Context-Aware Dynamic Object Relationship Modeling

Kentaroh Toyoda, Rachel Gan Kai Ying, Tan Puay Siew, and Allan Neng Sheng Zhang

Abstract—**Finding relationships in the data is essential for object modeling. However, existing methods generally focus on pre-defined static relationships using semantics and ontology, which is inappropriate when we are interested in dynamic relationships between objects that appear in data sources (e.g. log files). In this paper, we propose two novel methods to dynamically extract contextual relationships that appear in heterogeneous data sources. Our method detects contexts (e.g. time and location) in a given data source and quantifies the similarities between objects based on the detected contexts. Specifically, our methods consist of (i) a fast and accurate context detection method with carefully engineered discriminative features and (ii) a similarity measure that takes into account contexts. We evaluated our context detection method with an open dataset to show its detection accuracy and speed.**

Index Terms—**Context-aware approach, dynamic modeling, object relationship modeling, similarity measure.**

I. Introduction

Object modeling is a powerful tool to describe the relationships between objects (e.g., [1], [2]). Ontology has been widely used to semantically model the relationships among objects (e.g., [3], [4]). For instance, the relationships between machines and sensors in the manufacturing factories can be described with languages for ontologies (e.g., Web Ontology Language (OWL) [5] and Semantic Web Rule Language (SWRL) [6]). Relationships may change over time even if the objects themselves are not changed. Although such languages can handle pre-defined, static relationships very well, they cannot model *dynamic* object relationships such as the following:

Example 1. *In the supply chain, when two parcels, parcel A and parcel B, are delivered together in the same vehicle, we can say that they have strong temporal relationships. However, once they are delivered to different locations later, their relationships become weaker.*

In this example, it is obvious that we cannot use ontology-based modeling as it does not incorporate contexts that dynamically change. To capture such dynamic relationships, we believe that it can be defined as similarities between given objects that take into account their contexts. When doing so, two challenges must be solved. First, we must often handle multiple heterogeneous data sources. For instance, we may have two different event logs of a purchase order (PO) and a work order (WO), where they have different data formats. Hence, it is necessary to extract the common contexts (e.g., time and location) to streamline data sources. Second, as contexts are often a mixture of characters (e.g., address, datetime, description) and numbers, it would be better to treat them as quantitative values as much as possible, and the similarity measure must handle both numbers and characters well.

In this paper, we propose two methods to tackle the above issues: i) a context detection method with a supervised machine learning classifier and ii) a context-aware similarity measure for heterogeneous data types. The former is to detect latent contexts in given data as they are often implicit. The latter is to convert contextual information into numbers so that we can make use of it in calculating similarities. Specifically, if a detected context is (date)time or location, then we convert its value (e.g. "2021-12-01 01:02:03" or "2 Fusionopolis Way, Singapore 138634") into a numeric value of Unix epoch or a pair of latitude and longitude (e.g. "1638291723" or "1.298446, 103.78857"). However, we still have to deal with both numeric and character values. Hence, we use a decision-tree-based similarity measure (e.g., [7]) which can incorporate them well and be successful in many domains (e.g., [8], [9]).

We evaluated our methods with a publicly available dataset. Our context detection method outperforms the state-of-the-art semantic type detection method [10] in terms of detection accuracy (6-12% improvement in F_1 score) and speed (×500 improvement of computation time).

The rest of this paper is organized as follows: Section II summarizes related work. Section III states objectives of this research. Section IV describes the proposed method, while Section V presents performance evaluation. Finally, Section VI concludes this paper.

II. Related Work

Our work is closely related with three topics, namely (i) ontology-based object modeling, (ii) dynamic graph, and (iii) process mining. In this section, we summarize key research work for each topic.

A. Ontology-Based Object Modeling

An ontology is a way of representation of rich and complex knowledge about groups of objects and their relationships. The OWL and SWRL are well-known modeling languages designed to describe such an object relationship [5], [6]. For instance, using their syntax, a rule asserting that the composition of parent and brother properties implies the uncle property would be written:

$$\texttt{parent}(x, y) \wedge \texttt{brother}(y, z) \Rightarrow \texttt{uncle}(x, z).$$

Manuscript received October 25, 2021; revised February 10, 2022. This work was supported by the A*STAR Cyber-Physical Production System (CPPS) - Towards Contextual and Intelligent Response Research Program, under the RIE2020 IAF-PP Grant A19C1a0018, and Model Factory@SIMTech.

