

Understanding Graph and Understanding Map and their Potential Applications

Gangli Liu

Tsinghua University

Beijing, China 100084

gl-liu13@mails.tsinghua.edu.cn

ABSTRACT

Based on the previously proposed concept Understanding Tree, this paper introduces two concepts: Understanding Graph and Understanding Map, and explores their potential applications. Understanding Graph and Understanding Map can be deemed as special cases of mind map, semantic network, or concept map. The two main differences are: Firstly, the data sources for constructing Understanding Map and Understanding Graph are distinctive and simple. Secondly, the relations between concepts in Understanding Graph and Understanding Map are monotonous. Based on their characteristics, applications of them include quantitatively measuring a concept's complexity degree, quantitatively measuring a concept's importance degree in a domain, and computing an optimized learning sequence for comprehending a concept etc. Further study involves evaluating their performances in these applications.

KEYWORDS

Understanding Tree; Familiarity Measure; Understanding Map; Understanding Graph; mind map; semantic network; concept map; meaningful learning

ACM Reference format:

Gangli Liu. 0000. Understanding Graph and Understanding Map and their Potential Applications. In *Proceedings of 000, Beijing, China, 0000 (0000)*, 10 pages.
DOI: 00.000/000_0

1 INTRODUCTION

In [6], Understanding Tree is introduced for evaluating a person's understanding degree to a piece of knowledge. This paper introduces two more concepts, Understanding Graph and Understanding Map, and explores their potential applications. We first recap several concepts defined in previous papers [5, 6]. Section 2 tells about Understanding Graph and its applications. Section 3 gives an account of Understanding Map and its applications. Section 4 compares Understanding Graph and Understanding Map with other knowledge organizing and representing tools. Section 5 concludes the paper.

1.1 Recap of several definitions

Here we recap several concepts mentioned in previous papers.

Knowledge Point. A Knowledge Point is a piece of knowledge which is explicitly defined and has been widely accepted. A Knowledge Point's definition may be differently phrased, however, they should be consistent.

Basic Knowledge Point (BKP). A BKP is a Knowledge Point that is simple enough so that it is not interpreted by other Knowledge Points. In a sense, they are like axioms in mathematics, serving as premises or starting points for further reasoning and arguments. In practice, it is subjective which set of Knowledge Points should be categorized as BKPs. It can be decided by a group of experts empirically.

Familiarity Measure. A Familiarity Measure is a score that depicts a person's familiarity degree to a Knowledge Point at a particular time. It is calculated with the formulas mentioned in [7].

Understanding Tree. An Understanding Tree is a treelike data structure which compiles the background Knowledge Points that are essential to understand the root Knowledge Point.

Fully Extended Understanding Tree (FEUT). A fully extended Understanding Tree is a tree like Figure 3 of [6]. It can be constructed manually by some experts of a domain, or automatically with Algorithm 1 of [6]. Its construction is based on a collection of definitions that are self-contained. 'Self-contained' means for every Knowledge Point mentioned in a definition, either it is a BKP, or it has at least one definition in the collection. The root of an FEUT is the Knowledge Point at which the construction starts. If every non-BKP has only one definition, it is intuitive to construct the FEUT. If a non-BKP has multiple definitions, and they are equivalent (e.g., the term 'topological space' has several equivalent definitions), how to process the definitions is a problem. There are several strategies for dealing with it:

- Accept all its definitions and all the Knowledge Points mentioned in the definitions;
- Pick one definition and only accept the Knowledge Points mentioned in this definition;
- Use other rules like the one mentioned in Algorithm 1 of [6].

Standard Understanding Tree (SUT). A standard Understanding Tree is a tree that is generated from an FEUT, by removing identical nodes from it. Figure 1 is an example of SUT. It is similar to Figure 4 of [6], except that the BKPs are labeled with shading and the Knowledge Points are not tagged with Familiarity Measures. Figure 2 is another example of SUT. It is constructed according to the definitions of Table 1.

| Knowledge Point | Definition | Involved Knowledge Points |
|-----------------------------|--|--|
| law of cosines (LoC) | The law of cosines relates the lengths of the sides of a triangle to the cosine of one of its angles. the law of cosines states: $c^2 = a^2 + b^2 - 2ab \cos \gamma$, where γ denotes the angle contained between sides of lengths a and b and opposite the side of length c . | length, side, triangle, cosine, angle |
| cosine | The trigonometric function that is equal to the ratio of the side adjacent to an acute angle (in a right triangle) to the hypotenuse. | trigonometric function (TF), ratio, side, acute angle (AA), right triangle (RT), hypotenuse |
| Pythagorean theorem (PT) | The Pythagorean theorem states that the square of the hypotenuse is equal to the sum of the squares of the other two sides. | square, hypotenuse, sum, side |
| hypotenuse | A hypotenuse is the longest side of a right triangle, the side opposite the right angle. | side, right triangle (RT), right angle (RA) |
| right triangle | A triangle with a right angle. | triangle, right angle (RA) |
| law of sines (LoS) | The law of sines is an equation relating the lengths of the sides of a triangle to the sines of its angles. According to the law, $\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C} = d$ where a , b , and c are the lengths of the sides of a triangle, and A , B , and C are the opposite angles, while d is the diameter of the triangle's circumcircle. | equation, length, side, triangle, sine, angle, opposite angle (OA), diameter, circumcircle |
| sine | The sine is a trigonometric function of an angle. The sine of an acute angle is defined in the context of a right triangle: for the specified angle, it is the ratio of the length of the side that is opposite that angle to the length of the longest side of the triangle (the hypotenuse). | trigonometric function (TF), angle, acute angle (AA), right triangle (RT), ratio, length, side, triangle, hypotenuse |
| circumcircle | The circumcircle of a polygon is a circle which passes through all the vertices of the polygon. | polygon, circle, vertex |
| trigonometric function (TF) | The trigonometric functions are functions of an angle. They relate the angles of a triangle to the lengths of its sides. | function, angle, triangle, length, side |

