Learning Systems Engineering Domain Ontologies from Text Documents

Lan Yang
College of Engineering and Informatics
National University of Ireland, Galway
Galway, Ireland
https://orcid.org/0000-0002-7905-1438

Kathryn Cormican
College of Engineering and Informatics
National University of Ireland, Galway
Galway, Ireland
https://orcid.org/0000-0003-1688-1087

Ming Yu
Department of Industrial Engineering
Tsinghua University
Beijing, China
https://orcid.org/0000-0002-3980-7181

***Abstract*—Ontologies are playing an increasingly important role in knowledge management, and their functions have been appreciated and exploited by a broad range of communities, including systems engineering researchers and practitioners. Encompassing domain-related vocabularies, concepts, concept hierarchy, along with the properties and relationships, domain ontologies are becoming a promising medium for knowledge sharing and exchange. With the emergence of the semantic web and big data, learning domain ontologies from text is becoming a cutting-edge technique as it is an automatic process of deriving ontological knowledge. Specifically, a set of representative concepts and semantic relations can be rapidly derived from unstructured text documents in a hierarchical structure to model a domain. In this paper, we aim at exploiting the ontology learning approach to extract a domain ontology from systems engineering handbooks. An approach is proposed for learning terms, concepts, taxonomic and non-taxonomic relations. By incorporating both linguistic-based and statistical-based natural language processing techniques, we realized an automatic detection of complex domain terms and conceptualized the systems engineering body of knowledge in a semantic fashion. To evaluate the proposed approach, a case study is conducted, wherein the hybrid approach is applied with template-driven and machine learning algorithms. The result shows that the proposed approach has a robust performance in decreasing ontology development costs. This paper contributes to a good starting point for learning systems engineering ontologies to enhance knowledge acquisition and management.**

Keywords—ontology learning, natural language processing, systems engineering

I. INTRODUCTION

Systems engineering is an interdisciplinary approach to enable the success of the full life cycle of complex systems [1]. Nowadays, not only has it been recognized by traditional industries, but also accredited by a variety of socio-technical systems, such as medical systems, transportation systems, space and communications systems, as well as gaming and entertainment systems. Systems engineering approach continuously provides adequate support from system concept exploration through system disposal and holistically deals with systemic problems. Systems engineering covers an extensive range of knowledge, whose application fields are also multidisciplinary and complex.

Although the best practice is documented in the systems engineering guidelines and standards, the systems engineering body of knowledge is still primarily treated as heuristics learned by each practitioner [2]. Furthermore, systems engineering comprises various domain specialists, who participate in different kinds of activities, such as design, engineering analysis, manufacture and performance evaluation, resulting in the misunderstanding and miscommunication of domain-specific knowledge.

Vast amounts of data coming from heterogeneous sources need to be simultaneously retrieved and exchanged in a systems engineering project. However, the rapid change in the systems engineering domain leads to the creation of new terms all the time, making knowledge retrieval and management a challenge.

In order to improve the level of formality and consistency, ontologies are being used to conceptualize systems engineering knowledge. So far, ontology development for systems engineering knowledge is primarily concerned with the definition of domain concepts and relations between them [3]. However, knowledge about the symbols that are used to refer to the concepts is also essential. The requirement implies the acquisition of linguistic terms that are used to refer to a specific concept and possible synonyms of these terms [4].

Therefore, the research aims to improve the ontology development approach in the systems engineering domain by exploring the feasibility of learning domain ontologies from text documents. A domain corpus is taken as inputs, from which an ontology can be extracted to conceptualize and specify the implicit knowledge contained in the context. The rationale for creating the domain ontology is that common terminologies can be identified, the logical relations between essential concepts can be well-defined, and knowledge coming from different sources can be integrated and interoperable.

This paper contributes to the following perspectives. First, an ontology learning approach is proposed for aiding the automated generation of systems engineering domain ontologies from text. The methods and techniques used in the learning process follow the best practice of the state of the art of ontology learning. Second, as an intermediate product, a systems engineering domain corpus is created with part-of-speech tags. Third, the outcomes of the semi-automated generation are compared with manually constructed ones. The results denote that the auto-generated ontology has excellent precision and recall in concept extraction, which opens a new chapter of developing systems engineering ontologies.

The remainder of this paper proceeds as follows. Section II reviews the general ontology learning process and state-of-the-art research on ontologies for systems engineering. Section III presents the research methodology. Section IV details the proposed ontology learning approach. Section V reports a case study of applying the proposed approach to the INCOSE systems engineering handbook. Section VI concludes the paper.

