Systems & Control Transactions

Research Article
Foundations of Computer Aided Process Design (FOCAPD 2024)
Breckenridge, Colorado, USA. July 14-18, 2024
Peer Reviewed Conference Proceeding

PSE PRESS

Mining Chemical Process Information from Literature for Generative Process Design: A Perspective

Artur M. Schweidtmann[a,*]

[a] Process Intelligence Research, Delft University of Technology, Department of Chemical Engineering, Delft, The Netherlands
* Corresponding Author: a.schweidtmann@tudelft.nl.

ABSTRACT

Artificial intelligence (AI) and particularly generative AI led to recent breakthroughs, e.g., in generating text and images. There is also a potential of these technologies in chemical engineering, but the lack of structured big domain-relevant data hinders advancements. I envision an open Chemical Engineering Knowledge Graph (ChemEngKG) that provides big open and linked chemical process information. In this article, I present the concept of "flowsheet mining" as the first step towards the ChemEngKG. Flowsheet mining extracts process information from flowsheets and process descriptions found in scientific literature and patents. The proposed technology requires the integration of data mining, computer vision, natural language processing, and semantic web technologies. I present the concept of flowsheet mining, discuss previous literature, and show future potentials. I believe the availability of big data will enable breakthroughs in process design through artificial intelligence.

Keywords: Artificial Intelligence, knowledge graph, data mining, computer vision, natural language processing

INTRODUCTION

The transformation of the chemical process industry to renewable energy and feedstock supply requires the design of highly integrated, flexible, and efficient plants [1]. In the current setting, the development of chemical processes is a challenging task, which is mostly performed by manual simulation or optimization approaches that rely on hierarchical decomposition proposed in the 1980s [2]. There is a need for a paradigm shift that accelerates the development of chemical processes.

Machine learning (ML) and, more generally, artificial intelligence (AI) have great potential for chemical process design but usually require big data [3, 4]. Recent breakthroughs in ML led to success in games, computer vision, healthcare, finance, etc., even surpassing human performance in numerous tasks [5]. This great surge of AI applications often stems from the accessibility of big data, i.e., big in volume, variety, and velocity (cf., discussion on definitions of big data [6]) [7].

While engineers use a variety of data and knowledge to design chemical processes, most ML approaches that are used in the context of chemical process design currently do not rely on big data [8]. Instead, most ML approaches used for process synthesis rely on regression models that are trained on manually collected datasets for specific applications [4]. For example, there exist numerous works that train surrogate models on process simulations and subsequently optimize the process design using superstructure formulations (e.g., [9–14]). Although these works frequently deal with a large number of data points (i.e., big in volume), there is a lack of variability in the data. In particular, most previous approaches do not consider a variety of process topology data from different processes but rather keep the considered topology fixed. Since ML models cannot extrapolate, these isolated approaches are limited to their specific process applications and validity domains [15].

ML methods have the potential to learn from process typologies and assist the process design in the future. Interestingly, a few pioneering works presented methods that have the potential to learn from multiple processes. Gani and coworkers extended a group contribution method to flowsheet graphs to estimate the process performance and guide decision-making [16, 17]. Also, Sahinidis and co-workers identified common patterns in flowsheet graphs [18, 19]. Recently, we also proposed new generative AI algorithms for the

autocompletion of flowsheets [20], autocorrection of flowsheets [21], and automatic prediction of control structures [22]. However, these methods have not yet unfolded their full potential because they have only been applied to datasets with few instances of flowsheet graphs and lack physical knowledge. In this article, I will focus on the methods to make more information accessible to the ML algorithms through data mining.

Scientific literature and patents provide much information about chemical engineering processes (cf. big scholarly data [23]). Flowsheets are the most important building blocks to define and communicate the structure of chemical processes [24]. As shown in **Figure 1**, they are schematic drawings describing overall process design, i.e., interconnection and type of unit operations. Depending on the development phase of a process, there exist different flowsheets with varying levels of detail (i.e., ranging from block flow diagrams (BFDs) to process flow diagrams (PFDs) and piping and instrumentation diagrams (P&IDs)) [24]. There is at least one flowsheet for every chemical process ever developed or built. These flowsheets are commonly available in PDF format in scientific publications, simulation files, patents, and company reports. Most information about industrial processes is confidential and unavailable for public research. However, in this article, I focus on flowsheet information extraction from publicly available patents and scientific publications. In the future, the proposed methods can also be used on industry data. In addition to flowsheet images, process descriptions and stream tables usually provide additional information about flow compositions, operating conditions, and sizing of the unit operations (see **Figure 1**).