Table 1: A set of self-contained definitions

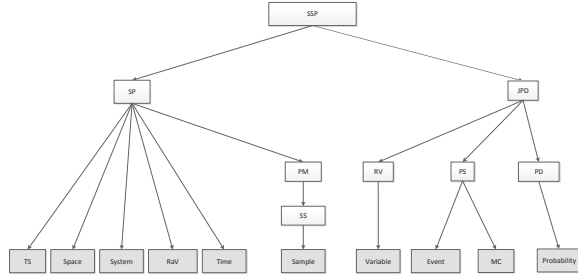


Figure 1: SUT of Strictly Stationary Process

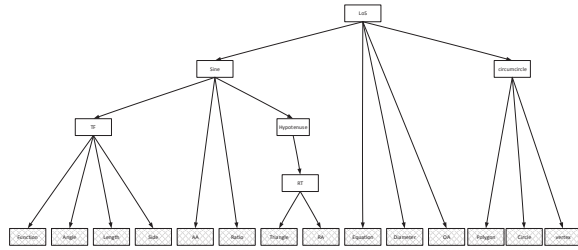


Figure 2: SUT of The Law of Sines

2 UNDERSTANDING GRAPH

An Understanding Graph is a rooted graph that is also generated from an FEUT. Instead of removing identical nodes from an FEUT,

it merges them. The merging of identical nodes keeps the connection relations (links and their directions) between Knowledge Points unchanged; only the nodes that representing the same Knowledge Point are merged into one node. E.g., Figure 3 is an Understanding Graph constructed based on Figure 3 of [6]. Figure 4 is another Understanding Graph constructed based on the definitions of Table 2. The definitions listed in Table 1 and 2 are cited from Wikipedia and other authoritative websites like Wolfram MathWorld. The shaded nodes in Figure 3 and 4 are BKPs. In Understanding Graph, each Knowledge Point is represented by one and only one node. If two Knowledge Points have the same name but with different meanings (such as homonyms), or a concept has a particular meaning in a specific field (such as the term ‘compact’ in Topology), they are differentiated with different nodes. If a Knowledge Point has multiple names, such as the law of sines, they are incorporated into one node.

2.1 Application of Understanding Graph

Several potential applications have been devised for Understanding Graph.

2.1.1 Extending keywords in Information retrieval. Information retrieval (IR) is the activity of obtaining information resources relevant to an information need from a collection of information resources. When a person has an information need, he inputs some keywords to a search engine. Usually, the more keywords are submitted, the more a search engine can know the user’s information need. However, more keywords require more cognitive burden for the user to formulate and input. With Understanding

| Knowledge Point | Definition | Involved Knowledge Points |
|---------------------|--|--|
| Poincare conjecture | Every simply connected, closed 3-manifold is homeomorphic to the 3-sphere. | simply connected, closed, 3-manifold, homeomorphic, 3-sphere |
| simply connected | A topological space is called simply-connected if it is path-connected and every path between two points can be continuously transformed, staying within the space, into any other such path while preserving the two endpoints in question. | topological space, path-connected, path, point, endpoint |
| closed manifold | A closed manifold is a type of topological space, namely a compact manifold without boundary. | topological space, compact manifold, boundary |
| 3-manifold | A topological space X is a 3-manifold if it is a second-countable Hausdorff space and if every point in X has a neighborhood that is homeomorphic to Euclidean 3-space. | topological space, second-countable, Hausdorff space, point, neighborhood, homeomorphic, Euclidean 3-space |
| homeomorphic | A homeomorphism is a continuous function between topological spaces that has a continuous inverse function. | continuous function, topological space, inverse function |
| the 3-sphere | A 3-sphere with center (C_0, C_1, C_2, C_3) and radius r is the set of all points (x_0, x_1, x_2, x_3) in real, 4-dimensional space (R_4) such that $\sum_{i=0}^3 (x_i - C_i)^2 = (x_0 - C_0)^2 + (x_1 - C_1)^2 + (x_2 - C_2)^2 + (x_3 - C_3)^2 = r^2$ | center, radius, point, real, 4-dimensional space |
| topological space | A topological space is an ordered pair (X, τ) , where X is a set and τ is a collection of subsets of X , satisfying the following axioms: (1) The empty set and X itself belong to τ . (2) Any (finite or infinite) union of members of τ still belongs to τ . (3) The intersection of any finite number of members of τ still belongs to τ . | ordered pair, set, subset, axiom, empty set, finite, infinite, union, intersection |
| path-connected | A topological space is said to be path-connected if given any two points on the topological space, there is a path starting at one point and ending at the other. | topological space, point, path |
| compact manifold | A compact manifold is a manifold that is compact as a topological space. | manifold, compact, topological space |
| boundary | The boundary of a subset S of a topological space X is the set of points which can be approached both from S and from the outside of S . | subset, topological space, point |
| Hausdorff space | The Hausdorff space is a topological space in which distinct points have disjoint neighborhoods. | topological space, point, disjoint, neighborhood |
| second-countable | A topological space is second countable if it has a countable topological basis. | topological space, topological basis |
| neighborhood | A neighborhood of a point is a set of points containing that point where one can move some amount away from that point without leaving the set. | Point, set |
| continuous function | A continuous function is a function for which sufficiently small changes in the input result in arbitrarily small changes in the output. | function, change, input, output |
| inverse function | An inverse function is a function that undoes the action of another function. | function, action |
| manifold | A manifold is a second countable Hausdorff space that is locally homeomorphic to Euclidean space. | second countable, Hausdorff space, homeomorphic, Euclidean space |
| compact | A topological space is compact if every open cover of X has a finite subcover. | topological space, open cover, subcover |
| topological basis | A topological basis is a subset B of a set T in which all other open sets can be written as unions or finite intersections of B . | subset, set, open set, union, finite, intersection |
| open cover | A collection of open sets of a topological space whose union contains a given subset. | open set, topological space, union, subset |
| subcover | Let S be a set. Let U be a cover for S . A subcover of U for S is a set $V \subseteq U$ such that V is also a cover for S . | set, cover |
| open set | Let S be a subset of a metric space. Then the set S is open if every point in S has a neighborhood lying in the set. | subset, metric space, set, point, neighborhood |
| intersection | The intersection of two sets A and B is the set of elements common to A and B . | set, element |
| subset | A subset is a portion of a set. | portion, set |
| cover | A cover of a set X is a collection of sets whose union contains X as a subset. | set, union, subset |
| metric space | A metric space is a set S with a global distance function that, for every two points x, y in S , gives the distance between them as a nonnegative real number $g(x, y)$. A metric space must also satisfy (1) $g(x, y) = 0$ iff $x = y$, (2) $g(x, y) = g(y, x)$, (3) The triangle inequality $g(x, y) + g(y, z) \geq g(x, z)$. | set, function, point, distance, nonnegative real number, triangle inequality |

