contains two NLSs which are somewhat different from each other. However, their similarity score was found to be 0.8609 which is above the threshold we set. Here, the problem may be due to two NLSs sharing two concepts effectively increasing the similarity score. An obvious solution would be to increase the threshold, but in such a scenario we may miss some similar NLSs. For example, in Figure 3, the calculated similarity score is only 0.8517 even though by observation they appear to be very similar to each other. If the threshold is increased, we may not be able to identify such cases.

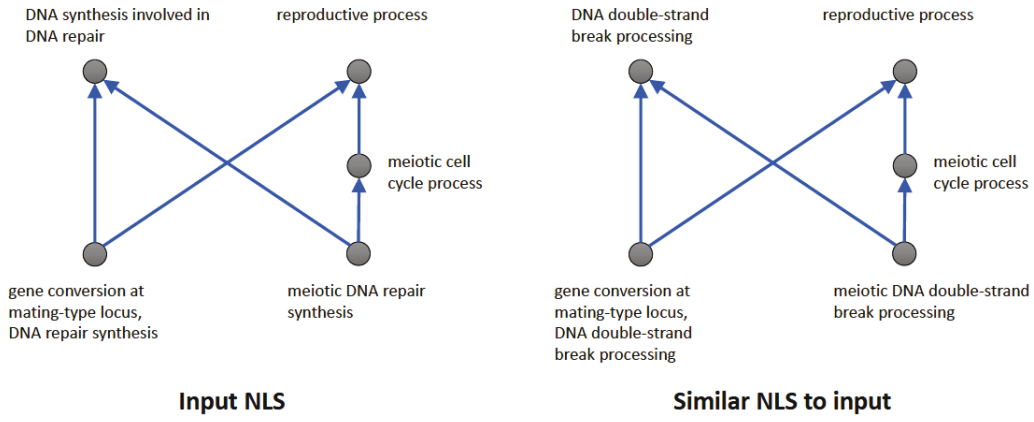

Figure 8: A somewhat dissimilar NLSs obtained for an input NLS.

5.2 Limitations and Future Work

One limitation of this work is having NLS isomorphism as a requirement for similarity. It was noted that when the size of the NLS increases, the types of its isomorphic structures also increases (e.g., size 4 has only 1 type of structure

but size 5 has 3 types). However, the number of NLSs with those structures decreases. A vast majority of large-sized (> 10) NLS structures only have a single NLS. A general approach where structural equivalence is not required but a structural similarity is required may address such cases.

Our preliminary analysis in terms of patterns that the similar NLSs exhibit, is limited since we only consider a pattern if it exists in all the concepts in the NLSs. For example, in Figure 7, the pattern observed between the input NLS and C (catabolic $\Leftrightarrow$ biosynthetic) exists in all the 5 concepts. 65% (528/811) of pairs of similar NLSs exhibited this type of patterns. However, there are others where the pattern exists in only a small portion of the NLS. For example, Figure 9 contains a different kind of pattern that is not covered in our analysis. We plan to exhaustively explore all the simple and complex patterns existing in similar NLSs in a future study.

We used the Doc2Vec model to compare concepts in NLSs in this work. We believe that the Doc2Vec model works better in making meaningful comparisons between concepts. However, we plan to compare this method with a simple word-by-word comparison method to investigate the difference between the two methods in a future study. A model combining word-by-word and Doc2Vec models may also be investigated to improve the current result. In addition, we did not leverage GO synonyms for training the Doc2Vec model, but this should be a natural extension to this study. In this way, we may be able to obtain more NLSs with at least one similar NLS. We also plan to perform a manual assessment on a random sample of NLSs to evaluate the effectiveness of our algorithm in obtaining similar NLSs.

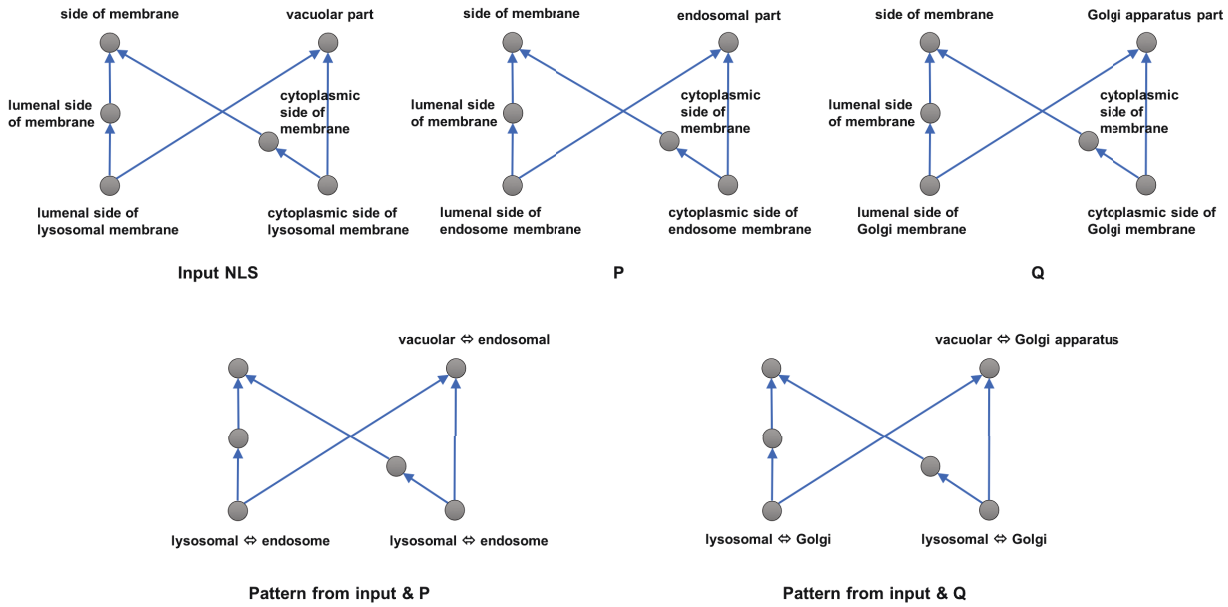

Figure 9: Two similar NLSs to an input NLS and the patterns they exhibit. This pattern was not considered in our analysis.

