AMIA Jt Summits Transl Sci Proc, 2015. TBI: p. 46-50.

Mining Relation Reversals in the Evolution of SNOMED CT Using MapReduce

Shiqiang Tao[1,2], MS, Licong Cui[1], PhD, Wei Zhu[1], Mengmeng Sun[1], Olivier Bodenreider[3], MD, Guo-Qiang Zhang[1,2], PhD
[1]Department of EECS, Case Western Reserve University, Cleveland, OH, USA
[2]Division of Medical Informatics, Case Western Reserve University, Cleveland, OH, USA
[3]National Library of Medicine, Bethesda, MD 20892, USA

Abstract. *Relation reversals in ontological systems refer to such patterns as a path from concept A to concept B in one version becoming a path with the position of A and B switched in another version. We present a scalable approach, using cloud computing, to systematically extract all hierarchical relation reversals among 8 SNOMED CT versions from 2009 to 2014. Taking advantage of our MapReduce algorithms for computing transitive closure and large-scale set operations, 48 reversals were found through 28 pairwise comparison of the 8 versions in 18 minutes using a 30-node local cloud, to completely cover all possible scenarios. Except for one, all such reversals occurred in three sub-hierarchies: Body Structure, Clinical Finding, and Procedure. Two (2) reversal pairs involved an uncoupling of the pair before the is-a coupling is reversed. Twelve (12) reversal pairs involved paths of length-two, and none (0) involved paths beyond length-two. Such reversals not only represent areas of potential need for additional modeling work, but also are important for identifying and handling cycles for comparative visualization of ontological evolution.*

Introduction

The focus of this paper is on ontology evolution [1, 2], most specifically on hierarchical relation reversals in SNOMED CT. A simple example of such a reversal consists of two concepts: "Joint structure of shoulder girdle" and "Joint structure of shoulder region." The 07/2013 version states that "Joint structure of shoulder girdle" is-a "Joint structure of shoulder region," although the (more recent) 03/2014 version asserts the opposite (first two components in Fig. 1): "Joint structure of shoulder region" is-a "Joint structure of shoulder girdle."

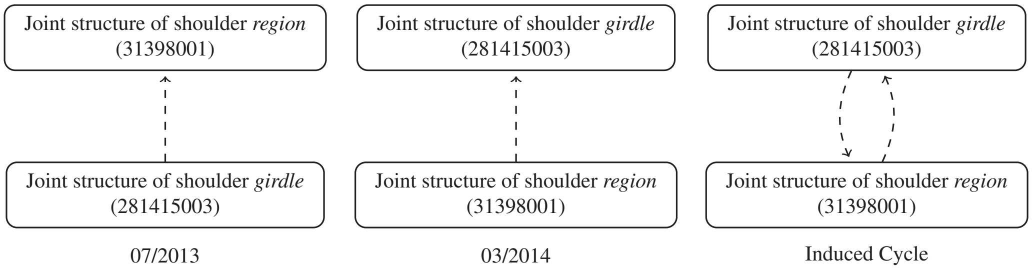
Joint structure of shoulder *region*
(31398001)
Joint structure of shoulder *girdle*
(281415003)
07/2013
Joint structure of shoulder *girdle*
(281415003)
Joint structure of shoulder *region*
(31398001)
03/2014
Joint structure of shoulder *girdle*
(281415003)
Joint structure of shoulder *region*
(31398001)
Induced Cycle

Figure 1: First two arrows: reversal of is-a relation between two versions of SNOMED CT. Right most: a cycle induced by the reversal pair when it appears in a merged graph. The numbers below concept labels are the corresponding SNOMED CT identifiers.

This is an example of a *direct (hierarchical relation) reversal*: "A is-a B" in one version has been changed to "B is-a A" in another version. An *indirect (hierarchical relation) reversal*, considered in this paper, may involve *an arbitrary number of steps* in an entire path: $A \rightarrow^* B$ in one version is changed to $B \rightarrow^* A$ in another version, where $\rightarrow^*$ represents several is-a steps in the same direction. We call the concepts A and B involved in either a direct or an indirect relation reversal *a reversal pair*.