Table 2: Another set of self-contained definitions

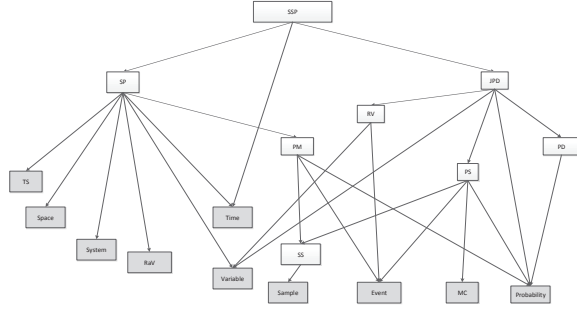


Figure 3: Understanding Graph of Strictly Stationary Process

Graph, we can extend the user's keywords automatically. E.g., if the user inputs 'Poincare Conjecture', it seems the user wants to know what Poincare Conjecture is, then the keyword is extended into "Poincare Conjecture, simply connected, closed manifold, homeomorphic, topological space, Hausdorff space, neighborhood ...". The extension can be accomplished according to a Knowledge Point's SUT, from low level nodes to high level nodes. A document or a series of documents containing all the extended keywords are good candidates for the information need.

2.1.2 Indicating the minimum information for understanding a Knowledge Point. Because an Understanding Graph is just a transformation of an Understanding Tree. The Understanding Tree of a Knowledge Point is constructed based on the definitions of itself and its descendants. It compiles the most essential information for understanding a Knowledge Point. Knowing other Knowledge Points may help understand it, but they are not the most necessary ones. E.g., knowing the Pythagorean Theorem is a special case of the Law of Cosines (LoC) helps understand the LoC, but it is not essential for understanding it. Thus it indicates the minimum information for understanding a Knowledge Point.

2.1.3 Indicating an optimized learning sequence for comprehending a Knowledge Point. In meaningful learning, the learners are 'integrating' new information into old information [8]. Meaningful learning is opposed to rote learning and refers to a learning method where the new knowledge to acquire is related with previous knowledge [1]. If a person is facing a piece of information full of concepts he does not understand, he cannot practice a meaningful learning. E.g., if you tell a person who knows little about Mathematics the content of Poincare Conjecture (now it is a theorem): "Every simply connected, closed 3-manifold is homeomorphic to the 3-sphere." He will be perplexed by what a manifold is and what do you mean by saying 'simply connected', 'closed', and 'homeomorphic'. To practice a meaningful learning, a person should have understood these concepts before facing this information. Understanding Graph provides a way to comprehend a complicated Knowledge Point in a from-easy-to-difficult order.

Take the Poincare Conjecture as an example, Figure 4 is its Understanding Graph. To learning it meaningfully, a person can learn the BKP's first. After comprehended them, they are removed from the Understanding Graph. The graph becomes Figure 5. The next learning target is determined by checking the out-degrees of the

nodes of Figure 5. The nodes with the smallest out-degrees are candidates (the light blue ones). When a node is learned and understood, it is removed from the Understanding Graph, then compares the out-degrees of the rest nodes to determine the next target. The learning process ends until the last node is removed. Because of the decaying of human memory, what have been learned may be forgotten, a person can review previously learned information at any time during the learning process.

Algorithm 1 calculates an optimized learning sequence for comprehending a Knowledge Point, based on its Understanding Graph. The subject is supposed having understood the BKP's before learning, so the learning sequence does not consider BKP's.

Algorithm 1 Calculating an optimized learning sequence for comprehending a Knowledge Point based on its Understanding Graph

Input:

Knowledge Point k_i 's Understanding Graph, G_i ;

Output:

An optimized learning sequence for comprehending k_i , S ;

- 1: Set $S = NULL$;
 - 2: Remove the BKP's and their edges from G_i ;
 - 3: Check whether G_i is $NULL$. If so, go to step 6;
 - 4: Check the out-degrees of the nodes of G_i . Select the node with the smallest out-degree, record it as n_i (if there are more than one such nodes, select one arbitrarily).
 - 5: Insert n_i into S . Remove n_i and its edges (to or from n_i) from G_i . Go to step 3;
 - 6: **return** S .
-

According to Algorithm 1 and Understanding Graph Figure 4, one of the suggested learning sequences for comprehending Poincare Conjecture is: *intersection, subset, metric space, neighborhood, continuous function, inverse function, 3-sphere, open set, cover, subcover, topological space, topological basis, open cover, compact, path-connected, simply connected, homeomorphic, boundary, Hausdorff space, second-countable, manifold, compact manifold, closed manifold, 3-manifold, Poincare Conjecture*.