6 Conclusion

In this paper, we investigated a novel structural-semantic approach to obtain similar NLSs for a given input NLS. Our approach first identified all the isomorphic NLSs for the input NLS. Then the similarity between each corresponding concept label in the two NLSs was computed by converting them to vectors and calculating the cosine similarity. We trained a Doc2Vec model by all the GO concept labels and used it to convert concept labels to vectors. The similarity scores obtained between corresponding concepts in two NLSs were averaged to get the similarity score for the two NLSs. NLSs which are isomorphic to the input NLS and having a similarity score equal to or greater than 0.85 were considered to be similar to the input. We also observed some patterns exhibited in similar NLSs. Our approach is useful to avoid redundant analysis of similar NLSs. This approach is general and may be applied to other biomedical terminologies for identifying similar NLSs.

Acknowledgment

This work was supported by the National Science Foundation (NSF) through grant IIS-1657306 and the National Institutes of Health (NIH) National Cancer Institute through grant R21CA231904. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NSF or NIH.

References

1. Bodenreider O. Biomedical ontologies in action: role in knowledge management, data integration and decision support. Yearbook of medical informatics. 2008;17(01):67-79.
2. Zhang GQ Bodenreider O . Large-scale, exhaustive lattice-based structural auditing of SNOMED CT. In AMIA Annual Symposium Proceedings. 2010:922-6.
3. Cui L, Zhu W, Tao S, Case JT, Bodenreider O, Zhang GQ. Mining non-lattice subgraphs for detecting missing hierarchical relations and concepts in SNOMED CT. Journal of the American Medical Informatics Association. 2017;24(4):788-98.
4. Abeysinghe R, Brooks MA, Talbert J, Cui L. Quality assurance of NCI Thesaurus by mining structural-lexical patterns. In AMIA Annual Symposium Proceedings. 2017:364-73.
5. Cui L, Bodenreider O, Shi J, Zhang GQ. Auditing SNOMED CT hierarchical relations based on lexical features of concepts in non-lattice subgraphs. Journal of biomedical informatics. 2018;78:177-84.
6. Zhu X, Fan JW, Baorto DM, Weng C, Cimino JJ. A review of auditing methods applied to the content of controlled biomedical terminologies. Journal of Biomedical Informatics. 2009;42(3):413-25.
7. Ochs C, Geller J, Perl Y, Chen Y, Agrawal A, Case JT, Hripcsak G. A tribal abstraction network for SNOMED CT target hierarchies without attribute relationships. Journal of the American Medical Informatics Association. 2014;22(3):628-39.
8. Ochs C, Geller J, Perl Y, Chen Y, Xu J, Min H, Case JT, Wei Z. Scalable quality assurance for large SNOMED CT hierarchies using subject-based subtaxonomies. Journal of the American Medical Informatics Association. 2015;22(3):507-18.
9. Ochs C, Perl Y, Halper M, Geller J, Lomax J. Quality assurance of the gene ontology using abstraction networks. Journal of Bioinformatics and Computational Biology. 2016;14(03):1642001.
10. Wei D, Bodenreider O. Using the abstraction network in complement to description logics for quality assurance in biomedical terminologies-a case study in SNOMED CT. Studies in Health Technology and Informatics. 2010;160(P2):1070-4.
11. Mougin F. Identifying redundant and missing relations in the gene ontology. Studies in Health Technology and Informatics. 2015; 210:195-9.
12. Verspoor K, Dvorkin D, Cohen KB, Hunter L. Ontology quality assurance through analysis of term transformations. Bioinformatics. 2009;25(12):i77-84.
13. Agrawal A, Perl Y, Ochs C, Elhanan G. Algorithmic detection of inconsistent modeling among SNOMED CT concepts by combining lexical and structural indicators. In 2015 IEEE International Conference on Bioinformatics and Biomedicine. 2015:476-83.
14. Abeysinghe R, Hinderer EW, Moseley HN, Cui L. Auditing subtype inconsistencies among gene ontology concepts. In 2017 IEEE International Conference on Bioinformatics and Biomedicine. 2017:1242-45.
15. Jiang G, Chute CG. Auditing the semantic completeness of SNOMED CT using formal concept analysis. Journal of the American Medical Informatics Association. 2009;16(1):89-102.
16. Cui L, Tao S, Zhang GQ. Biomedical ontology quality assurance using a big data approach. ACM Transactions on Knowledge Discovery from Data. 2016;10(4):41.
17. Isomorphic Graphs. `http://mathworld.wolfram.com/IsomorphicGraphs.html`. Accessed March 1, 2018.
18. Cordella LP, Foggia P, Sansone C, Vento M. A (sub) graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004;26(10):1367-72.
19. Le Q, Mikolov T. Distributed representations of sentences and documents. In International Conference on Machine Learning. 2014:1188-96.
20. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013 Jan 16.
21. Eclipse Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. `http://deeplearning4j.org`. Accessed March 1, 2018.