Funded by National Key R&D Program of China (2017YFF0209400)

978-1-7281-1783-6/19/$31.00 ©2019 IEEE

II. LITERATURE REVIEW

In recent years, much research has focused on applying ontology engineering methodologies to systems engineering discipline [5]–[7]. The trend is visible through an increasing number of articles, special issues of journals or dedicated sessions in international conferences. Ernadote [8] presents an approach to support model-based systems engineering by combining the advantage of standard meta-models such as UML (unified modeling language) and SysML (systems modeling language) with dedicated project ontologies. van Ruijven [9] develops an ontology for interpreting the ISO/IEC/IEEE 15288 standard [10]. Yang et al. [11] develop a formal systems engineering ontology in OWL (web ontology language) based on the input-process-output diagrams defined in the INCOSE systems engineering handbook [12]. Ontologies not only make the structure of engineered systems and their components explicit but also help different stakeholders better understand the inherent complexities of large engineered systems and their socio-technical environments [13]. However, Dori [14] indicates that systems science and engineering are still "in need of a well-defined foundational, universal, general, necessary and sufficient ontology that would underpin concepts and terms it uses in order for them to be precise and unambiguous".

In fact, according to Honour and Valerdi [2], systems engineering is still treated primarily as heuristics learned by each practitioner and the heuristics known by each practitioner slightly differ from each other, as can be seen by the fractured development of systems engineering standards and certification. Therefore, more and more researchers are calling for a shared conceptualization of systems engineering [15], which are commonly known as a systems engineering domain ontology. An ontology is an explicit specification of a shared conceptualization of concepts and relations for formal knowledge representation in a particular domain of interest [16]–[20]. Manual ontology construction relying on domain experts is a time-consuming, error-prone, costly, difficult, and tedious task [21]–[23]. To the best of our knowledge, the existing systems engineering ontologies are all constructed manually, thus suffer from the above knowledge acquisition bottleneck. Therefore, learning ontologies from natural language text documents can be a new angle to solve the problem.

Ontology learning has been given different definitions in different research. The variation is due to the distinction of research purpose, the depth and granularity of the process, and the expectation of the final outcomes. We reviewed the state-of-the-art literature in ontology learning process and methodologies and compared the activities involved in. However, none of them are fully in line with the scope of our research. Therefore, we propose a definition of ontology learning for our research based on relevant literature [24]–[29]: ontology learning is a process of extracting and identifying conceptual knowledge in a domain (concepts, relations, and axioms) from textual information sources of structured, semi-structured or unstructured types, that using a set of methods and techniques for constructing an ontology from scratch or for enriching or adapting an existing ontology, in an automatic or semi-automatic fashion.

In the following section, we will discuss the architecture of the ontology learning process.

III. RESEARCH METHODOLOGY

In order to extract domain ontologies for systems engineering body of knowledge, we designed a 4-phase research methodology to ensure the rigor and integrity of this study. Following this methodology step by step, we can obtain the phased outcomes after the completion of every phase. The research methodology is illustrated in Fig. 1.

Phase 1 is the pre-processing phase. In this phase, we conducted documents collection at the beginning and obtained the available document resources in PDF format. Then, we converted the files from the PDF format to TXT format to allow later text processing. Also, structure analysis of the documents is essential in this early stage since we will be able to control the content for later phases. Also, we conducted a cleaning on the text to obtain a plain text without garbled characters. The output of this stage is a domain corpus that allows further parsing and manipulating.

Phase 2 is the core stage for constructing the domain ontology. We referred to the best-known ontology learning layer cake [30] to design the activities to obtain the elements for an ontology. This phase contains eight activities according to the eight layers of the composition of "the cake". After the completion of each activity in this phase, we can gradually assemble the elements of an ontology, such as domain vocabularies, concepts, taxonomic and non-taxonomic relations.

Phase 3 is the process to build the ontology by populating the instances into the ontology model. As we received an initial ontology model from Phase 2, we can enrich the model by increasing the depth of the concept hierarchy and adding new non-taxonomic relations as well as extracting more complicated concepts and instances. The outcome of this phase is a visualized domain ontology consisting of systems engineering domain concepts and relations.

The last phase (Phase 4) is to evaluate the ontology generated from the former three phases. We followed the gold standard based evaluation method proposed by Zavitsanos et al. [31] and calculated the precision and recall in a case study.