The authors are with the Singapore Institute of Manufacturing Technology, A*STAR, 2 Fusionopolis Way, 138634 Singapore (email: kentaroh.toyoda@ieee.org).

DOI: 10.7763/IJMO.2022.V12.798

By interpreting such a syntax, we can visualize the relationship between objects with graphs. OWL and SWRL are well utilized in many domains such as industry, logistics, and web to name a few. For example, Giustozzi *et al.* proposed the context ontology, as shown in Fig. 1, which models the relationships between industrial objects and contexts (e.g., machines, sensors, manufacturing process, location, time, etc.) [4]. Another example in industrial domain is a product information modeling for product lifecycle management [11].

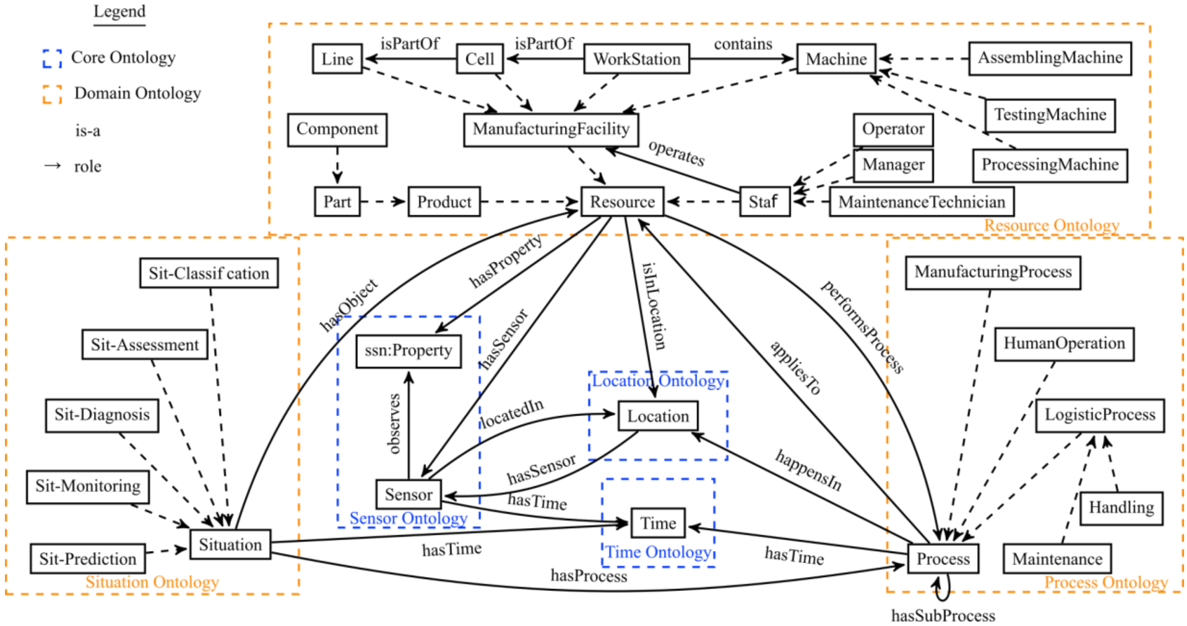


Fig. 1. Context ontology (Cited from Fig. 1 in [4]).

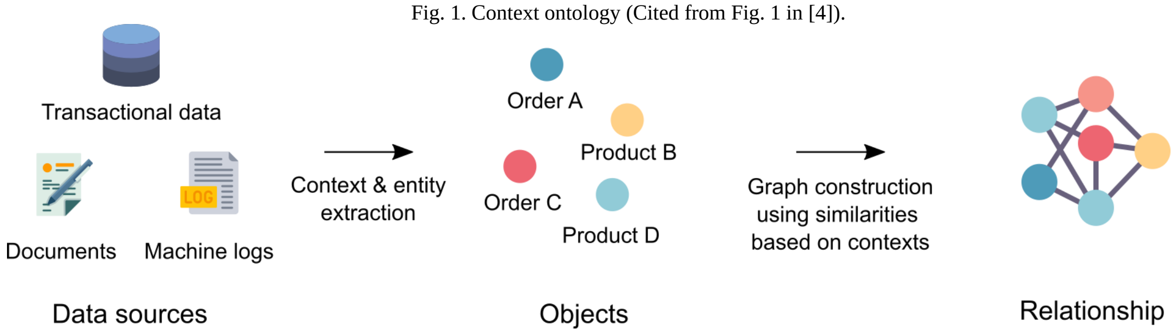


Fig. 2. Overview of the proposed dynamic object relationship modeling.