The purpose of this study is twofold: (1) Relation reversals represent an important and rather dramatic structural change, because all the parents and children of the reversed concepts are also affected. There may be good reasons for the occurrence of such reversals that could provide us insight for improving concept labels that better reflect the intended meaning. (2) Relation reversals are important for identifying and handling cycles for comparative visualization of ontological evolution. A common, perhaps most effective, tool for rendering directed acyclic graphs in general, and hierarchical relations in ontological structures in particular, is topological sort (a.k.a. Coffman-Graham algorithm). Topological sort enables each concept assigned a unique *level*, followed by edge rendering. We are interested in visualizing ontological changes in such a way that two related fragments from different versions of the same ontology are *merged* into a single graph for visual inspection of the changes (Fig. 2). However, if a reversal pair is involved (Fig. 1, right most), it causes the merged graph cyclic, making topological sort not directly applicable. By identifying and handling reversals (the only source for introducing cycles) ahead of rendering, topological sort can still be utilized.

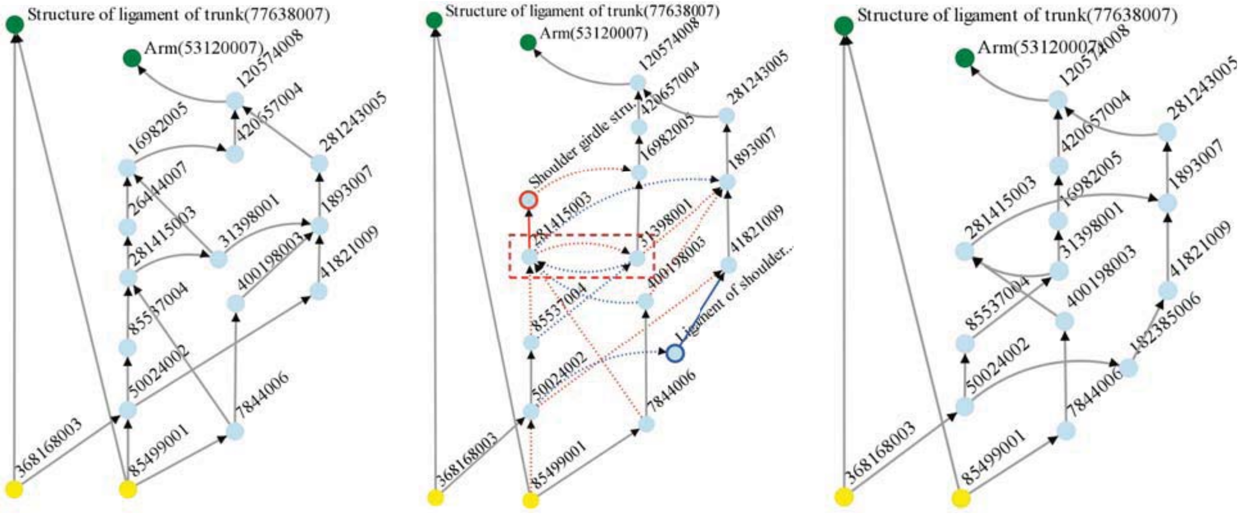

Figure 2: Semi-automatically rendered graphs from two SNOMED CT versions. Left: a non-lattice fragment of 7/2013 version of SNOMED CT. Right: a non-lattice fragment of 03/2014 version of SNOMED CT. Middle: merged graph showing the changes. The loop inside dotted red rectangle is caused by the reversal given in Fig. 1.

Mining all reversals (not just the direct ones) and between all SNOMED CT versions (not just the consecutive versions), is a computationally intensive task. We present a "Big Data" approach using MapReduce to systematically extract all such reversals among 8 SNOMED CT versions from 2009 to 2014. Taking advantage of transitive closure and a MapReduce algorithm to perform large-scale set operations, a total of 48 reversals were found among 8 SNOMED CT versions in 28 pairwise comparisons, using a total of 18 minutes with a 30-node local cloud. The systematic pairwise comparison is necessary to account for all possible situations such as "A is-a B" in a 2009 version has been changed to "B is-a A" in a 2012 version, but A and B are *not related by is-a* for all versions in 2010 and 2011. Except for one, all direct and indirect reversals occurred in three sub-hierarchies: Body Structure, Clinical Finding and Procedure.

1 Background

SNOMED CT. Developed by the International Health Terminology Standard Development Organization (IHTSDO), SNOMED CT is the world's largest clinical terminology and provides broad coverage of clinical medicine, including findings, diseases, and procedures for use in electronic medical records [3]. The international release of SNOMED CT is produced twice a year, reflecting both changes to medical knowledge (e.g., new drugs) and changes to the editorial process (e.g., changes to the representation of anatomical entities). The member countries of the IHTSDO also create extensions of SNOMED CT, with additional concepts specific to the needs of a particular country.