In the next section, we will mainly focus on describing the activities and methods of Phase 2 as it is the critical stage of obtaining the domain ontology. In Section V, we will demonstrate the approach by presenting a case study in the INCOSE systems engineering handbook and report the results of the evaluation.

IV. PROPOSED APPROACH

In this section, we will explain the core of the proposed approach to learn domain ontologies from systems engineering text documents. Due to the limited space of this paper, we will omit the specific activities in the pre-processing phase and begin with the hypothesis that we have obtained a systems engineering domain corpus. Fig. 2 shows the architecture of the learning process for deriving the ontology, with a map between the adopted methods and the deliverables or outputs.

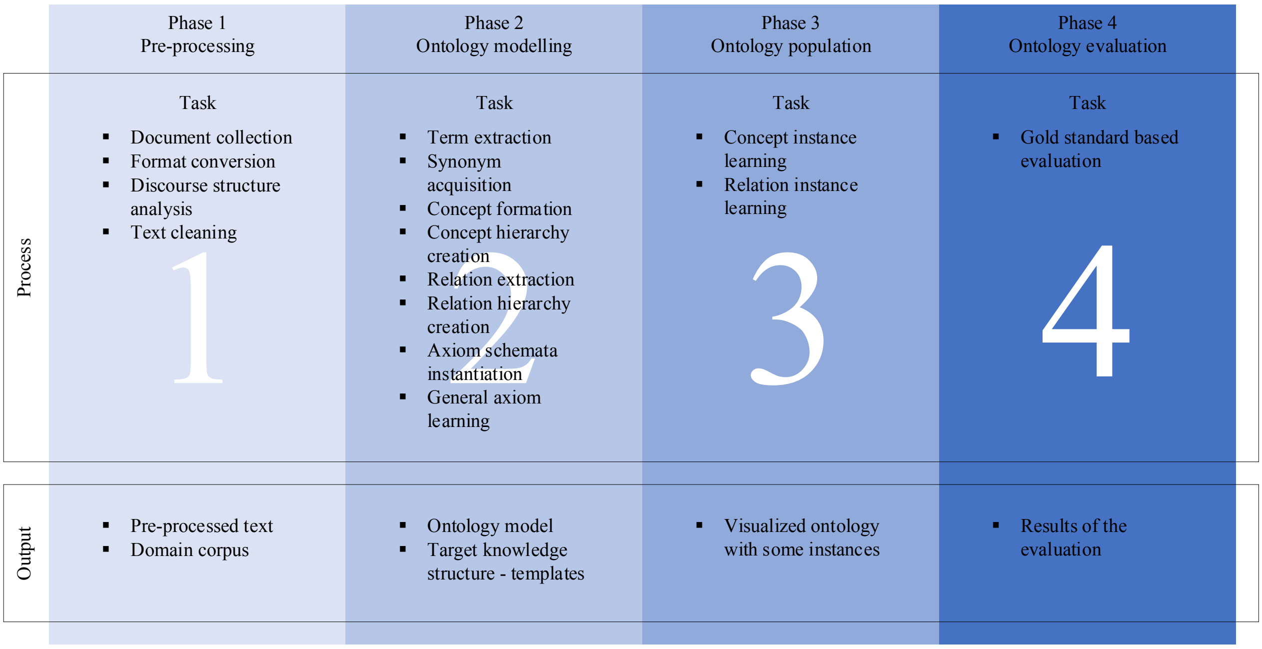

Fig. 1. Research Methodology

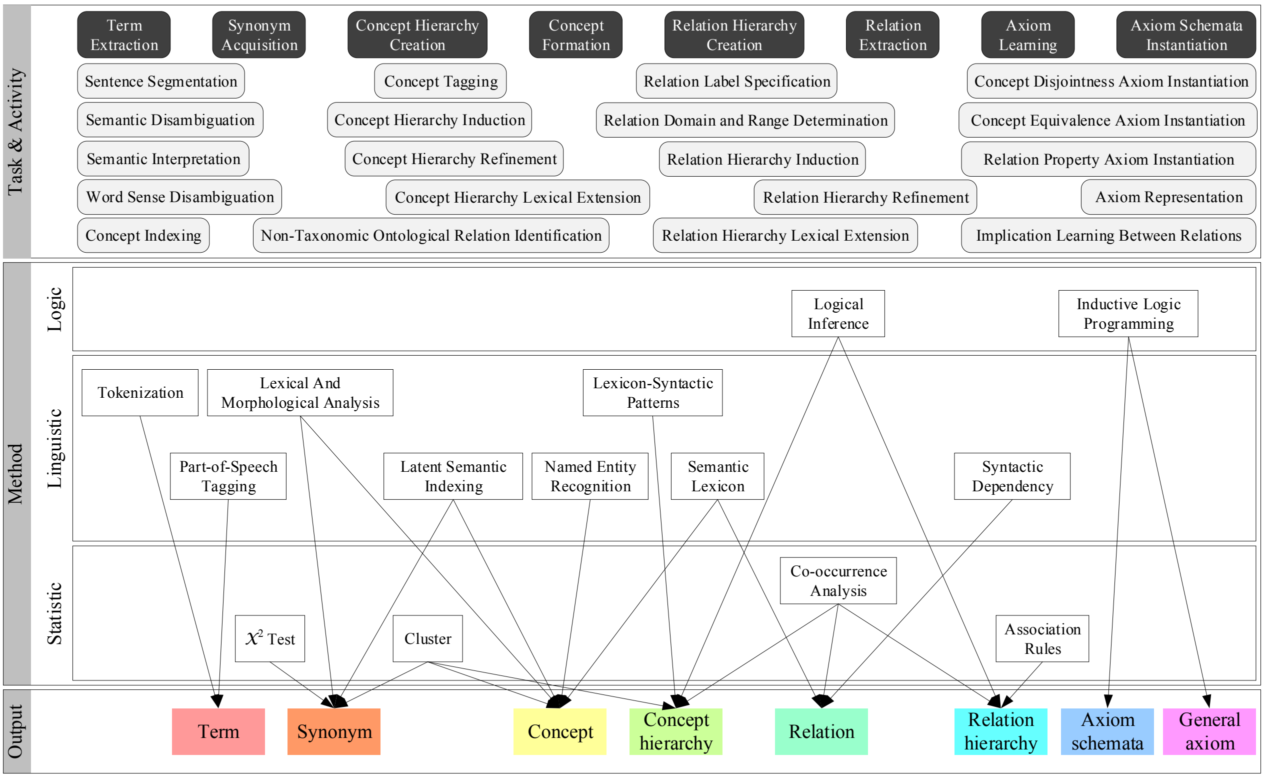

Fig. 2. The architecture of the proposed approach

Eight tasks are defined for learning domain ontologies according to the constitution of an ontology, i.e. term, synonym, concept, concept hierarchy, relation, relation hierarchy, axiom and general axiom. These elements are corresponding to the so-called ontology learning layer cake. A schematic diagram is shown in Fig. 3. From the bottom (Term) to the top (General Axiom), the abstract degree progressively arises, and the complexity of the learning process gradually increases [32].