B. Dynamic Graph

Modeling relationships with graph theory is another related approach (e.g., [12]-[16]). Harary and Gupta extended the approaches designed for static graph theory to dynamic graphs, i.e., graphs that change over time [12]. Casteigts *et al.* proposed a unified dynamic graph framework called time-varying graphs for dynamic networks, ranging from biology to transportation networks [13]. Latapy *et al.* defined the set of concepts for temporal and structural nature of interactions, e.g., density, clusters, cliques, degrees, and clustering coefficients [14]. Rossetti and Cazabet summarized community discovery approaches for dynamic networks [15].

C. Process Mining

Another related work includes process mining (e.g., [17], [18]), which discovers and analyzes a process flow from event logs output by systems such as ERP (Enterprise Resource Planning) and machines. Process mining is related to our method in the sense that it also tries to capture the relationships of "processes" from a given event log, whereas our objective is to capture the relationships of "objects." The mainstream of this research is process discovery algorithms that output a structured process flow that well explains given event logs and any possible unseen events (e.g., [19]-[21]). Once a process flow is obtained, it can be further analyzed for many objectives such as to check if there exists anomaly process flows in a given log, to execute a bottleneck analysis, and to analyze resources assigned to discovered processes [17]. For example, Lorenz et al. applied a process mining technique to the processes of sanitary products at a leading manufacturing company and identified some bottlenecks that can improve productivity [22].

III. Problem Statement

As we have seen in the previous section, none of the existing work focused on dynamic object relationships that appear in the data captured by underlying systems. Ontology is a great modeling toolset; however, it is dedicated to static relationships, and the knowledge of relationships must be given by experts. We are interested in a method that can capture the relationships between objects that dynamically change by their contexts. Likewise, process mining is more focused on processes rather than objects. In this work, we are more interested in the relationships between objects that appear in data sources such as an event log. Our method can be seen as pre-processing for dynamic graph modeling. It works as a bridge between captured row data and graphs which are ready to be analyzed with the dynamic graph theory.

"0001"	XX Toa Payoh	"310001"	2020-12-01 06:07:08	NA
"0002"	YY Leonie Hill Rd	"239191"	2020-12-03 11:31:49	Fragile
"0003"	ZZ Telok Ayer St	"069403"	2020-12-07 14:21:28	Fragile
…	…	…	…	…

↓

code	**location**	**location**	**datetime**	**description**

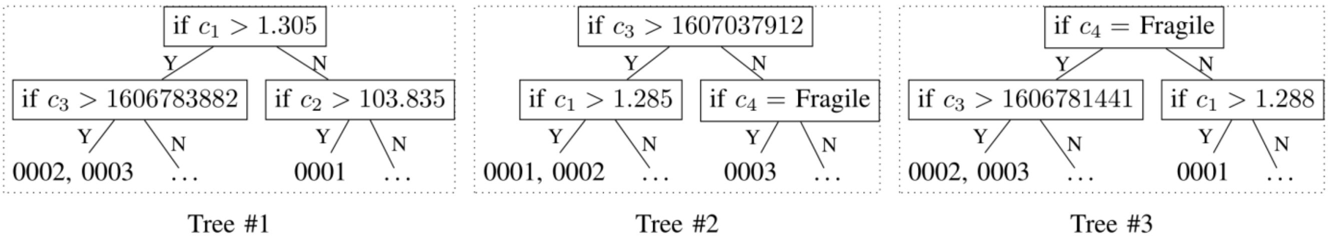
↓

Entity	c_1 [number]	c_2 [number]	c_3 [number]	c_4 [category]
"0001"	1.3321140	103.8475266	1606774028	NA
"0002"	1.2963658	103.8333449	1606966309	Fragile
"0003"	1.2785471	103.8467125	1607322088	Fragile
…	…	…	…	…

Fig. 3. Example of context extraction and conversion. **Middle**: The third column should be identified as location as it is zip-code. **Bottom**: The address column is converted into latitude and longitude and the datetime column is in the Unix time format. The zip-code column is ignored as it is the second location column.