Ontology Evolution. Most ontologies in life science evolve continuously to account for new discoveries, to improve quality, and to align and enrich with related ontologies [4, 5, 6]. Typical ontological changes include the insertion and deletion of concepts as well as the insertion and deletion of relations between concepts. One of the values of evolutionary analysis of ontological structures is to identify regions of more intensive change activities, for targeted ontology quality assurance work [7]. Non-lattice fragments (see Fig. 2) are often indicative of structural anomalies in ontological systems. Such fragments represent possible areas of focus for subsequent quality assurance work [8] because of two reasons: (1) such structures are somewhat incompatible with the generally applicable ontology design principle that the subsumption relationship (is-a hierarchy) should form a lattice [9]; and (2) these fragments have been experimentally validated to represent change rates up to about 38 times higher than those of the overall change [10].

"Big Data" Approach for Ontology Evolution Analysis and Quality Assurance. In our work [10], we introduced MaPLE, an approach using cloud computing to systematically extract non-lattice fragments in 8 SNOMED CT versions from 2009 to 2014, with an average total compute time of less than 3 hours per version. This work used the MapReduce [11] distributed programming environment to process large amounts of data in a scalable way. A MapReduce job consists of a mapper and a reducer function, specified by the user to process data in the form of key-value pairs. Such a job is automatically broken into tasks executed in parallel across a cluster of machines called compute nodes. The results are then aggregated and grouped by the keys by reducer tasks, also executed in parallel. In this paper we use MapReduce to systematically extract all reversals among 8 SNOMED CT versions from 2009 to 2014, taking advantage of transitive closure and a MapReduce algorithm to perform large-scale set-operations.

2 Methods

Our data source consists of 8 versions of SNOMED CT, dated 07/2009, 01/2010, 01/2011, 01/2012, 07/2012, 01/2013, 07/2013, and 03/2014. To detect direct and indirect hierarchical reversals among these versions of SNOMED CT, we first compute the transitive closure for each version based on the direct "is-a" relationship. Hierarchical relation reversals are then detected by set operations of the respective transitive closures using MapReduce. If all reversal pairs are direct reversal for these 8 versions, we would like to confirm so; but it does not rule out possible indirect reversals occurring between future versions. This is the rationale for using (indirect) transitive closure, to exhaustively detect all direct and indirect reversals, including pairs that are separated by several steps in a path.

Computing Transitive Closure Using MapReduce. On average, each SNOMED CT version contains about 300k concepts and 450k "is-a" relations. Sequential algorithms for computing transitive closure, such as the Floyd-Warshall algorithm, are time-consuming. MapReduce enables a parallel, distributed way to compute transitive closure in a more efficient manner. Fig. 3 (left) is our MapReduce algorithm for computing transitive closure. First, a hash map is setup to load concepts and their direct parents in each computing node using $DistributedCache$. Then, in the map phase (lines 3-16), each mapper reads in a concept, and recursively collects its ancestors level by level, and emits the concept and the set of its ancestors. In the reduce phase (lines 17-21), each reducer emits all concept-ancestor pairs.

MapReduce for Transitive Closure

```
 1: Input: Concept nodes and "is-a" relation pairs
 2: Output: Transitive closure concept pairs
 3: class MAPPER
 4: Setup a HashMap $CP$ and load it with concepts and their direct parents using DistributedCache.
 5: method MAP(concept $c$)
 6: $P = CP$.get($c$)                                        ▷ Get direct parents of $c$
 7: $A = \emptyset$                                          ▷ Initialize a set for ancestors of $c$
 8: while $P \neq \emptyset$ do
 9:     $A$.add($P$)
10:     $temp = \emptyset$
11:     for each concept $p$ in $P$ do
12:         $temp$.add($CP$.get($p$))
13:     end for
14:     $P = temp$
15: end while
16: EMIT($c$, $A$)
17: class REDUCER
18: method REDUCE(concept $c$, concept ancestors $A$)
19: for each concept $a$ in $A$ do
20:     EMIT($c$, $a$ )                                       ▷ Output transitive closure pairs
21: end for
```

MapReduce Set Operations for Reversal

```
 1: Input: Transitive closure concept pairs for two versions $O$ and $N$
 2: Output: Direct and indirect reversals between $O$ and $N$
 3: class MAPPER
 4: method MAP(concept $c_1$, concept $c_2$)
 5: if the concept pair $(c_1, c_2)$ is in $O$ then
 6:     EMIT($(c_1, c_2)$, $O$)
 7: else if the concept pair $(c_1, c_2)$ is in $N$ then
 8:     EMIT($(c_2, c_1)$, $N$) ▷ Reverse the concept pair in $N$
 9: end if
10: class REDUCER
11: method REDUCE($(c_1, c_2)$, versions $V$)
12: if $|V| = 2$ then      ▷ The concept pair is in both $O$ and reversed $N$
13:     EMIT($c_1$, $c_2$)
14: end if
```

Figure 3: Left - MapReduce steps to compute transitive closure. Right - MapReduce steps to compute reversals.