Algorithm 3 of [6] also facilitates meaningful learning. The difference between Algorithm 1 and Algorithm 3 of [6] is: for Algorithm 3 of [6], there are a collection of documents to be learned, the goal is to learn the documents meaningfully. The algorithm checks the subject's current knowledge status and suggests the most appropriate document for current learning; for Algorithm 1, there is no corpus to be learned. The goal is to understand the root Knowledge Point by meaningful learning. The algorithm outputs a plan for achieving the goal. The learning process is divided into small steps. In each step, there is a small target to learn and comprehend an intermediate Knowledge Point that is essential for understanding the root. The subject is free to choose any methods to accomplish this small target, such as reading a book, surfing the Internet, or discussing with others etc. The only requirement is that the subject has understood the intermediate Knowledge Point after that step. The sequence is 'optimized' in that it recommends the easier ones first, then the harder ones, and comprehending the harder ones relies on understanding the easier ones. In a sense, the learning process is similar to let a group of sappers clear a mine-field first, before troops pass through the field. An Understanding

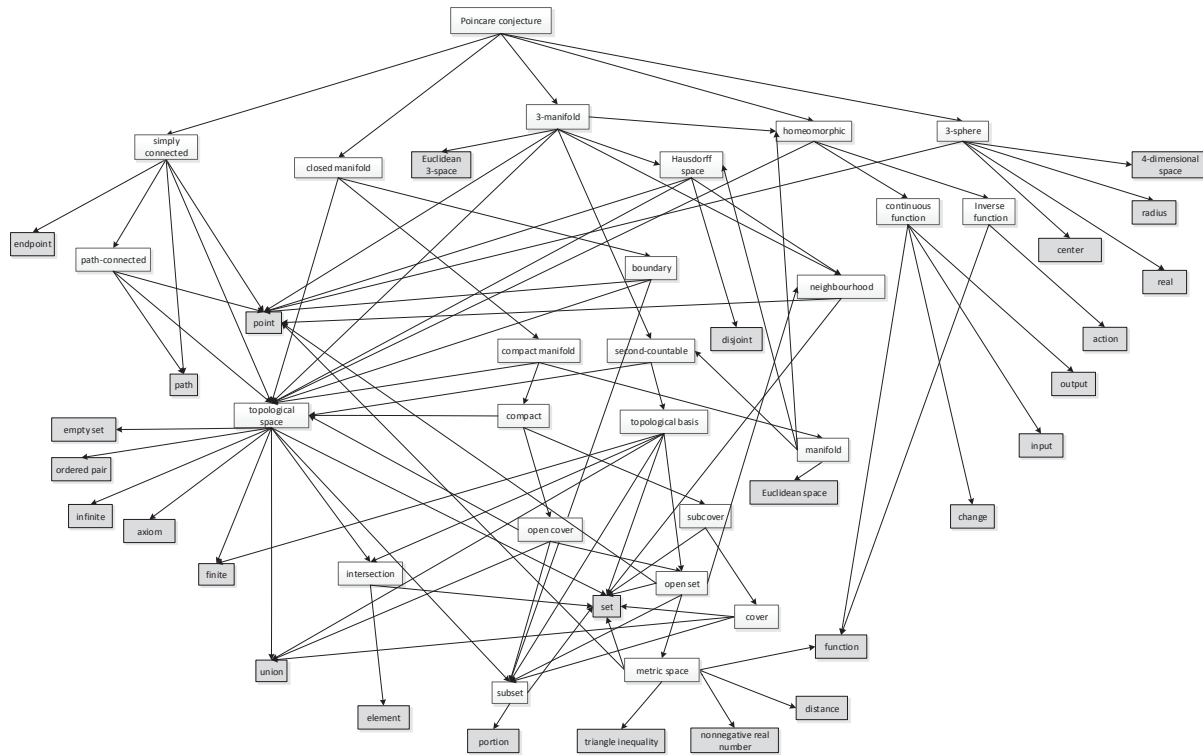


Figure 4: Understanding Graph of the Poincare Conjecture

Graph is like a map tagging where the mines are located and how they are connected; Algorithm 1 calculates an optimized sequence for clearing the mines. Therefore, it facilitates a meaningful learning path for comprehending the root Knowledge Point.

2.1.4 Measuring a Knowledge Point's complexity degree. In Understanding Graph, a Knowledge Point is represented as a bag-of-concepts and their relations. By examining the size of a Knowledge Point's Understanding Graph, its complexity degree can be obtained. E.g., by estimating empirically, we know that Poincare Conjecture is more complicated than Pythagorean Theorem. Understanding Graph facilitates a quantitative description of this relation. A Knowledge Point's complexity degree is defined as the number of vertices plus the number of edges of its Understanding Graph. Hence according to Poincare Conjecture and Pythagorean Theorem's Understanding Graphs (Figure 4 and Figure 6), the complexity degree of Poincare Conjecture equals 148; the complexity degree of Pythagorean Theorem equals 17. Based on this definition, Poincare Conjecture is indeed more complicated than Pythagorean Theorem.

In [6], I propose using the height and number of nodes of a Knowledge Point's SUT to characterize its complexity degree. In this article, I propose another method for measuring its complexity degree.

3 UNDERSTANDING MAP

Suppose we have constructed Understanding Graphs for a group of Knowledge Points, a map can be generated by merging the graphs. Understanding Map is defined as a graph which is constructed by merging a set of Understanding Graphs (at least one). The merging combines identical nodes into one node and keeps the link relations unchanged. E.g., Figure 7 is an Understanding Map constructed based on four Understanding Graphs: LoC, PT, LoS, and SSP. The Understanding Graphs of LoC, PT, and LoS are constructed based on the definitions of Table 1. Figure 3 is the Understanding Graph of SSP. Figure 4 is the Understanding Graph of Poincare Conjecture. It can also be deemed as an Understanding Map. Understanding Graph and Understanding Map are devised for different applications. The main difference is that an Understanding Graph is at least weakly connected; an Understanding Map does not need to be connected, like the one in Figure 7. Understanding Map usually contains large scale of Knowledge Points. If we combine all Knowledge Points’ Understanding Graphs, a very large Understanding Map is generated. It is called the Global Understanding Map (GUM), like a world map.