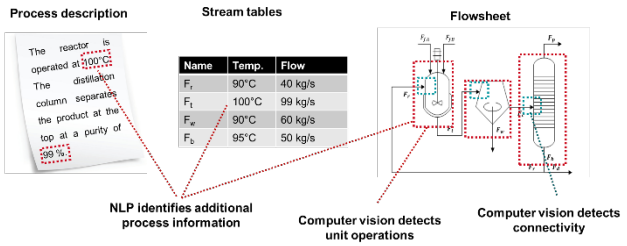

Figure 1. Illustration of the potential for information extraction from flowsheets, process descriptions, and stream tables. Note that NLP is the abbreviation for natural language processing.

The document-centric workflow in chemical process development is inadequate. Since the existing data is unstructured, necessitating the need for chemical engineers to manually review the literature to learn about existing process designs for their specific application. Manually reviewing, verifying, and utilizing this vast amount of unstructured data is not only cumbersome but also can be inaccurate. Given the sheer number of existing flowsheets, no human can comprehend all information that has been incorporated into flowsheets. Understanding how to store, structure, and link this vast amount of chemical flowsheet data and knowledge is key to further progress.

Semantic Web (SW) technologies offer functionality to connect previously isolated pieces of data and knowledge, associate meaning to them, and represent knowledge extracted from them. In particular, SW addresses data variety, by proposing graphs as a unifying data model, to which a data source can be mapped [25]. Such graphs not only contain data, but also metadata and domain knowledge (ontologies containing axioms or rules), all in the same uniform structure, and are then called knowledge graphs (KGs) (i.e., ontology + data = knowledge graph) [26, 27].

I envision that document-centric process information will be transformed into a findable, publicly accessible, interoperable, and reusable (FAIR) [28] knowledge base by representing information through a KG (cf. efforts to structure scholarly information [29]). The first step towards my vision involves the automatic extraction of information from flowsheets and process descriptions in scientific literature and patents. As illustrated in **Figure 1**, the automated extraction of information necessitates a combination of natural language processing (NLP) and computer vision techniques. In this contribution, I propose and concept of "flowsheet mining". Moreover, I review relevant interdisciplinary literature and outline perspectives for future research.

FLOWSHEET MINING

As illustrated in **Figure 2**, I propose a four-step approach for flowsheet mining. In Step 1, publications are automatically downloaded, relevant publications are identified, and flowsheet figures are extracted. In Step 2, flowsheet figures are digitized and saved in a graph format. In Step 3, information is extracted from process descriptions and stream tables. In Step 4, the extracted information is semantically enriched and saved in a knowledge graph.

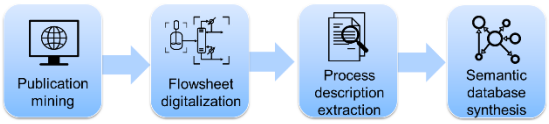

Figure 2. Proposed four-step approach for flowsheet mining.

Publication mining

The goal of the publication mining step is to identify relevant publications that describe a chemical process and extract flowsheet images (see **Figure 3**).

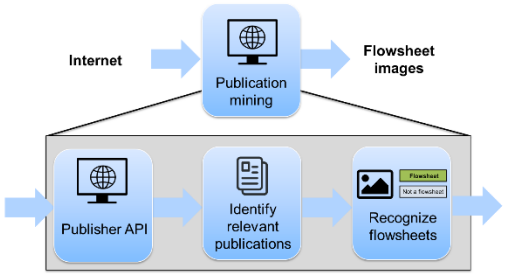

Figure 3. Illustration of Step 1 of the overall flowsheet mining approach: The publication mining.

The automated download of scientific publications and patents is possible through application programming interfaces (APIs). For example, CrossRef provides metadata of scientific publications through a Representational State Transfer (REST) API given a digital object identifier (DOI) [30]. This metadata includes titles, authors, license information, and links to the documents (e.g., publications in PDF or XML format), which are hosted by the corresponding publishers.

Given the large number of scientific publications, the automated identification of relevant publications is crucial. For example, crossref currently stores metadata for over 120 million records (January 2021). To identify chemical engineering publications, I propose to generate a list of all chemical engineering journals.

