Using Subtree Agreement for Complex Tree Integration Tasks

Marcin Maleszka and Ngoc Thanh Nguyen

Wroclaw University of Technology, Wyb. Wyspianskiego 27, 50-370 Wroclaw
Marcin.Maleszka@pwr.wroc.pl,
Ngoc-Thanh.Nguyen@pwr.edu.pl

Abstract. Hierarchical structures are common in modern applications. Tree integration is one of the tools for them that is not fully researched. We define a complex tree to model other common hierarchical structures. Complex tree integration is parametrized by specific integration criteria. Sub-tree agreement is a group of criteria that describes the relation of sub-tree number and structure between input trees and the integrated tree. This paper provides several definitions of sub-tree agreement, the most important properties of these criteria, and examples of algorithms based on sub-tree agreement.

Keywords: subtree agreement, tree integration, integration criteria, integration algorithms.

1 Introduction

Hierarchical data formats have become a common occurence in theoretical and practical applications. Even documents are nowadays stored in the XML format. Consequently, there is now a need for tools operating with hierarchical structures.

In our previous research ([10], [11] and others) we have created tools for hierarchical structures integration. By defining a generalized structured called complex tree, we are able to work with most existing structures by translating them to the new one. We have proposed the integration task for complex trees with specific integration criteria as properties of the process. These integration criteria allow easy determining of the aims in each task. In previous papers we proposed multiple such criteria and expanded on some of them, including completeness (all elements from input should remain in output), mininality (the output should not be much larger than the inputs) and optimality (the output should be a median of the inputs). In this paper we focus on the last undescribed group of integration criteria - sub-tree agreement and its variants.

The sub-tree agreement may be understood as a form of completeness - its aim may be for all sub-trees from the input to remain in the output (or similar, depending on the specific criterion used). Unfortunatelly, the methods developed for standard forms of completeness do not work with sub-trees. Consequently, it is necessary to analyze this group of criteria anew. In this paper we present

A. Selamat et al. (Eds.): ACIIDS 2013, Part II, LNAI 7803, pp. 148–157, 2013.

some most important properties of sub-tree agreement, as well as some basic integration algorithms using the criterion.

In our research we are aiming to create a collaborative recommendation system with hierarchically represented profiles. In such a system creating a centroid representing a group of profiles may be done by integrating profiles of this group. Classical approach to this is done by selecting the most average solution – in our research this is called Optimality criterion. Using Sub-Trees as an alternative allows transfering whole areas of user interest to the centroid, which may be preferable in many cases.

The rest of this paper is organized as follows: Section 2 provides a survey of related works from information retrieval and knowledge integration areas; Section 3 defines the complex tree used in this paper as well as the integration task; in Section 4 we define the sub-tree related criteria. Section 5 contains a short list of properties of these criteria, and section 6 provides an example of an algorithm maximizing the criteria. The paper is concluded with some summarizing remarks in Section 7.

2 Related Works

First research on integration of hierarchically structured data may be found in papers such as [1], [4], [6]. In those works a problem of determining a median tree was defined for structures called n-trees. At that time a single n-tree was desired an aggregation of results from multiple biochemical experiments giving different elementary trees. The inconsistencies between the input data had to be eliminated. The proposed solution was finding a so-called median tree that minimized the sum of distances to all other structures. Several approximate solutions for the problem were defined, like clusters [4] and their variations [1], [15], so-called Maximum Agreement Sub-Trees [8] or triads [6].

The Maximum Agreement Sub-Trees [8] were the main inspiration for the set of criteria presented in this paper. As stated before, the paper operates on a simple structure of n-trees. [8] does not provide an integration algorithm, but instead defines means to calculate the "distance" between two trees by calculating the number of common sub-trees. The trees are more similar, if more sub-trees are identical. In this research we use criteria based on this measure.