3.1 A node's n-level neighborhood

A node's n -level open neighborhood in an Understanding Map is defined as a set of nodes whose undirected distance to the node is less than or equal to n , excluding the node itself. The n -level closed neighborhood is defined in the same way but also includes the node itself. The n -level neighborhood is an extension of the

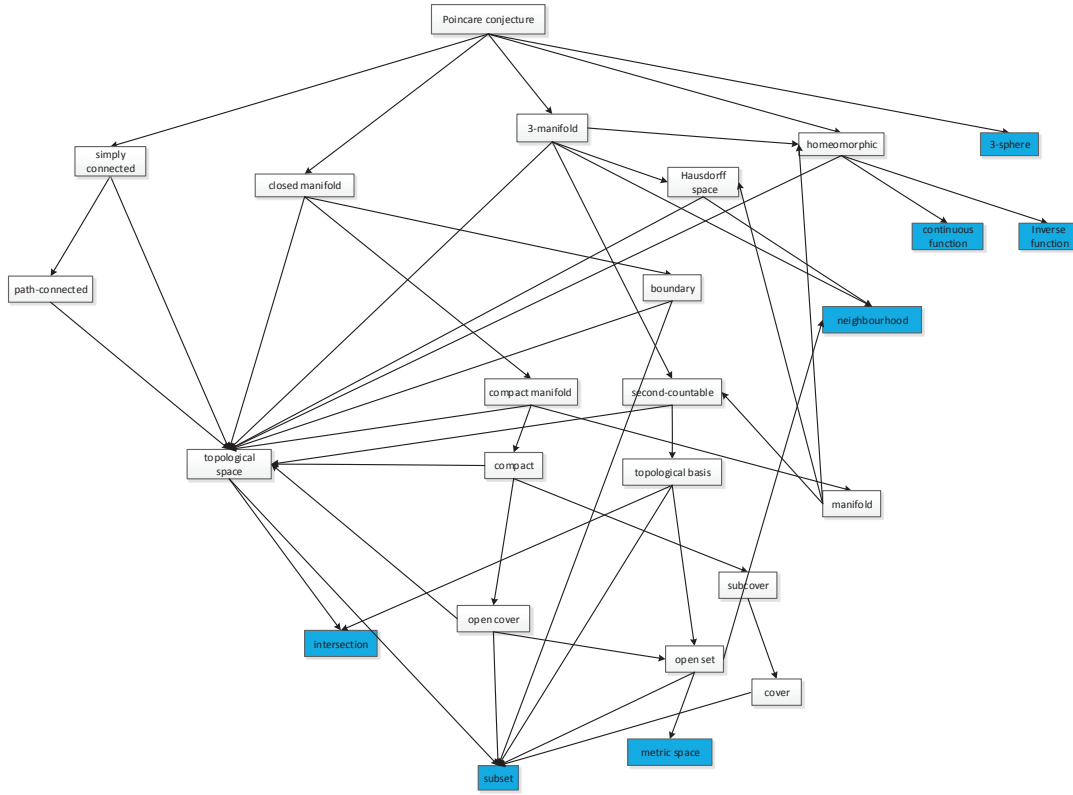


Figure 5: Understanding Graph of the Poincare Conjecture without BKPs

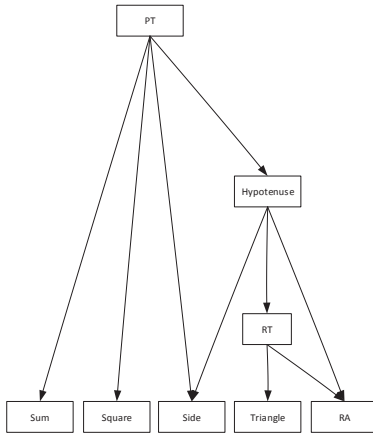


Figure 6: Understanding Graph of the Pythagorean Theorem

term ‘neighborhood’ in graph theory. Before counting distances, the edges in Understanding Map are converted to be undirected. The distance between any two vertices in a graph is the length of the shortest path having the two vertices as its endpoints. E.g., in the Understanding Map of Figure 8, the 1-level open neighborhood of the node ‘topological basis’ is a set that is comprised of the green

nodes; the 2-level open neighborhood is comprised of the green nodes plus the yellow nodes.

3.2 Application of Understanding Map

The following are some potential applications of Understanding Map.

3.2.1 Differentiating Homograph Knowledge Points (HKP). A HKP is defined as a Knowledge Point that shares the same written form with another Knowledge Point but has a different meaning. It is an extension of the term ‘homograph’ in linguistics. E.g., the Knowledge Point ‘scale’ is an HKP; it has different meanings in different context, such as:

- Scale (anatomy), a small rigid plate that grows out of an animal’s skin to provide protection.
- Scale (map), the ratio of the distance on a map to the corresponding actual distance.
- Scale (measuring), an instrument used to measure weight.
- Scale (music), a set of musical notes ordered by fundamental frequency or pitch.

In practice, people usually can differentiate a HKP’s meaning by its context without much effort, but it is not easy for a machine. Algorithm 2 provides a method for differentiating an HKP based on an Understanding Map U (preferably the Global Understanding Map) and a context C in which the HKP appears. Context C is represented by a small bag-of-words, coming from one or more