Only a small fraction of chemical engineering publications describe processes. Thus, the goal is to further identify chemical engineering publications that most likely contain a process flowsheet. I propose to train a topic model on the abstracts, title, and keywords of chemical engineering publications. Topic models are unsupervised ML models that can predict the topics of publications. The common latent Dirichlet allocation (LDA) for instance is a probabilistic topic model that relies on a bag-of-words approach, which means that the model considers the (tokenized) words of the passed documents disregarding their initial order [31]. To predict if an unseen publication contains a flowsheet, a classification model can be trained on the predicted topic distribution (cf. [32]).

After identifying and downloading relevant publications, the flowsheet images need to be identified. This step necessitates the extraction of all images from a PDF document (e.g., using Python package PyMuPDF or PDFFigures 2.0 [33]). Afterward, I propose to train a classification algorithm that recognizes flowsheet images. The flowsheet image recognition is a fairly simple classification problem as flowsheets are usually black-white technical drawings that follow conventions (e.g., ISO 10628). Given the success of deep convolutional neural networks (CNNs) and transfer learning in computer vision [34–36], I propose to use pre-trained state-of-the-art CNN architectures (e.g.,VGG16 [37]). The results of the whole publication mining step are the extracted images of flowsheets from scientific publications.

In our recent work, we demonstrate that flowsheet images can be recognized from literature [38]. We trained a CNN on a training set including about 1,000 PFDs and about 13,000 other images. The model showed a good overall performance with a precision of 80.7% and a recall of 94.4%. In a preliminary study, we identified about 2,500 PFDs in the journal Computers & Chemical Engineering which corresponds to approximately 4.5% of all images in the journal. Moreover, we identified about 2,300 PFDs in the journal Chemical Engineering Science and about 560 PFDs in the book Ullmann's Encyclopedia of Industrial Chemistry.

Flowsheet digitization

The goal of the flowsheet digitization step is to extract the flowsheet topologies from the flowsheet images and save them in a graph format (see **Figure 4**). The digitization of chemical process flowsheets involves an object detection step [39] and a pathway exploration step. In the object detection step, a model identifies the position and type of unit operation on the flowsheet. In the pathway exploration step, the connectivity of the unit operations is explored.

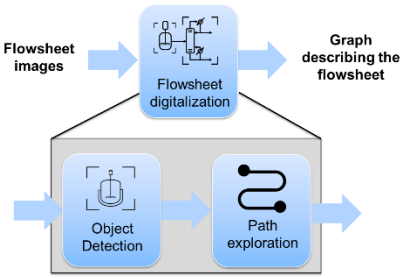

Figure 4. Illustration of Step 2 of the overall flowsheet mining approach: The flowsheet digitization.

Digitization of various types of engineering diagrams has been a focus of research in the computer vision domain since the 1980s, but most developed methods relied on classic computer vision methods (e.g., [40–46]). For example, Okazaki et al. [42] combined template matching and feature extraction in a hybrid model for symbol identification in circuit drawings. However, template matching can only be utilized when all drawings are identical in shape. With the emergence of deep CNNs, the field of computer vision has seen great advancements [47].

Recent literature distinguishes one stage from two-stage object detection algorithms [39]. One stage CNNs solve the tasks of (1) localizing and (2) classifying objects within a single network. Common frameworks of this network type are YOLO [48], its successors [49–51] as well as RetinaNet [52]. Single-stage networks have been

employed in the context of P&ID digitization [53–55]. However, single-stage networks do not perform as well as two-stage networks on common benchmark datasets and they often lack accuracy for the detection of small objects [50, 56].

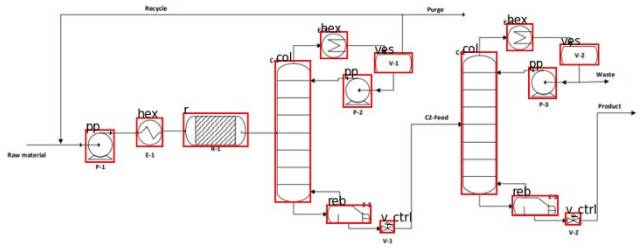

Figure 5. Illustration of our preliminary object detection algorithm for the digitization of PFDs.