The domain literature also provides research that defines some basic integration criteria, similarly to the approach used by authors of this paper. The criterion of optimality first appeared in works on n-trees (but it was not always explicitly stated). There are also some works done on classification of schema integration (including hierarchical XML schemas). A survey by Rahm and Bernstein [14] provides an interesting classification of matching approaches. Schema matching in general is a much wider area than just tree integration, but with widespread of hierarchical structures in practical applications, it is also used in the area.

The research done by Do [7] describes some criteria for XML schema integration, divided into four general areas: input criteria, output criteria, quality

measures, and effort criteria. The most relevant criteria for tree integration are named by the authors as: schema information (a criterion based on the size of input schemas), schema similarity (the closer the inputs are, the smaller the space to find the output in will be), element representation (if the elements are found in the output), cardinality (cardinality of relationships in the output), precision and recall (as defined in information retrieval).

Passi [13] provides definitions for the following three basic criteria for integrating XML schemas: completeness (all elements from the initial schemas are in the merged schema), minimality (each unique element is defined only once in the schema) and understandability (in this case, a proper formatting of the output). Although those criteria are based on the criteria created for schema integration, authors modify them for integrating a constructed hierarchical structure. Further work in the area [9] modifies those criteria to postulates known in the literature: completeness and correctness (the integrated schema must contain all concepts presented in any component schema correctly; the integrated schema must be a representative of the union of the application domains associated with the schemas), minimality (if the same concept appears in more than one component schema, it must occur only once in the integrated schema) and understandability (the integrated schema should be easy to be understood for the designer and the end user; this implies that among the several possible representations of results of integration allowed by a data model, the most understandable one should be chosen). The same definitions may be found in other papers, e.q. in [2] and [3]. A thorough analysis of the minimality criterion (although not specifically for the tree structures) was done by Batista and Salgado [3] and Comyn-Wiattiau and Bouzeghoub [5].

For ontologies, integration criteria are gathered in [16], where the authors describe legibility (comprising of minimality - every element appears only once - and clarity - it is easily readable), simplicity (a minimal possible number of elements occur), correctness (all elements are properly defined), completeness (all relevant features are represented) and understandability (the ease of navigation by the user). For ontologies the scope of transformation during the integration process is much larger than for simple data structures. This is based on the fact that not only the amount of knowledge included in the integrated ontology is often greater than the sum of knowledge represented in input ontologies, but also the structure of the output might be very different form each other. The criteria are constructed to describe more what the user would gain after the transformation, less how mathematically correct the effect would be.

3 Complex Tree Integration

The research described in this paper is based on authors' previous work in [10] and [11]. These papers proposed a criteria-based approach to integration, with specific normalized criteria measures. Due to parameterizing the integration process with different criteria these papers shown that it is possible to attain different goals. For example, the completeness criterion was used to measure how

much of initial data (knowledge) was retained after integration; 0 meaning that all data was lost and 1 that all data remained. In this paper the same approach is used for Sub-Tree Agreement criteria.

This research is conducted on a specific structure, the complex tree:

Definition 1. *Complex Tree*
A complex tree is a four $t = (Y, S, V, E)$*, where:*

- Y *is a set of allowed node types in the tree*
- S *is a function determining required attributes for types*
- *the pair* (V, E) *is a a labeled tree, with nodes defined as a triple* (l, y, A)*, where:*
  - l *is the label of the node*
  - y *is the type of the node*
  - A *is the set of attributes of the node*

Additionally, the set of all complex trees will be denoted as $\mathbf{T}$.

This definition of complex tree is a basic extension of the known labeled tree. In fact, most of the criteria researched by the authors work correctly with labeled trees. The complex tree structure was adopted to allow common mathematical description for all practical structures (i.e. n-trees, XML, ontologies) modelled by complex trees.