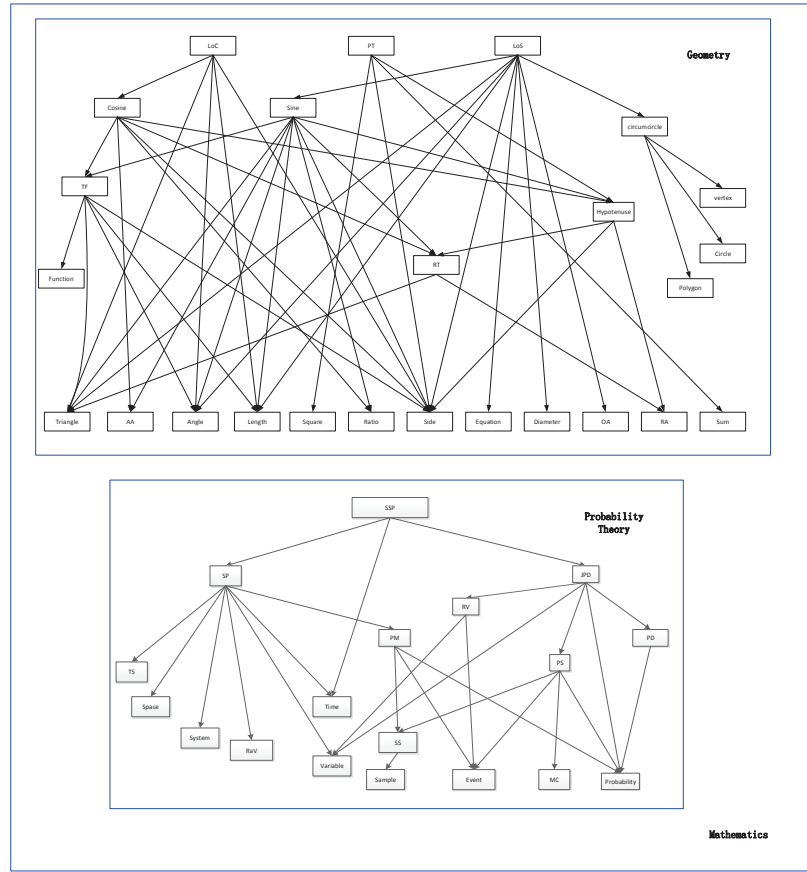


Figure 7: An exemplary Understanding Map

sentences, such as the sentences that containing or adjoining the HKP. In Understanding Map U , each meaning of the HKP possesses a different node; it is called a concrete of the HKP. Understanding Map U contains all Understanding Graphs that are related to any one of the concretes. Each concrete of the HKP has a different neighborhood in Understanding Map U . Algorithm 2 works by matching each concrete's n -level open neighborhood with the HKP's current context C . It starts at n equals 1. A concrete wins when it matches the most Knowledge Points with context C . If no concrete wins when n equals 1, then increase n by one level and compares again, until n equals k . If the winner still cannot be decided at last. The algorithm returns a *NULL*, indicating it has failed for differentiating the HKP.

3.2.2 Another way for calculating understanding degree. In [6], I have devised a method for calculating a person's understanding degree to a Knowledge Point, by examining his Familiarity Measures to the Knowledge Point's SUT. This article proposes another way for calculating the understanding degree, based on an Understanding Map (preferably the GUM). It calculates the subject's average Familiarity Measure to a Knowledge Point's n -level closed neighborhood (n is a parameter to be determined), then uses the average value as the understanding degree.

Algorithm 2 An algorithm for differentiating HKP

Input:

Understanding Map, U ; Context, C ; Homograph Knowledge Point, H ; Maximum attempts, k ;

Output:

One concrete of H according to context C , h_i ;

- 1: Set $n = 1$, $h_i = NULL$;
 - 2: If $n > k$, go to step 6;
 - 3: Calculate each concrete's n -level open neighborhood;
 - 4: Match the n -level open neighborhoods with context C . If there is only one concrete matching the most Knowledge Points, set h_i equals it, go to step 6;
 - 5: Increase n by one level. Go to step 2;
 - 6: **return** h_i .
-

3.2.3 Illustrating the knowledge characteristics of a person or a corpus. An Understanding Map is analogous to a real map. E.g., in a sense, Figure 7 is like Figure 9. A Knowledge Point is like a building; closely related Knowledge Points constitute a district, a city, or a country, just like a domain or discipline. A person lives in a city, he must be familiar with the buildings of the city than other cities. It is analogous to a person having expertise in a domain,