Two-stage detectors extract a feature map from the original image using a backbone network in the first stage, which is then used to find regions of interest and classify them. The most common two-stage detector is Faster R-CNN [57]. Recently, the potential of two-stage object detection for process symbols and piping detection has been demonstrated [58–60]. While there exist established algorithms for object detection, the automated exploration of the connectivity is still limited and relies on rule-based approaches [61]. The final result of the flowsheet digitization should be saved in a standardized data exchange format to enable interoperability (cf. DEXPI initiative for data exchange in the process industry [62, 63]).

In our recent work, we demonstrate that convolutional neural networks can digitize P&IDs and PFDs to a high accuracy. **Figure 5** shows the prediction of our object detection algorithm for the digitization of PFDs [60]. The figure shows the identified bounding boxes and associated object class abbreviations (e.g., hex for heat exchanger).

Process description extraction

The goal of the process description extraction step is to extract relevant process information from process descriptions in text format.

In the NLP community, the process of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents is called Information extraction (IE) [64, 65]. As illustrated in Figure 6, this includes several sub-steps that are described in the following.

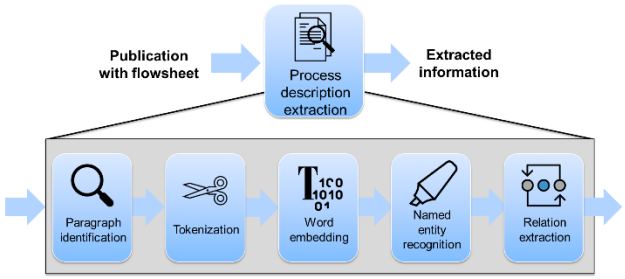

Figure 6. Illustration of Step 3 of the overall flowsheet mining approach: The process description extraction.

The process description is usually only a small part of scientific publications. Thus, the relevant paragraphs including the process description need to be identified. This task is closely related to the identification of relevant publications (see Section Publication mining). Consequently, I propose to use topic models to identify relevant paragraphs.

Tokenization splits text into smaller tokens for further processing [66]. The simplest (rule-based) tokenizers split sentences into tokens based on white spaces, punctuations, or grammatical and syntactical rules. Tokenization is well-established in NLP and there exists a broad variety of implementations and comparisons (e.g., [67]).

In the word-embedding step of the document processing pipeline, each token is represented by a real-valued vector that can be processed by subsequent ML models. The goal is to design a mapping that represents the token's meaning as a vector. For example, the word vectors of the words “king'' and “queen'' should be close to each other because they have a similar meaning. Thus, this step is also referred to as representation learning.

Named entity recognition performs classification on tokens. This is necessary to identify the type of information that is provided. For example, **Figure 7** shows an illustrative example where a named entity algorithm identifies the token “heat exchanger'' as a type [UNIT]. In the relation extraction step, possible relationships between entities are identified (i.e., green arrows in **Figure 7**).

Figure 7. Illustration of tokenization, named entity recognition and relation extraction.

Relation extraction can also involve relation classification, which is typically formulated as a classification problem to classify the relationship between the entities identified in the text [68]. A classifier takes the contextualized representation of two or more entities (e.g., words) as inputs and predicts possible relations between the entities as output. Feature- or kernel-based methods such as the conditional random field are earlier

approaches used for named entity recognition and relation extraction in a pipeline setting [68]. However, the performance of traditional approaches heavily depends on manual feature engineering, which requires domain-specific knowledge and a deep understanding of linguistics [68]. Also, several deep learning architectures have been proposed for entity recognition and relation extraction, which are mostly recurrent types of neural networks based on Bidirectional Long Short Term Memory (BiLSTM) or Gated Recurrent Unit cells [69, 70]. These models are often available in NLP tools (e.g., spaCy, Stanford CoreNLP, AllenNLP, and IBM Watson Natural Language Understanding).

Recently, transformer language models [71] have become the de-facto standard for representation learning in NLP allowing for domain-specific transfer learning. Compared to transformer models that benefit from abundant knowledge from pre-training and strong feature extraction capability, approaches based on BiLSTM have shown a lower generalization performance [68] and are less efficient in capturing long-distance context (due to vanishing gradients in the training process) [65]. Specifically, Bidirectional Encoder Representations from Transformers (BERT) [72] is a common language model that utilizes bidirectional attention mechanism and large-scale unsupervised corpora to obtain effective context-sensitive representations of each word in a sentence [68]. Moreover, various improved variants of BERT have been proposed for various downstream NLP tasks. For example, BERT-based approaches have been applied to scientific texts (e.g, SciBERT [73]) and have even been adapted for molecular property prediction (e.g., ChemBERTa [74]). Recently, large language models (LLMs) such as ChatGPT have also solved various IE tasks through prompting.