Entity	c_1 [number]	c_2 [number]	c_3 [number]	c_4 [category]
"0001"	1.3321140	103.8475266	1606774028	NA
"0002"	1.2963658	103.8333449	1606966309	Fragile
"0003"	1.2785471	103.8467125	1607322088	Fragile
…	…	…	…	…

↓



↓

$$\mathrm{sim}_{0001,0002} = 1/3,\ \mathrm{sim}_{0002,0003} = 2/3,\ \mathrm{sim}_{0003,0001} = 0$$

Fig. 4. Example of similarity measure with decision trees.

IV. Proposed Method

We propose a method to handle dynamic relationships extracted from heterogeneous data sources. Fig. 2 illustrates the proposed dynamic object relationship modeling. The idea is to quantify the dynamic similarities between objects with contextual information and to construct a weighted graph by using similarities as weights to visualize the relationships between objects. Our method consists of the following procedures.

1) Obtain data sources (e.g. machine logs, documents, transactional data) that contain contextual information
2) Detect contexts from data sources
3) Streamline the data sources based on the extracted contexts
4) Calculate the similarities of objects
5) Analyze the relationships with clustering or graph mining approaches

Although we basically handle table data in this paper, we will also handle unstructured data (e.g., documents) in future. In the following, we describe steps 2 and 4 in detail.

A. Context Detection with Supervised Machine Learning

Context is an important aspect of object modeling [23]. Context-aware approaches have been studied more than two decades in the domain of context-aware computing [24]. Implicit situational information, or context, is intractable yet important for computers to understand what humans need. We borrow the definition of the contexts used in this paper as follows:

Definition 1. *Context (Dey and Abowd [24]): Context is any information that can be used to characterize the* situation *of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*

In [24], four context types, namely location, identity, activity, and time, are defined. We use these four contexts but reword "activity" as "description" since it is more natural in our context. Some data sources may not involve complete column headers or even do not include a header at all. Hence, we must detect an entity and contexts in the given data sources. The idea of our context detection is to capture the characteristics of contexts from the given data sources. For example, datetime has a unique characteristic in its data format (i.e., the number of digits, hyphens, and colons). Hence, our idea is to extract statistical features (e.g., the count of alphabet, digits, and symbols) and to learn its characteristics with a supervised machine learning classifier.

Fig. 3 illustrates an example of context detection against a data source. We assume that five elements in this figure are given as a data source, and the objective is to identify the contexts of each element. We calculate the following two basic features, namely number of characters and number of words. We also calculate the frequency of the following characters as features:

- uppercase letters (`[A-Z]`) (one feature);
- lowercase letters (`[a-z]`) (one feature);
- digits (`[0-9]`) (one feature); and
- symbols (␣, ., “,”, -, _, =, +, |, %, @, #, ^, *, (,), &, ‘, “, $, :, ;, /, [,], {, }, <, >, !, ?, ~, `) (32 features)

In total, we extract 37 features. For instance, the features of “2020-12-01 06:07:08” in the above example are (19, 6, 0, 0, 0.737, 0.053, 0, 0, 0.105, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.105, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0). We use a supervised machine learning algorithm to train a classifier with these features and output labels (i.e., four contexts + “others”). The output labels are (i) entity, (ii) datetime, (iii) location, (iv) description, and (v) others.

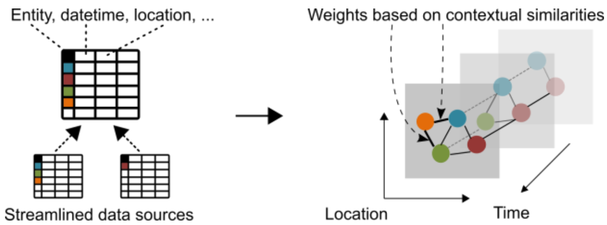


Fig. 5. Concept of dynamic relationships based on contexts.