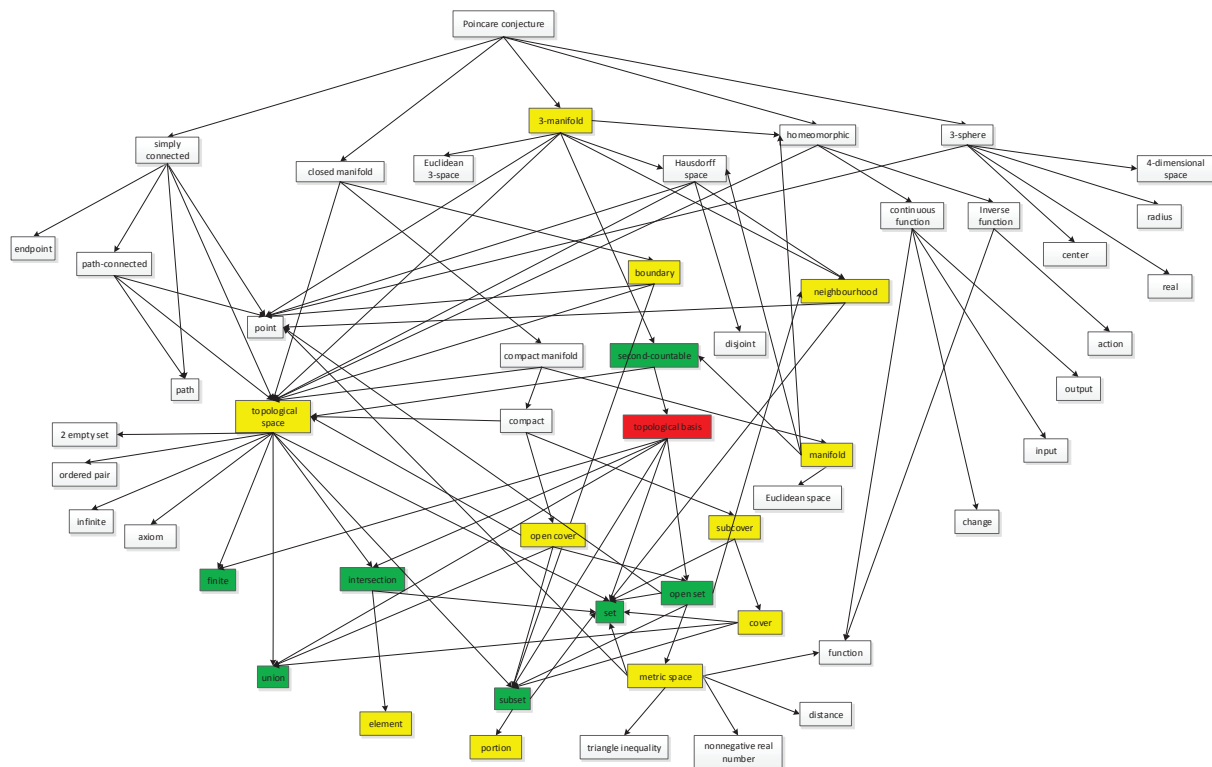


Figure 8: A node’s n-level neighborhood in an Understanding Map

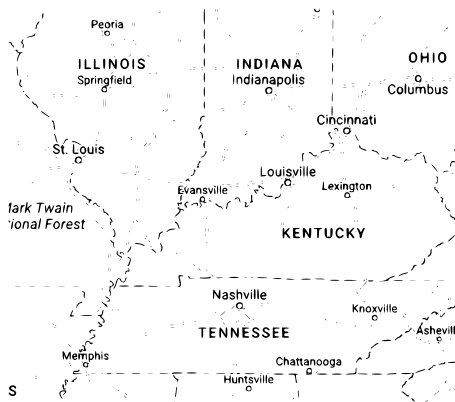


Figure 9: A part of a real map

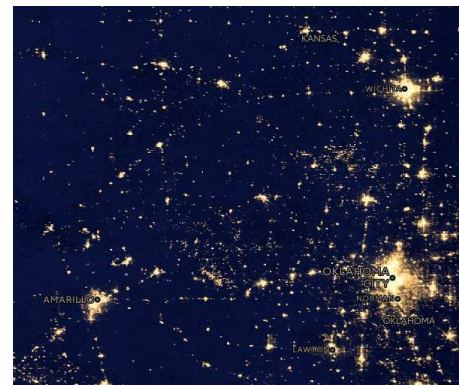


Figure 10: A part of a night light map

and being familiar with the topics of the domain. In [5] and [7], I propose a formula for calculating a person's familiarity degree to a Knowledge Point at a particular time. If we display the familiarity degrees in an Understanding Map, by setting a node's brightness being proportional to a person's familiarity degree to the node, a night light map like Figure 10 can be obtained. It illustrates a person's knowledge characteristics. A corpus can be illustrated similarly in an Understanding Map. A node's brightness can be set being proportional to its term frequency in the corpus.

3.2.4 *Measuring a Knowledge Point's importance degree in a domain.* In Understanding Graph and Understanding Map, an edge from node A to node B means understanding A relies on understanding of B. If a node is relied on by a lot of nodes, it implies the node is very important for understanding other nodes. Therefore, if we have an Understanding Map that contains all the Knowledge Points in a domain, by counting a node's in-degree, its importance rank in the domain is obtained. E.g., suppose we have a domain called "Poincare Conjecture's definition", also suppose the Understanding Map of Figure 4 contains all the Knowledge Points of this

domain, by counting each node's in-degree, the top 3 important Knowledge Points in this domain are: topological space (11), point (9), and set (9). If we ignore the BKP when ranking importance, the top 4 are: topological space (11), subset (6), homeomorphic (3), and neighborhood (3).

3.3 Conversion between Understanding Map, Understanding Graph, FEUT, and SUT

I have discussed how to convert an FEUT into an Understanding Graph and a set of Understanding Graphs into an Understanding Map. To obtain an Understanding Graph from an Understanding Map, firstly we designate the root node, then exclude nodes that are not descendants of the root. Algorithm 3 converts an Understanding Graph into a SUT. Firstly, it decides each node's level according to the root, then removes redundant edges according to some rules. An FEUT can be converted into a SUT by firstly converting it into an Understanding Graph.

Algorithm 3 Converting an Understanding Graph into a SUT

Input:

Understanding Graph, G ; The root Knowledge Point, R ;

Output:

A SUT corresponding to G , S ;

- 1: Decide each node's level according to R . A node's level is the length of the shortest path from R to it (the path is directed);
 - 2: Remove all the edges from higher level nodes to lower level ones, or between the same level from G ;
 - 3: Check each node's in-degree. If it is larger than one, keep one arbitrarily, remove others from G ;
 - 4: Set $S = G$;
 - 5: **return** S .
-

Figure 11 is a SUT converted from the Understanding Graph of Figure 4. The nodes' levels are differentiated by different colors. In SUT, a node's level indicates how important it is for understanding the root.

4 RELATED WORK

A variety of tools have been devised for organizing concepts and their relationships, such as mind map, spider diagram, semantic network, and concept map etc. A mind map is a diagram used to visually organize information. It is hierarchical and shows relationships among pieces of the whole [3]. Mind maps are considered to be a type of spider diagram. A spider diagram is a boolean expression involving unitary spider diagrams and the logical symbols conjunction, disjunction, and negation. A unitary spider diagram adds existential points to an Euler or a Venn diagram [4]. A semantic network is a network that represents semantic relations between concepts. It is a directed or undirected graph consisting of vertices, which represent concepts, and edges, which represent semantic relations between concepts [10]. Typical standardized semantic networks are expressed as semantic triples. A semantic triple is a set of three entities that codifies a statement about semantic data in the form of subject-predicate-object expressions (e.g. "a bear is a mammal", or "fish lives in water"). A concept

map is a diagram that depicts suggested relationships between concepts [2]. It typically represents ideas and information as boxes or circles, which it connects with labeled arrows in a downward-branching hierarchical structure. The relationship between concepts can be articulated in linking phrases such as causes, requires, or contributes to [9].