Performing Big-Set-Operations Using MapReduce. Using the computed transitive closures for each SNOMED CT version, reaching over 5 million edges each, we detect reversals between any two versions by intersecting concept pairs in one version and reversed concept pairs in the other version. Formally, given O and N, transitive closures for two SNOMED CT versions, the set of reversals between them is $\{(c_1, c_2) \mid (c_1, c_2) \in O\} \cap \{(c_2, c_1) \mid (c_1, c_2) \in N\}$. This involves big-set-intersection, since transitive closure for each version contains a large number of concept pairs and traditional way of performing set operations does not always fit into memory. Therefore, we perform big-set-intersections in a more feasible and efficient way using MapReduce. Fig. 3 (right) shows the MapReduce algorithm to perform big-set-intersections and detect reversals between any two SNOMED CT versions. In the map stage (lines 3-9), each mapper reads in a set of concept pairs, and emits key-value pairs $((c_1, c_2), O)$ if the concept pair (c_1, c_2) is in O, and $((c_2, c_1), N)$ if the concept pair (c_1, c_2) is in N. In the reduce stage (lines 10-14), each reducer aggregates versions involved for a concept pair, and emits the concept pair if it belongs to both versions.

3 Results

Relation reversals. We performed 28 pairwise comparisons among the 8 SNOMED CT versions and found 48 reversals (Table 1). Among the 48 reversals, 33 were from the sub-hierarchy Clinical Finding, 8 from Body Structure, 6 from Procedure, and 1 from Event. Two of the reversals (rows 07/2009 → 03/2014 and 01/2010 → 07/2013 in Table 1) had intermediate stages in which the pair is not coupled by an is-a relation, confirming our strategy to perform all

1. Premature or threatened labor (287979001), Premature labor (6383007);
2. Anesthesia for procedure on head and neck (82973008), Anesthesia for procedure on head (120212000);
3. Primary dilated cardiomyopathy (195021004), Primary idiopathic dilated cardiomyopathy (53043001);
4. Computed tomography of shoulder (241564007), Computed tomography arthrogram of shoulder (241583000);
5. Musculoskeletal structure of sacral spine (297169002), Sacral spine (303950008);
6. Rupture of tendon of biceps, long head (86128003), Rupture of tendon of biceps (428883008);
7. Joint structure of shoulder girdle (281415003), Joint structure of shoulder region (31398001);
8. Calcium deposits in tendon (404224009), Osteodesmosis (404225005);
9. Sciatic neuropathy (52585001), Sciatic nerve lesion (367137004);
10. Fly bite (283345006), Mosquito bite (283344005).

Figure 4: 10 sample reversal pairs among the result of 48.

28 pairwise comparisons rather than performing comparison only for 7 consecutive versions.

Ten sample reversal pairs are displayed in Fig. 4. On average, computing the transitive closure of an entire SNOMED CT version and computing big-set-intersection between transitive closures each took less than 40 seconds, amounting to a total computing time of 18 minutes.

Version Pair	C	B	P	E
07/2009 → 01/2010	7	0	2	0
01/2010 → 01/2011	7	0	3	0
01/2011 → 01/2012	7	0	0	0
01/2012 → 07/2012	0	0	0	0
07/2012 → 01/2013	1	0	0	0
01/2013 → 07/2013	1	0	0	0
07/2013 → 03/2014	9	7	1	1
07/2009 → 03/2014	0	1	0	0
01/2010 → 07/2013	1	0	0	0
Total (**48**)	33	8	6	1

Table 1: Distribution of relation reversals in SNOMED CT sub-hierarchies and the versions in which they occurred. C: Clinical Finding. B: Body Structure. P: Procedure. E: Event.