For the complex tree the integration task may be defined as follows:

Definition 2. *Criteria-based Integration Task*
Given a multiset of N *complex trees*

$$T = \{t_1, t_2, \ldots, t_N\}$$

one should determine a complex tree $t^* \in \boldsymbol{T}$ *which best represents the trees from* T*.*

The use of words „best represents" in the definition means that t^* maximizes some defined criteria measures. In previous works we have proposed several such criteria,

In our research we use a specific description of the criteria, using normalized functions to measure their values. The arguments of these functions are the integrated tree and the input tree (for ease of readability, we use | instead of a comma to distinguish different types of arguments). A criterion is thus defined as follows:

$$C(t^*|t_1, t_2, \ldots, t_N) \geq \alpha$$

This notation represents the requirement that the criterion measure is equal or greater than the given threshold value. Thus, the integration aim is clearly stated.

4 Sub-Tree Agreement

In our previous work [cybernetics] we have defined two main criteria for Sub-Tree Agreement, based on a common definition of Sub-Tree. The Sub-Tree Agreement has several practical applications. For example, one may observe the structure of company employees before and after a corporate merger. It may be necessary to keep large sub-structures as close to the input as necessary as the cost of detail reorganization may be high, thus keeping entire divisions unchanged. This is directly translated to Initial Sub-tree Agreement, which should be maximum in that case. More detailed examples are provided at the end of this section

Definition 3. *Sub-Tree*
A Complex Tree $t_s = (Y_s, S_s, V_s, E_s)$ *is called a sub-tree of a complex tree* $t = (Y, S, V, E)$ *if* $Y_s = Y$, $S_s = S$, *and* (V_s, E_s) *is a connected sub-graph of* (V, E).

Accordingly $ST(t)$ is a set of all sub-trees of t. We will measure the size of a sub-tree or a complex tree $d(t)$ as the number of nodes in it.

Definition 4. *Initial Sub-Tree Agreement*
Initial Sub-Tree Agreement is a measure for a criterion comparing the size of the largest sub-tree from the input trees to be found in the integrated tree.

$$A_I(t^*|t_1, \ldots, t_N) = \frac{max_{t_s \in ST(t_1) \cup \ldots \cup ST(t_N)}\{d(t_s)\}}{d(t^*)} \tag{1}$$

Definition 5. *Final Sub-Tree Agreement*
Final Sub-tree Agreement is a measure for a criterion comparing the size of the largest sub-tree from the integrated tree to be found in the integrated trees.

$$A_F(t^*|t_1, \ldots, t_N) = \frac{max_{t_s \in ST(t^*)}\{d(t_s)\}}{max\{d(t_1), \ldots, d(t_N)\}} \tag{2}$$

Initial Sub-Tree Agreement and Final Sub-Tree Agreement attain the maximum value of 1 if t^* is identical to the largest of the set $\{t_1, \ldots, t_N\}$. They attain the minimum value of 0 if there is no common sub-tree in t^* and any of $\{t_1, \ldots, t_N\}$.

Alternately, the following are also proper sub-tree criteria:

Definition 6. *Input Sub-Tree Agreement*
Input Sub-Tree Agreement is a measure for a criterion comparing the number of unique subtrees in the input complex trees with the number of unique subtrees in the integrated tree.

$$A_{In}(t^*|t_1, \ldots, t_N) = min\{1, \frac{card\{ST(t_1) \cup \ldots \cup ST(t_N)\}}{card\{ST(t^*)\}}\} \tag{3}$$

Definition 7. *Output Sub-Tree Agreement*
Output Sub-Tree Agreement is a measure for a criterion comparing the number of unique subtrees in the integrated complex tree with the number of unique subtrees in the input trees.

$$A_{Out}(t^*|t_1, \ldots, t_N) = min\{1, \frac{card\{ST(t^*)\}}{card\{ST(t_1) \cup \ldots \cup ST(t_N)\}}\} \tag{4}$$

4.1 Sub-Tree Agreement in Practical Applications

Sub-Tree Agreement may be applied in several different practical applications. The most natural application is the case of employee hierarchy. Each node in the complex tree represents an employee (or, in some cases, only a work position). If a node is a parent to another, this represents a person being a direct supervisor of another. Consequently, all descendants of a node are direct and indirect underlings of some employee.