TABLE I: PRE-PROCESSED DATASET

CrimeCode	DateTime	Latitude	Longitude	Description	I/O	Weapon	Total Incidents
3CF	1478919360	-76.59288	39.28242	ROBBERY - COMMERCIAL	I	FIREARM	1
4E	1478919600	-76.56709	39.37070	COMMON ASSAULT	I	HANDS	1
3CO	1478922300	-76.64455	39.28624	ROBBERY - COMMERCIAL	O	OTHER	1
4E	1478937600	-76.63842	39.30340	COMMON ASSAULT	I	HANDS	1
4E	1478943600	-76.67216	39.30587	COMMON ASSAULT	I	HANDS	1
4E	1478943600	-76.61042	39.29883	COMMON ASSAULT	I	HANDS	1
4E	1478948400	-76.53906	39.32120	COMMON ASSAULT	I	HANDS	1
4B	1478950200	-76.64888	39.28810	AGG. ASSAULT	O	KNIFE	1
3AF	1478910600	-76.64539	39.31249	ROBBERY - STREET	O	FIREARM	1

B. Context-Aware Similarity Measure

After applying our context detection method to multiple data sources, we can streamline them by the contexts. If more than one same context is found in the data, then we only use the leftmost one. Fig. 4 illustrates the concept of dynamic relationships based on the contexts. As shown in this figure, we can map objects onto a time-location space, visualizing relationships among objects. For this, we transform time and locational contexts into numbers. Specifically, when an element is detected as address or zip-code, then we use a web API (e.g., Google's Geocoding API[1]) to convert them into geographic coordinates (i.e., a pair of latitude and longitude), enabling them to be treated as numbers. Similarly, when an element is detected as datetime, then we convert it into Unix epoch, which is the number of seconds that have elapsed since January 1, 1970.

The remaining task is to quantify the strength of the relationships. For this, we calculate similarity scores with contexts. However, as each object is represented as a vector of numeric, categorical, and character values, classical similarity measures such as Euclidean distance and cosine similarity cannot handle such heterogeneous data well. Hence, we leverage unsupervised decision trees for a similarity measure to handle such cases [7], [9]. Fig. 5 illustrates how similarities between objects (i.e., 0001, 0002, and 0003) which are represented as four features, c_1, c_2, c_3, and c_4, are calculated with three decision trees. We first generate decision trees by randomly generating splitting criteria by considering each feature’s value range. The depth of each tree and the number of trees are configurable. After generating decision trees, each entity is fed into the trees and records which terminal nodes they reach. If the entities fell into the same terminal node, then their similarity count is incremented by one. Finally, a similarity between objects i and j are calculated as

$$\mathrm{sim}_{i,j} = N_{i,j} / N_T,$$

where $N_{i,j}$ and N_T denote the number of trees where objects i and j fall in the same terminal node and number of trees, respectively. In our example, objects 0002 and 0003 reach the same terminal nodes of the tree #1 and #3, and thus their similarity is 2/3 = 0.67. Similarly, the similarities between 0001 and 0002 and between 0003 and 0001 are 1 / 3 = 0.33 and 0 / 3 = 0, respectively.

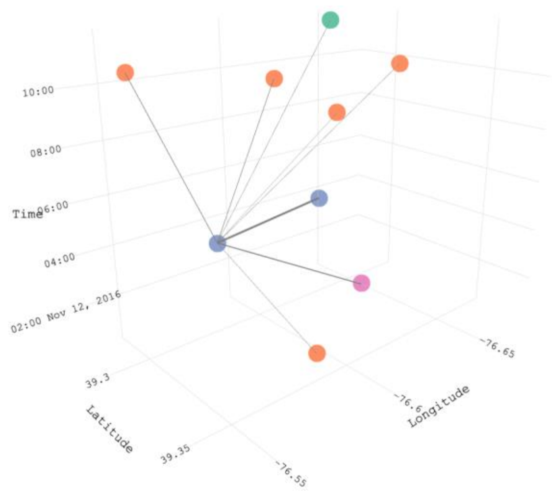
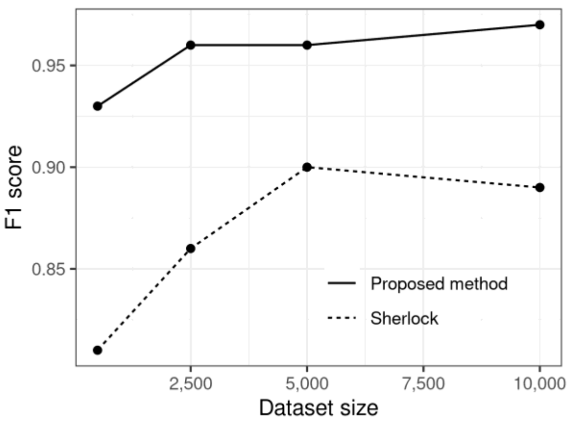