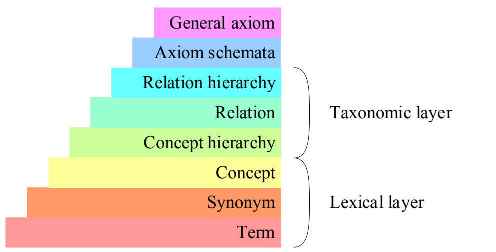

Fig. 3. Ontology learning layer cake (adapted from [30])

So far, much research has proposed methods to support the lower layers, i.e. the lexical layer and taxonomic layer. The lexical layer consists of term extraction, synonym acquisition and concept formation. Taxonomic layer focuses on the creation of the concept hierarchy (taxonomic relation extraction) and the relation hierarchy (non-taxonomic relation extraction).

In the following sub-sections, we will describe five critical tasks during the extraction of systems engineering domain ontology. As shown in Fig. 2, we categorized the methods into three types, statistic methods, linguistic methods and logic methods, according to Liu et al. [33]. For each task, we will detail what activities are conducted and what methods are considered.

A. *Term Extraction*

The first task for analyzing the domain corpus is to extract domain vocabularies, i.e. term extraction. This process uses linguistic natural language processing methods such as tokenization and part-of-speech tagging to make sentence segmentation. The tokenization activity will turn the text into word tokens, while the part-of-speech tagging method will allow chunking words tokens into semantically meaningful phrases and parsing sentences. We can also conduct fundamental linguistic analysis, such as obtaining the most frequent nouns and verbs based on the output of this task.