Sub-Tree Agreement becomes necessary in case of reorganizations in such structure. This may occur e.q. during a company merger. Such situation directly translates to an integration task for the employee hierarchies. Different aims are possible during this process, but here we only focus on sub-trees. A sub-tree is a representation of some estabilished department or team in one of the companies. If it is desirable for the departments and teams to be kept intact after the reorganization, the Sub-Tree Agreement should be used. The simplest approach would be using the Output Sub-Tree Agreement.

In such case, if Output Sub-Tree Agreement were to be 1, all sub-trees from the input would remain intact. Consequently, all departments from the source companies would be kept. This may lead to additional connections (or even nodes) to be created in the integrated structure - a situation that is not desirable. Thus, Output Sub-Tree Agreement of $0.8 - 0.9$ may be a better alternative. In this case, while most departments from the source companies remain unchanged, some may be removed to provide a clearer result. The application of the Minimality criterion for the addtional aim of integration may be desirable.

5 Properties of Sub-Tree Agreement

Our research determined that the various Sub-Tree Agreement criteria from the previous section have the following properties. Due to space constraints we only provide proof outlines instead of full proofs. These will be provided in other publications.

Theorem 1. *High values of Sub-Tree Agreement are possible only in cases where the Relationship Completeness and Path Completeness criteria have high values.*

Proof Outline. The Sub-Tree Agreement requires that some sub-structures of the complex tree remain the same in input and output trees. These structures have the same edges (and in lower number - paths), which means that the calculated value of Relationship Completenes (and Path Completeness) has to be high. The opposite does not hold, as the same edges in the tree may not mean the same sub-trees.

Theorem 2. *Output Sub-Tree Agreement is always higher than Input Sub-Tree Agreement.*

$$\forall_{t_1,\ldots,t_N \in \boldsymbol{T}} A_{Out}(t^*|t_1,\ldots,t_N) \geq A_{In}(t^*|t_1,\ldots,t_N)$$