Indirect Reversals. We found 12 indirect reversals. All such pairs involved one direct is-a relation in one version and a length-two path in the other version. Fig. 5 shows 2 such indirect reversals. This confirms the validity of our strategy to compute transitive closures of SNOMED CT versions, because using the direct relations alone would have missed such reversals. Our exhaustive analysis using transitive closure also assured that no reversals involving a path-length of more than 2 existed for the versions we analyzed. However, this does not rule out the existence of indirect reversals involving longer paths between future versions.

Enabling Visualization of Merged Fragments from Distinct SNOMED CT Versions. Our work on detecting direct and indirect reversals is also motivated by removing a technical barrier in visualizing merged ontological fragments from distinct versions of SNOMED CT. Ontological fragments in SNOMED CT can be visualized using SVG (scalable vector graphics - see Fig. 2), supported by common web browsers using the D3 drawing library (http://www.d3js.org). By convention, nodes represent concepts and edges represent "is-a" relation between concepts, with edge direction going from child (lower) to parent (higher). A well-known rendering algorithm for directed acyclic graphs is based on topological sort. We use this algorithm to render merged non-lattice fragments from different versions of SNOMED CT. However, relation reversals introduce cycles, making topological sort non-terminating. Thus detecting reversals before applying topological sort is required.

Fig. 2 illustrates an example of a semi-automatically generated merged graph of two non-lattice fragments from [10] in distinct SNOMED CT versions. The fragments are generated from Body Structure concepts "Arm (53120007)" and "Structure of ligament of trunk (77638007)" in 01/2013 and 03/2014 versions of SNOMED CT. The loop (inside the red dotted rectangle) in the graph involves the reversal pair (item 7 in Fig. 4) of "Joint structure of shoulder girdle" (281415003) and "Joint structure of shoulder region" (31398001).

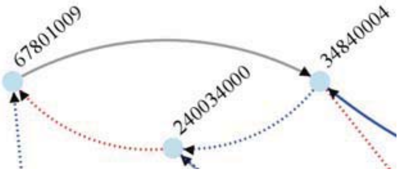

Figure 5: Two pairs of indirect reversals. One pair consists of Tendinitis AND/OR tenosynovitis (240034000) and Inflammatory disorder of tendon (34840004); the other pair consists of Tendinitis AND/OR tenosynovitis (240034000) and Tenosynovitis (67801009).

For Fig. 2, the source concept pair is colored in green. Nodes representing the greatest common descendants of the source pair are painted in yellow. All nodes lying in-between a green node and a yellow node appear in light gray. Additional graphical elements are used for visualizing graph changes (for both Fig. 2 and Fig. 5): red represents deletion and blue insertion. Nodes and edges are also marked with distinct styles to represent additional information: (a) nodes with solid red borders and solid red edges represent deletion: they appear in old fragment but not new; (b) nodes with solid blue borders and solid blue edges represent addition: they appear in new fragment but not old; (c) nodes with dashed borders, and edges drawn in dashed style, represent insertion and deletion in the SNOMED CT versions (such changes must appear in the respective fragments).

Discussion. Clinically, hierarchical relation reversals may involve concepts whose positions in the hierarchical structure are not immediately obvious. Concepts that are the object of relation reversals during ontological evolution may be worth further analysis (such as items 7 to 10 in Table 4). A majority of the reversals are consistent with two prior studies on the use of "and" and "or" [12] as well as "lexically assign, logically refine" [13].

And/Or. We found 18 reversal pairs involved the connectives "and" and "or" (or implicit logical conjunction, e.g., anorectal). For example (see items 1 and 2 in Table 4), "Premature or threatened labor, Premature labor" and "Anesthesia for procedure on head and neck, Anesthesia for procedure on head" are reversal pairs that represent a common possible misinterpretation of "and," which uses intersection in form to represent union in meaning. To make the intended meaning clearer, it is perhaps helpful to normalize the connective to "AND/OR" when the intended meaning is union, especially with respect to Body Structure and Procedure. With this convention, a concept in the form of A should always be a subclass of a concept of the form A AND/OR B. Further, all concepts "A and B" should be normalized to "A AND/OR B," so "Anesthesia for procedure on head and neck" should be normalized to "Anesthesia for procedure on head AND/OR neck" to avoid potential confusion. This is consistent with the analysis given in [12].

Lexically assign, logically refine. Examples due to this type of phenomenon include the pairs in items 3, 4, 5 and 6 in Table 4. The lexicographical difference between the pairs involves the insertion of words, such as "idiopathic" in item 3, "arthrogram" in item 4, "Musculoskeletal structure of" in item 5, and "long head" in item 6. It is arguable that any such insertion of words results in a more specialized concept, and hence should be a subclass of the parent concept. However, in the latest version we analyzed (3/2014), the opposite seems to be the case sometimes. Further consideration is needed to come up with a guiding principle (rule) that can be systematically applied.