B. *Synonym Acquisition*

As terms are the smallest unit of semantic expression, there exist different ways to describe the same or similar things. Two terms are synonymous relatively to a context if both terms are syntactically identical and semantically substitutable in the context [34]. Therefore, we need to group synonyms so that similar terms can be clustered to form a concept. There are many cluster algorithms for acquiring synonyms [35]–[37]. Meanwhile, apart from statistic methods, we also adopted lexical and morphological analysis as they can make up the shortcomings of cluster algorithms.

C. *Concept Formation*

The concept is the key element of an ontology, and the cluster of synonyms is an importance prerequisite for forming domain concepts. A series of activities can be carried out to distinguish concepts, such as semantic disambiguation, semantic interpretation and word sense disambiguation. We utilized many linguistic methods since different lexical and morphological analyses can complement each other. As for concept indexing, we used latent semantic indexing proposed by Deerwester et al. [38]. Among various methods, we implemented named entity recognition to locate and classify named entities in text into pre-defined categories.

D. *Taxonomic Relation Extraction*

The taxonomic relation is also considered as 'is a' relation, which presents the hierarchical relations between concepts. In order to obtain this kind of relation, we adopted logical inference methods to conduct logical reasoning. Moreover, we also combined lexicon-syntactic patterns with cluster methods to make concept hierarchy induction and refinement.

E. *Non-taxonomic Relation Extraction*

The non-taxonomic relation extraction includes two tasks, extracting the relation used to link concepts (usually a verb) and the expression of a relation which consists of two concepts and the relation label in between. The method of co-occurrence analysis aimed at non-taxonomic ontological relation identification. The activities related to the relation definition includes relation label specification and domain and range determination. We also used logical inference in relation to hierarchy induction and refinement.

Due to the limit of the paper, we will not detail the ontology population process. However, the visualized ontology will be presented in the next section.

V. CASE STUDY

We applied the proposed approach in a case study that the INCOSE systems engineering handbook was chosen to be the targeted text document. We carried out the pre-processing activities, as described in Section III. The corpus consists of nine chapters of the handbook, from Chapter 2 to Chapter 10. We excluded the contents, history of changes, Chapter 1 and appendices, as we believe they will not contribute much to the knowledge extraction. The following sub-sections will report the results of the case study and the performance of the proposed approach.

A. *Tools*

The corpus is saved in TXT format and edited in Microsoft Notepad. We used Spyder platform for programming in the Python language to process the data. Notably, we employed the text processing libraries of Natural Language Toolkit (NLTK) for tokenization, stemming, tagging, and parsing the raw text. The final ontology is built in Protégé for visualization and semantic reasoning.

B. *Results*

Firstly, we carried out some fundamental linguistic analysis of the raw text. The corpus contains 133,817 words and punctuation symbols, i.e. tokens. Although it has 131,282 tokens, this corpus has only 9,147 distinct word types, which include words and punctuation symbols. The lexical richness of the text is just 7% of the total number of words, or equivalently that each word is used 14 times on average.

Secondly, we use 'stopwords' corpus provided by NLTK to filter out commonly seen and high-frequency words such as 'the', 'to' and 'also', since the stopwords usually have little lexical content. After filtering out the stopwords, we obtained a data set whose size is 89,137. Moreover, we generated a vocabulary for the corpus, which contains 6,881 unique words or word stem. Then, we calculated a frequency distribution of the 30 most frequently appeared word stem (shown in Fig. 4).

Thirdly, we conducted a part-of-speech tagging and chunked the words to phrases. An example of a tagged and chunked sentence is shown in Fig. 5.

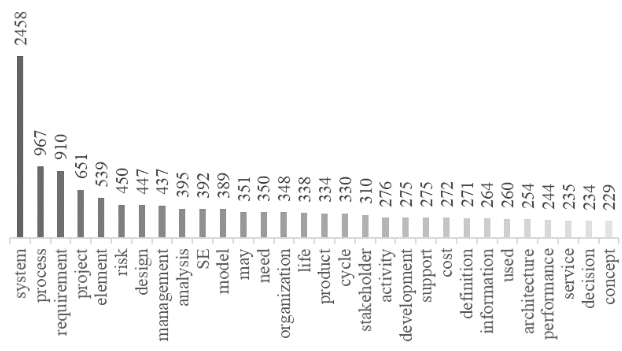

Fig. 4. The 30 most frequently used word stem

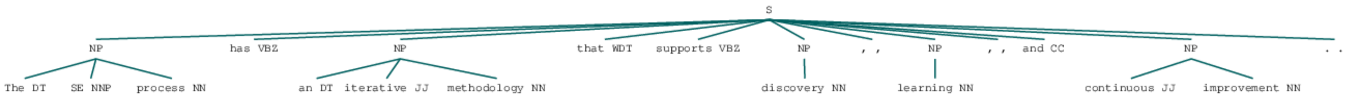

Fig. 5. An example of a tagged and chunked sentence

Therefore, we are able to extract concepts and relations that are frequently used in the corpus. Cshows the 30 most frequently appeared verbs, which are identified as candidates for labelling relations. TABLE II. is an excerpt of the concepts.