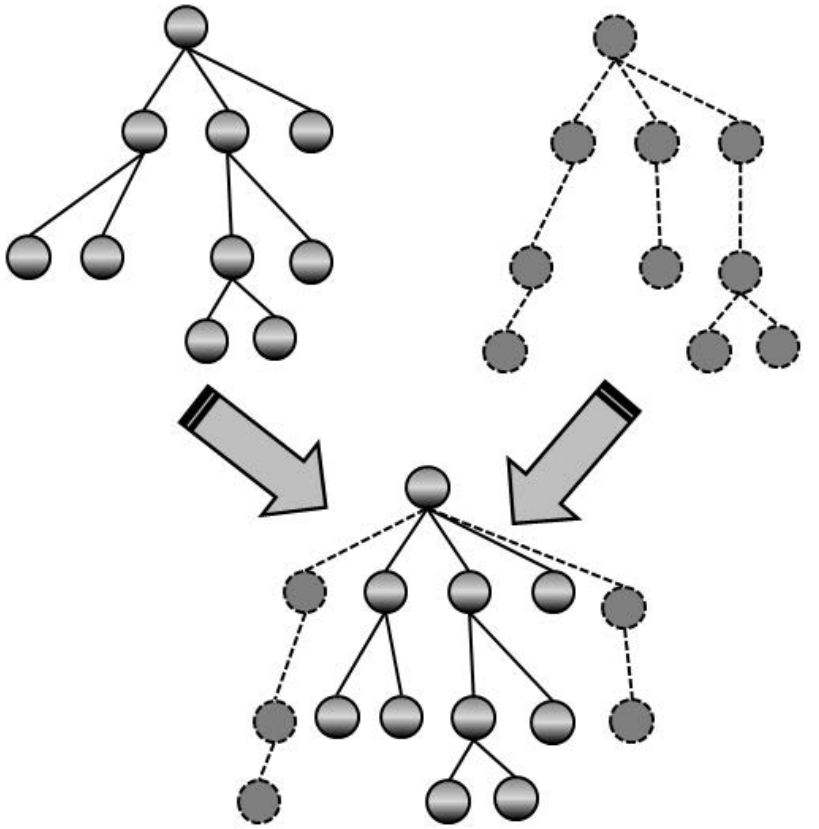

Fig. 1. Example of Sub-Tree based Integration. The left-side tree is included as a whole in the integrated tree. Two out of three main sub-trees from the right-side tree are included. The specific values of sub-tree agreement in this case are: $A_I = 1$, $A_F = 1$, $A_{In} = 1$, $A_{Out} = \frac{5}{6}$.

Proof outline. This is a natural consequence of using set cardinality in calculation. One of the cardinalities is always higher than the other.

Theorem 3. *For specific types of trees, Sub-Tree Agreement is inversly propostional to Precision criterion.*

Proof Outline. The Precision criterion is used to minimize redundancy in the integrated tree. If the same node appears in multiple sub-trees in different input trees, then high Sub-Tree Agreement requires that this node is also present in multiple locations after integration. This leads to low Precision.

Theorem 4. *For specific types of trees, Sub-Tree Agreement is inversly propostional to various Minimality criteria.*

Proof outline. Minimality criteria are used to minimize the size of integrated tree, with various size measures. The proof shows that high Minimality in some variants prevents high Sub-Tree Agreement by reducing the number of allowed sub-trees.

6 Sub-Tree Agreement Based Algorithms

Preliminary research by the authors indicates, that it is not possible to archieve maximum input or output sub-tree agreement in practical applications. Algorithms created must then provide a high value of the criterion, for example above a given threshold. Below we provide an algorithm that guarantees Final Sub-Tree Agreement equal to 1 and Input Sub-Tree Agreement above $\frac{max\{card\{ST(t_1)\},\ldots,card\{ST(t_N)\}\}}{card\{ST(t_1)\cup\ldots\cup ST(t_N)\}}$, as Algorithm 1.

The algorithm works in three simple steps:

1. Create the basic output tree by selecting the input tree with the largest sub-tree.
2. Divide other input trees into sub-trees
3. Attach selected sub-trees to the output tree

Algorithm 1. Basic STA Algorithm

Input: A set $T = \{t_1, \ldots, t_N\}$ of input trees
Output: A single output tree t_{STA}
BEGIN
Set $t_{STA} = t_1$ and $int_{max} = card\{V_1\}$;
foreach *Tree* t_i *in* T **do**
  if $card\{V_i\} > int_{max}$ **then**
    $t_{STA} = t_i$
    $int_{max} = card\{V_i\}$
Create a set of all sub-trees $ST = (ST(t_1) \cup \ldots \cup ST(t_N)) - ST(t_{STA})$
foreach *sub-tree* $st \in ST$ **do**
  if *root of st is a child of the* t_{STA}*'s root* **then**
    Add st to t_{STA}
END

Another simple algorithm is a modification of work done in [4]. In that work the authors use structures defines as clusters, that are sets of tree leafs with

a common ancestor. Such structures are very similar to sub-trees used in this paper. An algorithm modified to maximize some Sub-Tree Agreement consists of following steps:

1. For each input tree create the set of all clusters.
2. Select clusters that are to occur in the integrated tree.
3. Build the integrated tree out of the created clusters.

It may be noted that some new sub-trees may be created by using this approach, so this is not an universal solution. For example, using all clusters from step 1 in step 2 will maximize Input Sub-Tree Agreement and Final Sub-Tree Agreement, but Initial Sub-Tree Agreement and Output Sub-Tree Agreement may in some cases be smaller.

7 Conclusions

In this paper the various sub-tree agreement criteria were described. These criteria may be useful in multiple applications, with the simplest example being the case of reorganizing companies.