Big Data approach for ontology quality assurance work. In this paper we refer to *"Big Data" as a frame of mind, or a "bigger vision,"* in perceiving the scientific landscape from a grander data scale, emboldened by the scalability of cloud computing, such as MapReduce for massive parallel processing. Such an approach can dramatically accelerate the speed of analysis in cases of complex tasks that are less computationally feasible [10, 14]. We believe that such a scalable approach is beneficial for ontology quality assurance work in general, even for computationally feasible problems (such as the work presented here), because it allows us to ask bigger questions and to answer them faster, putting computational barrier on the back of our minds so we can focus more on the scientific content.

Conclusion. We presented a scalable and generalizable method using MapReduce to mine reversals during ontological evolution. 48 hierarchical reversals have been found in 8 SNOMED CT versions from 2009. Identification of such reversals allowed avoidance of cycles in applying topological sort for rendering merged ontological graphs for visual comparison and change illustration. The reversals confirmed prior findings in the literature about concept labeling convention recommendations, but also revealed new cases for further consideration. In general, our closure-based technique has shown to be powerful and efficient for analyzing large ontological structures as well as their evolution. Although this investigation focused on hierarchical reversals, our approach suggests the exploration of reversals of other kinds of relations, which we plan to address in future work.

Acknowledgement. The authors acknowledge partial support by the Case Western Reserve University CTSA Grant UL1TR000439 and in part by the National Library of Medicine throughout its Intramural Research Program.

References

[1] Hartung M, Kirsten T, Gross A, Rahm E. OnEX: Exploring changes in life science ontologies. BMC Bioinformatics. 2009 Aug 13;10:250.

[2] Ceusters W. Applying Evolutionary Terminology Auditing to SNOMED CT. AMIA Annu Symp Proc. 2010 Nov 13;2010:96-100.

[3] Donnelly K. SNOMED-CT: The advanced terminology and coding system for eHealth. Stud Health Technol Inform Vol. 121, pages 279-90, 2006.

[4] Groß A, Hartung M, Prüfer K, Kelso J, Rahm E. Impact of ontology evolution on functional analyses. Bioinformatics. 2012 Oct 15;28(20):2671-7.

[5] Hartung M, Grob A, Rahm E. COnto-Diff: generation of complex evolution mappings for life science ontologies. J Biomed Inform. 2013 Feb;46(1):15-32.

[6] Kirsten T, Gross A, Hartung M, Rahm E. GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. J Biomed Semantics. 2011 Sep 13;2:6. doi: 10.1186/2041-1480-2-6.

[7] Zhu X, Wei JW, Baorto D, Weng C, Cimino J. A review of auditing methods applied to the content of controlled biomedical terminologies. J Biomedical Informatics, Vol. 42, pages 412-25, 2009.

[8] Zhang GQ and Bodenreider O. Large-scale, exhaustive lattice-based structural auditing of SNOMED CT. American Medical Informatics Association (AMIA) Annual Symposium, pages 922-926, 2010.

[9] Zweigenbaum P, Bachimont B, Bouaud J, Charlet J, Boisvieux JF. Issues in the structuring and acquisition of an ontology for medical language understanding. Methods Inf Med. Vol. 34(1-2), pages 15-24, 1995.

[10] Zhang GQ, Zhu W, Sun M, Tao S, Bodenreider O, Cui L. MaPLE: A MapReduce Pipeline for Lattice-based Evaluation of SNOMED CT. IEEE International Conference on Big Data, 2014;754-9.

[11] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. OSDI, 2004.

[12] Mendona EA, Cimino JJ, Campbell KE, Spackman KA. Reproducibility of interpreting "and" and "or" in terminology systems. Proc AMIA Symp. 1998:790-4.

[13] Dolin RH, Huff SM, Rocha RA, Spackman KA, Campbell KE. Evaluation of a "lexically assign, logically refine" strategy for semi-automated integration of overlapping terminologies. J Am Med Inform Assoc. 1998 Mar-Apr;5(2):203-13.

[14] Zhu W, Zhang GQ, Tao S, Sun M, Cui L. NEO: Systematic Non-Lattice Embedding of Ontologies for Comparing the Subsumption Relationship in SNOMED CT and in FMA Using MapReduce. AMIA Joint Summits 2015.