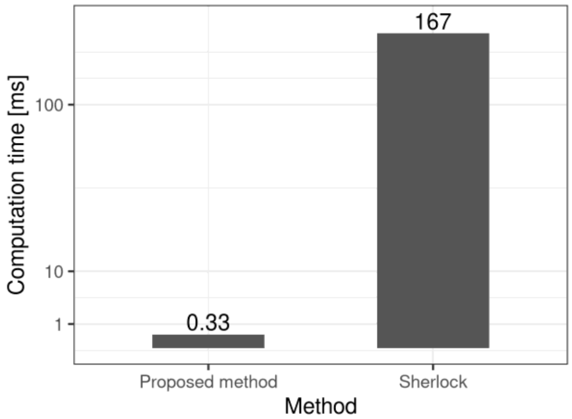
Fig. 6. Example of context-aware object relationships. The colors of objects are based on the values of description. The weights of edges are similarities calculated with unsupervised random forest.

[1] https://developers.google.com/maps/documentation/geocoding

Once we obtain a similarity matrix, we can cluster objects with clustering algorithms such as *k*-medoids clustering (e.g., [25]) or construct a weighted dynamic graph with similarities for further analysis (e.g., [14], [26]).



(a) F_1 score versus dataset size.



(b) Computation time (feature selection).

Fig. 7. Comparison of context detection methods. Both methods use random forest as a classification algorithm and the number of trees is set to 250.

C. Example of Context-Aware Object Relationships

For better understanding of our concept, we show an example of context-aware dynamic object relationships with an openly available dataset. For simplicity's sake, we used a single dataset, crime data in Baltimore[2] as it contains rich contextual information (i.e., datetime, location, description, and other attributes) and pre-processed it beforehand as shown in Table 1. Fig. 6 shows ten crimes (objects) in the dataset and the relationships between a crime (indicated in orange) and others. We calculated similarities between objects in the table with the values shown in the table, and they are used as the weights of edges between objects. A similarity measure with unsupervised random forest can handle heterogeneous value types as in Table I without explicit type conversion.

V. Performance Evaluation

We compare our context detection method with the state-of-the-art semantic detection method called Sherlock [10] in terms of detection accuracy and speed. Sherlock was chosen as it can be used to detect contexts. It calculates four categories of features (e.g., character distribution and pre-trained word embedding) which add up to 1,587 features. The evaluated performance metrics are F_1 score, an overall accuracy measure calculated with recall r and precision p as $F_1 = 2rp / (r + p)$, and the computation time of feature extraction. We vary the number of samples to clarify the relationship between dataset size and accuracy. Each data is labeled as one of the five classes, namely (i) entity, (ii) datetime, (iii) location, (iv) description, and (v) others. When dataset size is 5,000, it means that we uniformly sample 1,000 cases per class from an entire dataset for training and testing.

For both methods, we used the random forest as a supervised machine learning classifier [27] and T2Dv2 Gold Standard[3] as a labeled dataset, respectively. The number of trees in the random forest is set to 250, and a F_1 score was evaluated with 5-fold cross validation. We repeat the same trial 100 times and calculate an average F_1 score and computation time. We conducted performance evaluation on a workstation equipped with 16 CPUs and 128 GiB RAM.