The majority of traditional IE approaches are focused on informal text (e.g., social media texts), or biomedical text (e.g., PubMedInfo Crawler [75]), while only a little literature on chemical engineering-related text exists (e.g., chemical patents [76]). In particular, very little attention is paid to IE for chemical process design. Xu et al. [77] showed that bio-entity extraction based on BioBERT [78] can significantly outperform other methods. Thus, a great potential for transformer-based language models on chemical engineering entity extraction tasks can be expected. However, relation extraction is highly domain-specific, due to the variety of underlying ontologies and relation types. To overcome this issue, approaches that combine LLMs, information retrieval, and SW technologies are promising (cf. [79]).

Semantic database synthesis

The goal of the semantic database synthesis step is to build a knowledge graph by integration of the extracted information (see **Figure 8**).

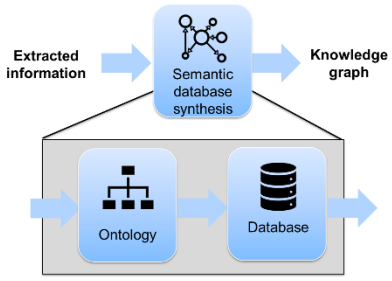

Figure 8. Illustration of Step 4 of the overall flowsheet mining approach: The semantic database synthesis.

A KG is an effective means for capturing and structuring a large amount of multi-relational data from various disciplines [26]. KGs are typically represented in the standardized Resource Description Framework (RDF) data model and queried using the SPARQL Protocol and RDF Query Language. It can be formally defined as $G = \{E, R, T\}$, where G is a labeled and directed multi-graph, and E, R, T are the sets of entities, relations, and triples, respectively. Each triple is formalized as $(u, e, v) \in T$, where $u \in E$ is the head node, $v \in E$ is the tail node, and $e \in R$ is the edge connecting u and v [27]. KGs are usually stored in a graph database system (e.g., Neo4j or virtuoso).

Ontologies are semantic data models that define the types of things that exist in a domain and the properties that can be used to describe them, including the relationships between them [27]. The main components of an ontology are individuals (instances or objects) of classes (distinct types of concepts that exist in the data), relationships (ways to relate classes and individuals), and attributes (aspects, properties, features, characteristics that objects and classes can have). RDF Schema (RDFS) and OWL (Web Ontology Language) are two different language models that enable the construction of quantified statements in the form of RDF graphs. Currently, a few ontologies have been developed for process systems [80, 81], process safety [82, 83], process operation [84–86], and kinetic reactions [87, 88]. However, these ontologies are mostly generalized data models that have not yet been used in conjunction with big data to form a KG. Currently, there exists only one publicly available chemical engineering KG, which creates a digital twin of the eco-industrial park on Jurong Island, Singapore [89]. To build the process flowsheet KG, I recommend extending the ONTOCAPE ontology [80, 81]. Furthermore, I recommend linking or integrating flowsheet information into existing open scholarly KGs, e.g., the open research knowledge graph (ORKG) [90].

CONCLUSIONS

In the current setting, information about chemical

processes is often not easily findable, accessible, interoperable, and reusable (FAIR). In particular, process design information is depicted in flowsheets and described in text format (e.g., in scientific literature, patents, or company reports). This lack of structured data is a major hurdle for the development of processes. Also, the lack of structured data is a bottleneck for process design through AI algorithms.

This article describes the steps of a larger ongoing research agenda that aims to build an open Chemical Engineering Knowledge Graph (ChemEngKG). I envision that all (publicly available) chemical processes will be accessible in the ChemEngKG. To achieve this goal, I proposed the concept of "flowsheet mining" that will enable the autonomous extraction of chemical process information from scientific literature. The proposed concept mines scientific literature and patents, identifies and digitizes flowsheet images, extracts information from process descriptions, and saves all information in a semantic database. Thus, flowsheet mining requires the integration and further development of algorithms from several domains. I review relevant interdisciplinary literature in data mining, computer vision, natural language processing, and semantic web technologies. Moreover, I highlighted the potentials for the development of algorithms that are relevant for flowsheet mining.