Multiple properties were presented for the defined criteria, with short outlines of the proofs. Relations between different criteria are the most important properties, as common applications require the use of multiple criteria – this represents multiple parallel aims in a single integration task.

Examples of basic integration algorithms were provided to show the applicability of the approach used.

In our future research we aim to use Sub-Tree Agreement criteria in a collaborative recommendation system, as described in the introduction. Initial and Input Sub-Tree Agreement types may be used to define the minimal number of user interest hierarchies that we want to transfer unchanged from the input to the integrated centroid of the group. The earlier approach of finding an average profile as the centroid is mostly satisfying, with the result representing a group of users. By slightly diverging from that solution, through the introduction of Sub-Tree Agreement criteria, we may be also able to represent more heterogeneous groups in a situation where splitting them is impossible.

Acknowledgment. This research was co-financed by Ministry of Science and Higher Education grant no. B20073/I32 and by the Fellowship co-financed by European Union within European Social Fund.

References

1. Adams, E.N.: N-Trees as Nestings: Complexity, Similarity, and Consensus. Journal of Classification 3, 299–317 (1986)
2. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. ACM Computing Surveys (CSUR)

3. Batista, M.D.C.M., Salgado, A.C.: Minimality Quality Criterion Evaluation for Integrated Schemas. In: Proceedings of 2nd International Conference on Digital Information Management, ICDIM 2007, pp. 436–441 (2007)
4. Barthelemy, J.P., McMorris, F.R.: The Median Procedure for n-Trees. Journal of Classification 3, 329–334 (1986)
5. Comyn-Wattiau, I., Bouzeghoub, M.: Constraint Confrontation: An Important Step in View Integration. In: Loucopoulos, P. (ed.) CAiSE 1992. LNCS, vol. 593, pp. 507–523. Springer, Heidelberg (1992)
6. Day, W.H.E.: Optimal Algorithms for Comparing Trees with Labeled Leaves. Journal of Classification 2, 7–28 (1985)
7. Do, H.-H., Melnik, S., Rahm, E.: Comparison of Schema Matching Evaluations. In: Chaudhri, A.B., Jeckle, M., Rahm, E., Unland, R. (eds.) NODe-WS 2002. LNCS, vol. 2593, pp. 221–237. Springer, Heidelberg (2003)
8. Farach, M., Przytycka, T.M., Thorup, M.: On the agreement of many trees. Information Processing Letters 55, 297–301 (1995)
9. Madria, S., Passi, K., Bhowmick, S.: An XML Schema integration and query mechanism system. Data & Knowledge Engineering 65, 266–303 (2008)
10. Maleszka, M., Nguyen, N.T.: Path-Oriented Integration Method for Complex Trees. In: Jezic, G., Kusek, M., Nguyen, N.-T., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2012. LNCS (LNAI), vol. 7327, pp. 84–93. Springer, Heidelberg (2012)
11. Maleszka, M., Nguyen, N.T.: A Method for Complex Hierarchical Data Integration. Cybernetics and Systems 42(5), 358–378 (2011)
12. Nguyen, N.T.: Inconsistency of Knowledge and Collective Intelligence. Cybernetics and Systems 39(6), 542–562 (2008)
13. Passi, K., Lane, L., Madria, S., Sakamuri, B.C., Mohania, M., Bhowmick, S.: A Model for XML Schema Integration. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) EC-Web 2002. LNCS, vol. 2455, pp. 193–202. Springer, Heidelberg (2002)
14. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal 10, 334–350 (2001)
15. Stinebrickner, R.: s-Consensus Trees and Indices. Bulletin of Mathematical Biology 46, 923–935 (1984)
16. Trinkunas, J., Vasilecas, O.: Ontology Transformation: from Requirements to Conceptual Model. Scientific Papers, University of Latvia, Computer Science and Information Technologies 751, 52–64 (2009)