Fig. 7 shows (a) F_1 score versus dataset size and (b) the computation time of feature extraction. As can be seen from the figures, our method outperforms the state-of-the-art method in terms of accuracy and speed. We can see that a F_1 score improves by 6-12% while reducing computation time significantly. This is mainly because we calculated only 37 features in contrast to Sherlock's 1,587 features. In addition, we vectorized operations for feature extraction, which might contribute to fast calculation.

VI. Conclusion

In this paper, we have proposed a concept of dynamic relationship extraction. The heart of this paper consists of context detection and a context-aware similarity measure. We have shown with an open dataset that our context detection method is fast and accurate.

However, this work is still in a very early stage. The open questions include (i) how to better handle the case when a given data contains the multiple same contexts which we currently only use the firstly appeared one, (ii) how to incorporate implicit contexts (e.g. location inside a factory which may not be given as explicit representation like address), (iii) how to extend our method to deal with unstructured data, (iv) how to quantitatively measure the goodness of the context-aware similarity measure, and (v) how to leverage dynamic relationships for useful analysis.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

All authors conducted the research, analyzed the data, wrote the paper, and had approved the final version.

References

[1] I. G. Terrizzano, P. M. Schwarz, M. Roth, and J. E. Colino, "Data wrangling: The challenging journey from the wild to the lake.," 2015.

[2] https://data.world/data-society/city-of-baltimore-crime-data

[3] http://webdatacommons.org/webtables/goldstandardV2.html

[2] F. Nargesian, E. Zhu, R. J. Miller, K. Q. Pu, and P. C. Arocena, "Data lake management: Challenges and opportunities," in *Proc. of the Very Large Data Base (VLDB) Endowment*, vol. 12, pp. 1986–1989, 12 2019.

[3] D. M. Giménez, M. Vegetti, H. P. Leone, and G. P. Henning, "PRoduct ONTOlogy: Defining product-related concepts for logistics planning activities," *Computers in Industry*, vol. 59, pp. 231–241, Mar. 2008.

[4] F. Giustozzi, J. Saunier, and C. Z. Merk, "Context modeling for Industry 4.0: An ontology-based proposal," *Procedia Computer Science*, vol. 126, pp. 675–684, 2018.

[5] D. L. McGuinness, F. V. Harmelen *et al.*, "OWL web ontology language overview," *W3C Recommendation*, vol. 10, no. 10, p. 2004, 2004.

[6] Horrocks, P. F. P. Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean *et al.*, "SWRL: A semantic web rule language combining OWL and RuleML," *W3C Member Submission*, vol. 21, no. 79, pp. 1–31, 2004.

[7] L. Breiman and A. Cutler, Random forests manual v4, 2003.

[8] K. Toyoda, M. Park, N. Okazaki, and T. Ohtsuki, "Novel unsupervised SPITters detection scheme by automatically solving unbalanced situation," *IEEE Access*, vol. 5, no. 1, pp. 6746–6756, 2017, ISSN: 2169-3536.

[9] K. Dalleau, M. Couceiro, and M. S. Tabbone, "Unsupervised extra trees: A stochastic approach to compute similarities in heterogeneous data," *International Journal of Data Science and Analytics*, vol. 9, no. 4, pp. 1–13, 2020.

[10] M. Hulsebos, K. Hu, M. Bakker *et al.*, "Sherlock: A deep learning approach to semantic data type detection," in *Proc. of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1500–1508.

[11] R. Sudarsan, S. J. Fenves, R. D. Sriram, and F. Wang, "A product information modeling framework for product lifecycle management," *Computer-Aided Design*, vol. 37, no. 13, pp. 1399–1411, 2005.

[12] F Harary and G Gupta, "Dynamic graph models," *Mathematical and Computer Modelling*, vol. 25, no. 7, pp. 79–87, 1997.

[13] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, "Time-varying graphs and dynamic networks," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 27, no. 5, pp. 387–408, 2012.

[14] M. Latapy, T. Viard, and C. Magnien, "Stream graphs and link streams for the modeling of interactions over time," *Social Network Analysis and Mining*, vol. 8, no. 1, pp. 1–29, 2018.

[15] G. Rossetti and R. Cazabet, "Community discovery in dynamic networks: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, pp. 1–37, 2018.

[16] G. B. Mertzios, H. Molter, and V. Zamaraev, "Sliding window temporal graph coloring," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7667–7674.