Finally, we used Protégé to establish the ontology by populating the extracted concepts and relations into a top-level ontology as described in our previous work [39] for enrichment. Fig. 6 illustrates the network consisting of stakeholder needs and requirements definition process, system requirements definition process, and their related concepts. We can identify that there is a traceable relation. Through stakeholder needs and requirements definition process, stakeholder requirements are generated and input in the system requirements definition process, while the system requirements traceability is returned. Also, we can quickly identify a loop between the system requirements definition process and verification process.

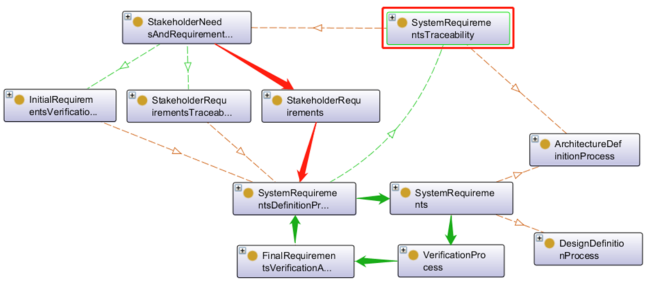

Fig. 6. An excerpt of the systems engineering ontology

TABLE I. THE 30 MOST FREQUENTLY USED VERBS

Item	**Frequency**	**Item**	**Frequency**
is	1554	ensure	129
are	1158	defined	126
be	1158	needed	109
can	467	required	105
may	351	based	98
should	269	using	97
used	259	been	93
must	197	meet	91
will	196	support	91
have	189	see	88
include	164	identify	87
provide	145	identified	83
has	144	provides	81
includes	137	define	79
including	131	developed	78

TABLE II. AN EXCERPT OF THE CONCEPTS

Item	**Item (cont.)**	**Item (cont.)**
architectural characteristics	failure modes	quality management
architectural entities	following activities	requirements definition
architecture definition	hazardous materials	resource management
bidirectional traceability	information management	risk management
black box	IPO diagram	root cause
business case	ISO/IEC/IEEE 15288	security engineering
candidate architectures	ISO/IEC/IEEE 29148	stakeholder concerns
concept stage	knowledge assets	stakeholder needs
configuration management	lean thinking	stakeholder requirements
constituent systems	life cycle	supply chain
corrective action	matter experts	system architecture
customer satisfaction	mission analysis	system elements
decision gate	operational environment	system requirements
decision making	operational scenarios	system safety
design characteristics	passive infrastructure	system security
design definition	preventive maintenance	systems engineer
development stage	primary objective	systems engineering
emergent properties	product line	systems science
enabling systems	project manager	systems thinking
environmental impact	project planning	technical measurement
exploratory research	qualified personnel	validation actions
external interfaces	quality assurance	verification actions

C. Evaluation

We first performed a term level evaluation which relies on the domain experts' concept per concept based evaluation of the ontologies to conclude their usefulness. Then, the domain experts generated corresponding gold standard ontologies. The gold standard ontologies are developed manually based on the knowledge of the experts and the interpretation of the handbook. Finally, we compared the extracted ontologies to the corresponding gold standards to assess their domain coverage. The results of extraction were successfully evaluated in the context of the INCOSE systems engineering handbook, leading to around 80% precision and 55% recall. The recall was estimated by manually identifying truly relevant terms from a list of syntactically plausible multi-word expressions.

VI. CONCLUSION

The state of the art of ontology development in systems engineering is improved by introducing ontology learning methods to derive systems engineering domain ontologies from text documents. A corpus for INCOSE systems engineering handbook is created, and an ontology is learned from the implicit corpus. However, the corpus can be expanded to include more systems engineering domain documents; thus, the domain ontology can be enriched. The proposed ontology learning process is an initial starting point to generate systems engineering domain ontologies. We will continue the study in depth to focus on the taxonomic layer in the future.