Finally, I envision that the ChemEngKG will be an enabler technology for innovative generative AI algorithms facilitating chemical process design. There exist a multitude of powerful ML algorithms that have already demonstrated breakthrough results in other domains such as molecular design. I believe that these methods have also a great potential for AI-assisted process design. First methods already show the potential of AI for the autocompletion, autocorrection, and auto-generation of flowsheets. In the future, further development is needed to integrate engineering knowledge, big data, and AI for process design.

ACKNOWLEDGEMENTS

The author acknowledges NWO funding for the Veni talent programme.

REFERENCES

1. A. Mitsos, N. Asprion, C. A. Floudas, M. Bortz, M. Baldea, D. Bonvin, A. Caspari, and P. Schäfer, *Comput Chem Eng* 113:, 209–221 (2018)
2. D. W. Green and M. Z. Southard, *Perry's chemical engineers' handbook*, McGraw-Hill Education (2019)
3. V. Venkatasubramanian, *AIChE Journal* 65:, 466–478 (2019)
4. J. H. Lee, J. Shin, and M. J. Realff, *Comput Chem Eng* 114:, 111–121 (2018)
5. Y. LeCun, Y. Bengio, and G. Hinton, *Nature* 521:, 436–444 (2015)
6. A. De Mauro, M. Greco, and M. Grimaldi, *Library Review* (2016)
7. L. Zhou, S. Pan, J. Wang, and A. V Vasilakos, *Neurocomputing* 237:, 350–361 (2017)
8. M. Wiedau, G. Tolksdorf, J. Oeing, and N. Kockmann, *Chemie Ingenieur Technik* 93:, 2105–2115 (2021)
9. C. A. Henao and C. T. Maravelias, *AIChE Journal* 57:, 1216–1232 (2011)
10. I. Fahmi and S. Cremaschi, *Comput Chem Eng* 46:, 105–123 (2012)
11. M. Jones, H. Forero-Hernandez, A. Zubov, B. Sarup, and G. Sin, Superstructure optimization of oleochemical processes with surrogate models, in *Computer Aided Chemical Engineering*, Elsevier (2018), pp. 277–282
12. H. A. Pedrozo, S. B. R. Reartes, Q. Chen, M. S. D\'\iaz, and I. E. Grossmann, *Comput Chem Eng* 141:, 107015 (2020)
13. W. R. Huster, A. M. Schweidtmann, and A. Mitsos, *Optimization and Engineering* 21:, 517–536 (2020)
14. D. Rall, A. M. Schweidtmann, B. M. Aumeier, J. Kamp, J. Karwe, K. Ostendorf, A. Mitsos, and M. Wessling, *J Memb Sci* 600:, 117860 (2020)
15. A. M. Schweidtmann, J. M. Weber, C. Wende, L. Netze, and A. Mitsos, *Optimization and Engineering* 1–22 (2021)
16. L. d'Anterroches and R. Gani, *Fluid Phase Equilib* 228:, 141–146 (2005)
17. A. K. Tula, M. R. Eden, and R. Gani, *Comput Chem Eng* 81:, 245–259 (2015)
18. T. Zhang, N. V Sahinidis, and J. J. Siirola, *AIChE Journal* 65:, 592–603 (2019)
19. C. Zheng, X. Chen, T. Zhang, N. V Sahinidis, and J. J. Siirola, *Comput Chem Eng* 107676 (2022)
20. G. Vogel, L. Schulze Balhorn, and A. M. Schweidtmann, *Comput Chem Eng* 171:, 108162 (2023)
21. L. S. Balhorn, M. Caballero, and A. M. Schweidtmann, Toward autocorrection of chemical process flowsheets using large language models, (2023)
22. E. Hirtreiter, L. Schulze Balhorn, and A. M. Schweidtmann, *AIChE Journal* n/a:, e18259
23. F. Xia, W. Wang, T. M. Bekele, and H. Liu, *IEEE Trans Big Data* 3:, 18–35 (2017)
24. G. Nasby, *Chem Eng Prog* 108:, 36–44 (2012)
25. P. Hitzler, M. Krötzsch, and S. Rudolph, *Foundations of semantic web technologies*, CRC press (2009)
26. X. Wilcke, P. Bloem, and V. De Boer, *Data Science* 1:, 39–57 (2017)
27. A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G.