[17] W. Van Der Aalst, "Process mining," *Communications of the ACM*, vol. 55, no. 8, pp. 76–83, 2012.

[18] S. J. van Zelst, F. Mannhardt, M. de Leoni, and A. Koschmider, "Event abstraction in process mining: Literature review and taxonomy," *Granular Computing*, vol. 6, no. 3, pp. 719–736, 2021.

[19] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering block-structured process models from event logs-a constructive approach," in *Proc. International Conference on Applications and Theory of Petri Nets and Concurrency*, Springer, 2013, pp. 311–329.

[20] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, and A. Polyvyanyy, "Split miner: Automated discovery of accurate and simple business process models from event logs," *Knowledge and Information Systems*, vol. 59, no. 2, pp. 251–284, 2019.

[21] C. Cabanillas, L. Ackermann, S. Schönig, C. Sturm, and J. Mendling, "The RALph miner for automated discovery and verification of resource-aware process models," *Software and Systems Modeling*, vol. 19, no. 6, pp. 1415–1441, 2020.

[22] R. Lorenz, J. Senoner, W. Sihn, and T. Netland, "Using process mining to improve productivity in make-to-stock manufacturing," *International Journal of Production Research*, vol. 59, no. 16, pp. 4869–4880, 2021.

[23] J. Aguilar, M. Jerez, and T. Rodríguez, "CAMeOnto: Context awareness meta ontology modeling," *Applied Computing and Informatics*, vol. 14, no. 2, pp. 202–213, 2018, ISSN: 2210-8327.

[24] A. K. Dey and G. D. Abowd, *Towards A Better Understanding of Context and Context-Awareness*, Springer, Berlin, Heidelberg, 1999.

[25] L. Kaufman and P. J. Rousseeuw, Finding groups in data: An introduction to cluster analysis. John Wiley & Sons, 2009, vol. 344.

[26] D. B. West *et al.*, *Introduction to Graph Theory*, Prentice Hall Upper Saddle River, 2001, vol. 2.

[27] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

Kentaroh Toyoda was born in Tokyo, Japan in 1988. He received his B.E., M.E. and the Ph.D. (engineering) degrees in the Department of information and computer science, Keio University, Yokohama, Japan, in 2011, 2013, and 2016, respectively. He was an assistant professor at Keio University from Apr. 2016 to Mar. 2019, and is currently a scientist at the Singapore Institute of Manufacturing Technology, A*STAR, Singapore. His research interests include blockchain, security and privacy, data analysis, and mechanism design.

Rachel Gan Kai Ying received a B.E. (engineering) in information systems technology and design from Singapore University of Technology and Design, Singapore in 2019. She is currently a research engineer at the Singapore Institute of Manufacturing Technology, A*STAR, working on cyber-physical production system and logistics management research. Her research interests include data analytics, machine learning and their applications in manufacturing and supply chain domains.

Tan Puay Siew received the Ph.D. degree in computer science from the School of Computer Engineering, Nanyang Technological University, Singapore. She is presently an adjunct associate professor with the School of Computer Science and Engineering, Nanyang Technological University and also the co-director of the SIMTECH-NTU Joint Laboratory on Complex Systems. In her full-time job at Singapore Institute of Manufacturing Technology (SIMTech), she leads the Manufacturing Control Tower™ (MCT™) as the Programme Manager. She is also the deputy division director of the Manufacturing System Division. Her research interests are in the cross-field disciplines of Computer Science and Operations Research for virtual enterprise collaboration, in particular sustainable complex manufacturing and supply chain operations in the era of Industry 4.0.

Allan NengSheng Zhang is a senior scientist with the Singapore Institute of Manufacturing Technology, A*STAR, Singapore. He has more than 25 years of experiences in the development of operations technologies (OT) using AI such as knowledge-based systems and enterprise information systems development. His research interests include knowledge management, data mining, machine learning, artificial intelligence, computer security, software engineering, software development methodology and standards, and enterprise information systems. He and his group are currently working toward research in cyber-physical system modelling, manufacturing system analyses including data mining, supply chain information management, supply chain risk management using a complex systems approach, multi-objective vehicle routing problems, and urban last-mile logistics.