ACKNOWLEDGMENT

We sincerely thank David Ward, systems engineering manager at Flex Milan Design Centre for providing support for understanding the metamorphosis of systems engineering standards.

REFERENCES

[1] BKCASE Editorial Board., "The Guide to the Systems Engineering Body of Knowledge (SEBoK)," The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 1.9. R.D. Adcock (EIC). Hoboken, NJ: The Trustees of the Stevens Institute of Technology., 2017. [Online]. Available: www.sebokwiki.org. [Accessed: 22-Mar-2018].

[2] E. C. Honour and R. Valerdi, "Advancing an ontology for systems engineering to allow consistent measurement," 2006.

[3] E. W. Aslaksen, C. Street, and S. Leonards, "Elements of a Systems Engineering Ontology," in Paper presented on 2011 Systems Engineering and Test and Evaluation in the Next Decade Symposium, 2011, pp. 1–18.

[4] C. Brewster, "Ontology Learning from Text: Methods, Evaluation and Applications Paul Buitelaar, Philipp Cimiano, and Bernado Magnini (editors) (DFKI Saarbrücken, University of Karlsruhe, and ITC-irst), Amsterdam: IOS Press (Frontiers in artificial intelligence and appl," Comput. Linguist., vol. 32, no. 4, pp. 569–572, Dec. 2006.

[5] N. Hallberg, E. Jungert, and S. Pilemalm, "Ontology for Systems Development," Int. J. Softw. Eng. Knowl. Eng., vol. 24, no. 03, pp. 329–345, 2014.

[6] M. B. Sarder and S. Ferreira, "Developing systems engineering ontologies," in Conference Proceedings of 2007 IEEE International Conference on System of Systems Engineering, 2007, pp. 1–6.

[7] T. Hoppe, H. Eisenmann, A. Viehl, and O. Bringmann, "Guided systems engineering by profiled ontologies," in 2017 IEEE International Systems Engineering Symposium (ISSE), 2017, pp. 1–6.

[8] D. Ernadote, "Ontology-Based Pattern for System Engineering," in 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS), 2017, pp. 248–258.

[9] L. C. van Ruijven, "Ontology for Systems Engineering as a base for MBSE," INCOSE Int. Symp., vol. 25, no. 1, pp. 250–265, 2015.

[10] ISO/IEC/IEEE 15288:2015, Systems and software engineering -- System life cycle processes. Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions / Institute of Electrical and Electronics Engineers, 2015.

[11] L. Yang, K. Cormican, and M. Yu, "Towards a methodology for systems engineering ontology development — An ontology for system life cycle processes," in 2017 IEEE International Systems Engineering Symposium (ISSE), 2017, pp. 1–7.

[12] INCOSE, International Council on Systems Engineering. Systems engineering handbook: A guide for system life cycle processes and activities, 4th Editio. Hoboken, New Jersey, USA: John Wiley & Sons, Inc., 2015.

[13] V. Mezhuyev, "Ontology based development of Domain Specific Languages for Systems Engineering," 2014 Int. Conf. Comput. Inf. Sci., pp. 1–6, 2014.

[14] D. Dori, Model-Based Systems Engineering with OPM and SysML, 1st ed. New York: Springer-Verlag New York, 2016.

[15] J. Ring, "Toward an Ontology of SE," INSIGHT, vol. 5, no. 1, pp. 19–22, 2002.

[16] M. Alobaidi, K. M. Malik, and M. Hussain, "Automated ontology generation framework powered by linked biomedical ontologies for disease-drug domain," Comput. Methods Programs Biomed., vol. 165, pp. 117–128, Oct. 2018.

[17] A. A. Barforush and A. Rahnama, "Ontology learning: Revisted," J. Web Eng., vol. 11, no. 4, pp. 269–289, 2012.

[18] X. Jiang and A.-H. Tan, "CRCTOL: A semantic-based domain ontology learning system," J. Am. Soc. Inf. Sci. Technol., vol. 61, no. 1, pp. 150–168, Jan. 2010.

[19] M. Rani, A. K. Dhar, and O. P. Vyas, "Semi-automatic terminology ontology learning based on topic modeling," Eng. Appl. Artif. Intell., vol. 63, pp. 108–125, Aug. 2017.