de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, and others, *arXiv preprint arXiv:2003.02320* (2020)
28. M. D. Wilkinson, M. Dumontier, Ij. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, and others, *Sci Data* 3:, 1–9 (2016)
29. S. Auer, V. Kovtun, M. Prinz, A. Kasprzik, M. Stocker, and M. E. Vidal, Towards a knowledge graph for science, (2018), pp. 1–6
30. R. Lammey, *Insights* 28: (2015)
31. D. M. Blei, A. Y. Ng, and M. I. Jordan, *Journal of machine Learning research* 3:, 993–1022 (2003)
32. M. Pavlinek and V. Podgorelec, *Expert Syst Appl* 80:, 83–93 (2017)
33. C. Clark and S. Divvala, PDFFigures 2.0: Mining figures from research papers, (2016), pp. 143–152
34. A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, *Comput Intell Neurosci* 2018: (2018)
35. A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, *Artif Intell Rev* 53:, 5455–5516 (2020)
36. W. Rawat and Z. Wang, *Neural Comput* 29:, 2352–2449 (2017)
37. K. Simonyan and A. Zisserman, *arXiv preprint arXiv:1409.1556* (2014)
38. L. Schulze Balhorn, Q. Gao, D. Goldstein, and A. M. Schweidtmann, Flowsheet Recognition using Deep Convolutional Neural Networks, in *In Press: Computer Aided Chemical Engineering*, Elsevier (2022)
39. Z.-Q. Zhao, P. Zheng, S. Xu, and X. Wu, *IEEE Trans Neural Netw Learn Syst* 30:, 3212–3232 (2019)
40. H. Bunke, Automatic Interpretation of Lines and Text in Circuit Diagrams, (1982), pp. 297–310
41. C. Howie, J. Kunz, T. Binford, T. Chen, and K. H. Law, *Advances in Engineering Software* 29:, 563–570 (1998)
42. A. Okazaki, T. Kondo, S. Tsunekawa, and E. Kawamoto, *IEEE Trans Pattern Anal Mach Intell* 10:, 331–341 (1988)
43. F. C. A. Groen, A. C. Sanderson, and J. F. Schlag, *Pattern Recognit Lett* 3:, 343–350 (1985)
44. C.-S. Fahn, J.-F. Wang, and J.-Y. Lee, *Comput Vis Graph Image Process* 44:, 119–138 (1988)
45. M. K. Gellaboina and V. G. Venkoparao, Graphic Symbol Recognition Using Auto Associative Neural Network Model, (2009), pp. 297–301doi:10.1109/ICAPR.2009.45
46. C. F. Moreno-García, E. Elyan, and C. Jayne, *Neural Comput Appl* 31:, 1695–1712 (2019)
47. A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Adv Neural Inf Process Syst* 25:, 1097–1105 (2012)
48. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You only look once: Unified, real-time object detection, (2016), pp. 779–788
49. J. Redmon and A. Farhadi, YOLO9000: Better, Faster, Strongerat <https://arxiv.org/pdf/1612.08242>
50. A. Farhadi and J. Redmon, Computer Vision and Pattern Recognition, cite as (2018)
51. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detectionat <https://arxiv.org/pdf/2004.10934>
52. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, Focal loss for dense object detection, (2017), pp. 2980–2988
53. B. Sezen, *Evaluation of Machine Learning Algorithms for Object Detection in Technical Drawings like P&IDs and Circuit Diagrams*, Technical University of Munich Press (2019)at <https://www.researchgate.net/publication/340307540_Evaluation_of_Machine_Learning_Algorithms_for_Object_Detection_in_Technical_Drawings_like_PIDs_and_Circuit_Diagrams>
54. E. Elyan, L. Jamieson, and A. Ali-Gombe, *Neural Netw* 129:, 91–102 (2020)
55. Yu, Cha, Lee, Kim, and Mun, *Energies (Basel)* 12:, 4425 (2019)
56. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, Feature Pyramid Networks for Object Detectionat <https://arxiv.org/pdf/1612.03144>
57. S. Ren, K. He, R. Girshick, and J. Sun, *IEEE Trans Pattern Anal Mach Intell* 39:, 1137–1149 (2016)
58. W. Gao, Y. Zhao, and C. Smidts, *Progress in Nuclear Energy* 128:, 103491 (2020)
59. D.-Y. Yun, S.-K. Seo, U. Zahid, and C.-J. Lee, *Applied Sciences* 10:, 4005 (2020)
60. M. F. Theisen, K. N. Flores, L. Schulze Balhorn, and A. M. Schweidtmann, *Digital Chemical Engineering* 6:, 100072 (2023)
61. S. Mani, M. A. Haddad, D. Constantini, W. Douhard, Q. Li, and L. Poirier, Automatic digitization of engineering diagrams using deep learning and graph search, (2020), pp. 176–177
62. M. Wiedau, L. von Wedel, H. Temmen, R. Welke, and N. Papakonstantinou, *Chemie Ingenieur Technik* 91:, 240–255 (2019)
63. S. Fillinger, H. Bonart, W. Welscher, E. Esche, and J.-U. Repke, *Chemie Ingenieur Technik* 89:, 1454–1463 (2017)
64. E. D. Liddy, *Encyclopedia of Library and Information Science, 2nd Ed* (2001)at <http://surface.syr.edu/cgi/viewcontent.cgi?article=1019&context=cnlp>
65. D. Jurafsky and J. H. Martin, *Speech and language processing, second edition*, Harlow, Pearson Education (2014)
66. J. J. Webster and C. Kit, Tokenization as the initial