Understanding Graph and Understanding Map can be considered as special cases of the tools listed above. The two main differences are: Firstly, the data sources for constructing Understanding Maps and Understanding Graphs are distinctive and simple. The sufficient and prerequisite condition for constructing an Understanding Graph is a set of self-contained definitions of Knowledge Points. Merging a set of Understanding Graphs (at least one) produces an Understanding Map. Secondly, the relations between concepts in Understanding Graph and Understanding Map are monotonous. An edge from node A to node B always means understanding A relies on understanding of B. Since the relation is simplex, it is omitted in the graphs.

5 CONCLUSION

There should be a day, people will realize we have almost completed the knowledge edifice; it is time to organize our knowledge seriously, giving each concept a standardized, authoritative, and rigorous definition. Understanding Graph and Understanding Map provide a way for organizing all the well justified concepts, exploring their relationships according to their definitions. Besides proposing notions, I have explored several potential applications of the two data structures, such as quantitatively measuring a concept's complexity degree according to its Understanding Graph, quantitatively measuring a concept's importance degree in a domain according to the domain's Understanding Map, and computing an optimized learning sequence for comprehending a concept based on its Understanding Graph etc. Further study is necessary for evaluating their performances in these applications. In addition, three algorithms are devised. Algorithm 1 calculates the optimized learning sequence for comprehending a concept; Algorithm 2 differentiates an HKP based on an Understanding Map and a context in which the HKP appears; Algorithm 3 converts an Understanding Graph into a SUT.

REFERENCES

- [1] David Paul Ausubel. 2012. *The acquisition and retention of knowledge: A cognitive view*. Springer Science & Business Media.
- [2] Peter J Hager, Howard Jeffrey Scheiber, and Nancy C Corbin. 1997. *Designing & delivering: Scientific, technical, and managerial presentations*. John Wiley & Sons.
- [3] Carolyn H Hopper. 2012. *Practicing college learning strategies*. Cengage Learning.
- [4] John Howse, Gem Stapleton, and John Taylor. 2005. Spider diagrams. *LMS Journal of Computation and Mathematics* 8 (2005), 145–194.
- [5] Gangli Liu. 2016. Knowledge model: a method to evaluate an individual's knowledge quantitatively. *CoRR abs/1604.06252* (2016). arXiv:1604.06252 <http://arxiv.org/abs/1604.06252>
- [6] Gangli Liu. 2016. Understanding Tree: a tool to estimate one's understanding of knowledge. *CoRR abs/1612.07714* (2016). arXiv:1612.07714 <http://arxiv.org/abs/1612.07714>
- [7] Gangli Liu. 2017. Understanding tree: a tool for estimating an individual's understanding of conceptual knowledge. *CoRR abs/1708.00335* (2017). arXiv:1708.00335 <http://arxiv.org/abs/1708.00335>
- [8] Joseph D Novak. 2002. Meaningful learning: The essential factor for conceptual change in limited or inappropriate propositional hierarchies leading to empowerment of learners. *Science education* 86, 4 (2002), 548–571.

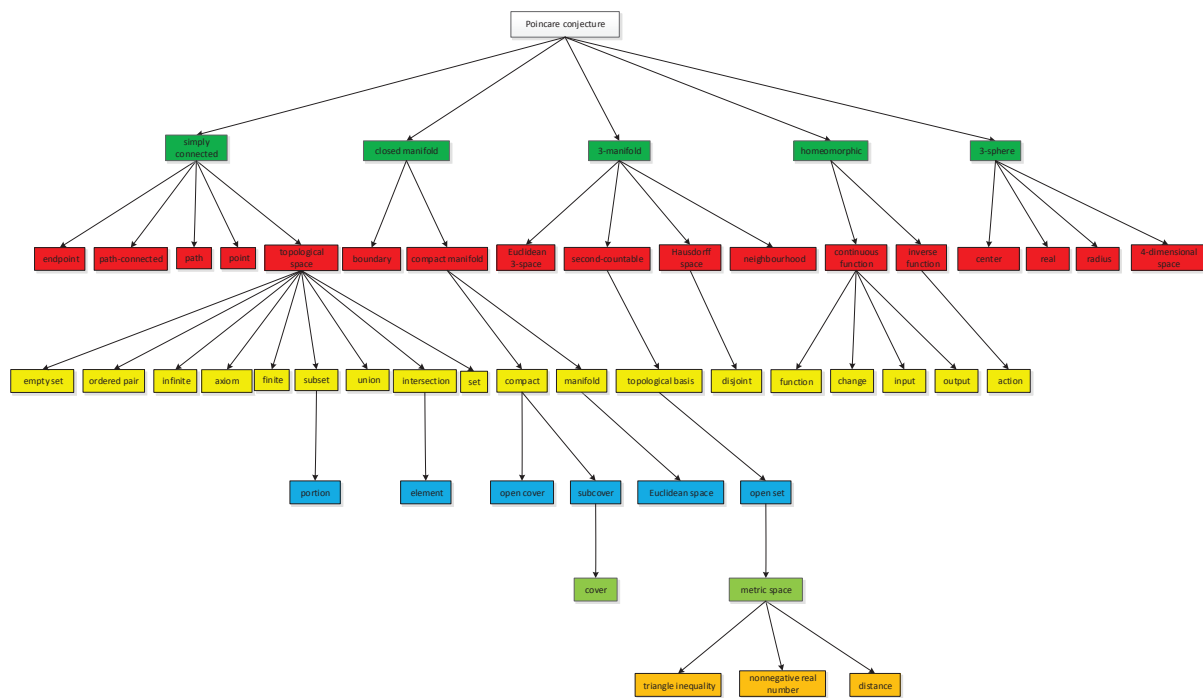


Figure 11: SUT of the Poincaré Conjecture

- [9] Joseph D Novak and Alberto J Cañas. 2008. The theory underlying concept maps and how to construct and use them. (2008).
- [10] John F Sowa. 2006. Semantic networks. *Encyclopedia of Cognitive Science* (2006).