[20] J. M. Ruiz-Martínez, R. Valencia-García, J. T. Fernández-Breis, F. García-Sánchez, and R. Martínez-Béjar, "Ontology learning from biomedical natural language documents using UMLS," Expert Syst. Appl., vol. 38, no. 10, pp. 12365–12378, Sep. 2011.

[21] L. Yiqing, "An Improved Ontology Learning Algorithm from Relational Schema," Rev. Tec. La Fac. Ing. Univ. Del Zulia, vol. 39, no. 7, pp. 63–68, Nov. 2016.

[22] L. Lin, Z. Xu, and Y. Ding, "OWL ontology extraction from relational databases via database reverse engineering," J. Softw., vol. 8, no. 11, pp. 2749–2760, 2013.

[23] B. El Idrissi, S. Baïna, and K. Baïna, "Ontology Learning from Relational Database: How to Label the Relationships Between Concepts?," in Communications in Computer and Information Science, vol. 521, 2015, pp. 235–244.

[24] A. Gómez-Pérez and D. Manzano-Macho, "An overview of methods and tools for ontology learning from texts," Knowl. Eng. Rev., vol. 19, no. 03, pp. 187–212, 2005.

[25] J. Kong, "Ontology Learning for Chinese Information Organization and Knowledge Discovery in Ethnology and Anthropology," Data Sci. J., vol. 6, no. SUPPL., pp. S500–S510, 2007.

[26] E. Simperl, C. Tempich, and D. Vrandečić, A methodology for ontology learning, vol. 167, no. 1. 2008.

[27] H. A. Santoso, S.-C. Haw, and Z. T. Abdul-Mehdi, "Ontology extraction from relational database: Concept hierarchy as background knowledge," Knowledge-Based Syst., vol. 24, no. 3, pp. 457–464, Apr. 2011.

[28] B. E. Idrissi, S. Baina, and K. Baina, "Upgrading the semantics of the relational model for rich OWL 2 ontology learning," J. Theor. Appl. Inf. Technol., vol. 68, no. 1, pp. 138–148, 2014.

[29] F. Colace, M. De Santo, L. Greco, F. Amato, V. Moscato, and A. Picariello, "Terminological ontology learning and population using latent Dirichlet allocation," J. Vis. Lang. Comput., vol. 25, no. 6, pp. 818–826, Dec. 2014.

[30] P. Cimiano, Ontology Learning and Population from Text. Springer US, 2006.

[31] E. Zavitsanos, G. Paliouras, and G. A. Vouros, "Gold Standard Evaluation of Ontology Learning Methods through Ontology Transformation and Alignment," IEEE Trans. Knowl. Data Eng., vol. 23, no. 11, pp. 1635–1648, Nov. 2011.

[32] M. N. Asim, M. Wasim, M. U. G. Khan, W. Mahmood, and H. M. Abbasi, "A survey of ontology learning techniques and applications," Database, vol. 2018, Jan. 2018.

[33] K. Liu, W. R. Hogan, and R. S. Crowley, "Natural Language Processing methods and systems for biomedical ontology learning," J. Biomed. Inform., vol. 44, no. 1, pp. 163–179, Feb. 2011.

[34] T. Hamon and A. Nazarenko, "Detection of synonymy links between terms," in Recent Advances in Computational Terminology, 2001, pp. 185–208.

[35] T. T. Quan, S. C. Hui, and A. C. M. Fong, "Automatic Fuzzy Ontology Generation for Semantic Help-Desk Support," IEEE Trans. Ind. Informatics, vol. 2, no. 3, pp. 155–164, Aug. 2006.

[36] M. Bhatt, C. Wouters, A. Flahive, W. Rahayu, and D. Taniar, "Semantic Completeness in Sub-ontology Extraction Using Distributed Methods," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3045, 2004, pp. 508–517.

[37] J. J. Jung, "An application of collaborative web browsing based on ontology learning from user activities on the web," Comput. Informatics, vol. 23, no. 4, pp. 337–353, 2004.

[38] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," J. Am. Soc. Inf. Sci., vol. 41, no. 6, pp. 391–407, Sep. 1990.

[39] L. Yang, K. Cormican, and M. Yu, "An Ontology Model for Systems Engineering Derived from ISO/IEC/IEEE 15288: 2015: Systems and Software Engineering-System Life Cycle Processes," World Acad. Sci. Eng. Technol. Int. J. Comput. Electr. Autom. Control Inf. Eng., vol. 11, no. 1, pp. 1–7, 2016.