phase in NLP, (1992)
67. Y. He and M. Kayaalp, Bethesda, MD: The Lister Hill National Center for Biomedical Communications 48: (2006)
68. K. Xue, Y. Zhou, Z. Ma, T. Ruan, H. Zhang, and P. He, Fine-tuning BERT for joint entity and relation extraction in chinese medical text, (2019), pp. 892–897
69. S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, *arXiv preprint arXiv:1706.05075* (2017)
70. M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, *arXiv preprint arXiv:1802.05365* (2018)
71. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, (2017), pp. 5998–6008
72. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *arXiv preprint arXiv:1810.04805* (2018)
73. I. Beltagy, K. Lo, and A. Cohan, SciBERT: Pretrained Language Model for Scientific Text, (2019)
74. S. Chithrananda, G. Grand, and B. Ramsundar, *arXiv preprint arXiv:2010.09885* (2020)
75. A. Kanwal, S. Fazal, A. I. Bhatti, M. Ullah, and M. A. Khalid, *Meta Gene* 20:, 100550 (2019)
76. C. Sun, Z. Yang, L. Wang, Y. Zhang, H. Lin, and J. Wang, *arXiv preprint arXiv:2009.01560* (2020)
77. J. Xu, S. Kim, M. Song, M. Jeong, D. Kim, J. Kang, J. F. Rousseau, X. Li, W. Xu, V. I. Torvik, and others, *arXiv preprint arXiv:2005.04308* (2020)
78. J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, *Bioinformatics* 36:, 1234–1240 (2020)
79. R. Anantharangachar, S. Ramani, and S. Rajagopalan, *arXiv preprint arXiv:1302.1335* (2013)
80. J. Morbach, A. Yang, and W. Marquardt, *Eng Appl Artif Intell* 20:, 147–161 (2007)
81. W. Marquardt, J. Morbach, A. Wiesner, and A. Yang, *OntoCAPE: A Re-Usable Ontology for Chemical Process Engineering*, Springer Publishing Company (2010)
82. S. Natarajan, K. Ghosh, and R. Srinivasan, *Comput Chem Eng* 46:, 124–140 (2012)
83. M. Rodriguez and J. Laguia, *Chem Eng Trans* 77:, 67–72 (2019)
84. R. Batres and Y. Naka, Process plant ontologies based on a multi-dimensional framework, (2000), pp. 433–437
85. R. Batres, A. Aoyama, and Y. Naka, *Comput Chem Eng* 26:, 487–498 (2002)
86. E. Muñoz, A. Espuña, and L. Puigjaner, *Comput Chem Eng* 34:, 668–682 (2010)
87. F. Farazi, J. Akroyd, S. Mosbach, P. Buerger, D. Nurkowski, M. Salamanca, and M. Kraft, *J Chem Inf Model* 60:, 108–120 (2019)
88. F. Farazi, M. Salamanca, S. Mosbach, A. Eibeck, L. K. Aditya, A. Chadzynski, K. Pan, X. Zhou, S. Zhang, and others, *ACS Omega* 5:, 18342–18348 (2020)
89. A. Eibeck, M. Q. Lim, and M. Kraft, *Comput Chem Eng* 131:, 106586 (2019)
90. M. Y. Jaradeh, A. Oelen, K. E. Farfar, M. Prinz, J. D'Souza, G. Kismihók, M. Stocker, and S. Auer, Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge, (2019), pp. 243–246