Information Extraction from Text Documents for the Semantic Enrichment of Building Information Models of Bridges

P. Schönfelder[1], T. Al-Wesabi[2], A. Bach[2] and M. König[1]

[1]Department of Civil and Environmental Engineering, Ruhr University Bochum, Germany
[2]Schüßler-Plan Ingenieurgesellschaft mbH, Düsseldorf, Germany

phillip.schoenfelder@rub.de, talwesabi@schuessler-plan.de, abach@schuessler-plan.de, koenig@inf.bi.rub.de

Abstract -

The majority of innovative approaches in the realm of the retrospective generation of building information models for existing structures deal with geometry extraction from point clouds or engineering drawings. However, many building-specific o r o bject-specific at tributes fo r th e en richment of building models cannot be inferred from these geometric and visual data sources, and thus their acquisition requires the analysis of textual building documentation. One type of such documents are structural bridge records, which include specifications r egarding u sed m aterial, l ocation, structural health, modifications, a nd a dministrative d ata. T he documents are semi-structured and hardly allow a robust information extraction based on traditional programming, since the implementation of such an approach would result in a complex nesting of conditional clauses, which is not guaranteed to remain effective for future versions of the document structure. Therefore, a data-driven approach is adopted for the information extraction. This paper demonstrates an end-to-end semantic enrichment method, taking a bridge status report as input and feeding structured object parameters directly to the building information modeling software for the enrichment of the model. The proposed method requires little user interaction and achieves production-ready accuracy. It is tested on an as-built model of an actual bridge and shows promising results.

Keywords -

Building information modeling; Information extraction; Semantic enrichment; Natural language processing; Named entity recognition; Machine learning

1 Introduction

In the operations and maintenance phase of a structure, building information modeling (BIM) can facilitate the access to important condition data. [1] For the majority of the bridges currently in use, however, an as-is model does not exist. [2] Since the manual and retrospective modeling of these structures is a demanding task even for expert engineers [3], current research aims at the full or partial automation of the task. The geometrical aspect of the model is mostly dealt with by the acquisition and processing of point clouds, images or construction plans. [4, 5] In any case, to produce a useful BIM model, not only the structure's geometry, but also semantic information about the covered parts has to be provided and included into the model. [6] This sort of information is hardly present in 2D drawings and not at all in point clouds or images. Therefore, additional sources of information have to be utilized. For bridges specifically, these can be documents like structural records or inspection reports.

In particular, this study deals with the automatic extraction of the needed information from structural bridge records and the integration of the discovered information into bridge models with BIM software. The source documents are text-based and semi-structured in the sense that the data is represented in an almost tabular manner, which is however not encoded as a formal table in the document. This is to facilitate the common exchange and storage of the documents in widely available formats such as PDFs. For that purpose, multiple strategies are pursued to extract the textual data, and to fill in pre-defined building part attributes. After the relevant information is extracted, it is fed into the modeling software Autodesk Revit by means of an import tool developed with the open source visual programming language Dynamo. The proposed method therefore represents an end-to-end semantic enrichment procedure for bridge digital twins. Both labor-intensive tasks, namely the extraction of information from text documents and its integration into BIM models, are addressed by the method. This study therefore contributes to automating the process of semantic enrichment.

The paper is organized as follows: In Section 2, an overview of existing semantic enrichment approaches for digital twins is provided. Also, prior works towards information extraction from bridge documents are listed. Section 3 gives deeper insight into the structure of the source document type for this study and deals with the underlying implementation of the proposed information extraction method. It also includes an overview of the model enrichment procedure. In Section 4, the information extraction method is tested by using a test set of admissible bridge records. To give an impression of the practical value provided by the methodology, it is demonstrated in Section 5 by the example of a real world bridge, along with its respective BIM geometry and the carried out enrichment. Section 6 concludes the paper by discussing some of the methodology limitations and an outlook to future works.

2 Related Works

Though the number of scientific contributions on the subject of semantic enrichment of BIM models is limited, the studies [7, 8, 9] provide elaborate literature reviews. Previous works have focused mainly on inferring semantic information from the BIM model geometry itself. Belsky et al. [10] presented the SeeBIM software, which allows engineers to manually define rules to create new object relations in BIM models according to the defined conditions. For example, adjacent objects can be aggregated if certain requirements are fulfilled, e.g., matching object types and near proximity. An extension of the software is shown in [11], which also includes the possibility to add alphanumerical information from external sources (e.g., bridge management systems provided by highway agencies) to the model. Bloch and Sacks [8] presented an approach to enrich objects with a semantic specification by inferring it from their geometry with the help of machine learning. In particular, in this example, the room types of a building story are inferred from the floor plan geometry. In comparison with a rule-based approach, the machine learning approach shows superior results in this example. In an approach by Isailović et al. [12], damage information is gathered by means of point cloud acquisition and processing and is later inserted into the BIM model of a bridge. In this example, the model is enriched with detailed geometric and with semantic information.

A difference of the method proposed in this paper to most of the previous semantic enrichment approaches is the source of the additional semantic information. Instead of inferring the semantic information directly from the model's parts (e.g., by their shape and adjacencies), in this approach, external documents are consulted and processed.

On the subject of information extraction, there have been some publications with application in the built environment as well: Liu and El-Gohary [13] applied a CRF-based named entity recognition approach to extract information about deficiencies, their causes and other related entities from bridge status reports. Using their developed bridge deterioration ontology [14], they made use of domain-specific semantics to improve the NER classifier. The corpus used in their study consists of the reports for multiple bridges, each including many years of bridge condition records, and is written in natural language. Moon et al. [15] followed a similar method, while choosing a Bi-LSTM architecture instead of the CRF model. Also, they made use of the active learning concept to reduce the annotation effort. Capitalizing on the US National Bridge Inventory database, Li and Harris [16] created a text corpus from Virginia bridge status reports. They trained a Bi-LSTM-CRF classifier to recognize damage types, damage severity and the respective locations in the bridges. It was shown that the model outperforms alternative models for the task at hand [17]. Li et al. [18] propose an innovative recognition method for both flat and nested named entities in bridge inspection reports. It is based on the BERT language model, a Bi-LSTM neural network and uses lexicon augmented word embeddings. Their approach mainly differs from previous approaches by the novelty that the question-answering technique is used to find the desired entities (name, structural elements, location, defects, descriptions) in the text. Liu and El-Gohary [19] developed a dependency parsing method to extract the relations between found entities in bridge reports. They used semantic and syntactic features of the text to train a neural network ensemble classifier and achieved promising results, both for the entity-level recognition and the relation extraction. Inspired by the literature, multiple machine learning models are tested in this study.

3 Methodology

For the understanding of the method presented in the following sub-sections, it is useful to have a basic idea of the source documents' structure. Each bridge record in the document corpus follows a strict table of contents and all reports include the same chapters. However, it is not guaranteed for each chapter to actually display content. If a certain piece of information is not provided, the respective section in the document might be empty. Also, even though the chapters themselves are organized in a very structured way, there exist certain exceptions to that structure, e.g., if special works have been carried out on site and are documented accordingly. After all, varying bridge designs also come with varying document structures – may it be for different numbers of parts or even whole part-structures.

Provided by the regional highway agencies, the records list bridge details such as administrative data (owner, operator), geometric data (position, coordinate system, alignment), mechanical information (material, steel grade, coating) and various other types of information. For a human reader, the documents appear to have tabular structure. However, encoded in the PDF files is only line-by-line text with lots of white space and formatting, which makes the documents *appear* like tables. To recreate the tabular structure, one cannot rely on the occurrence of specific keywords or line-breaks, as the included keys vary from

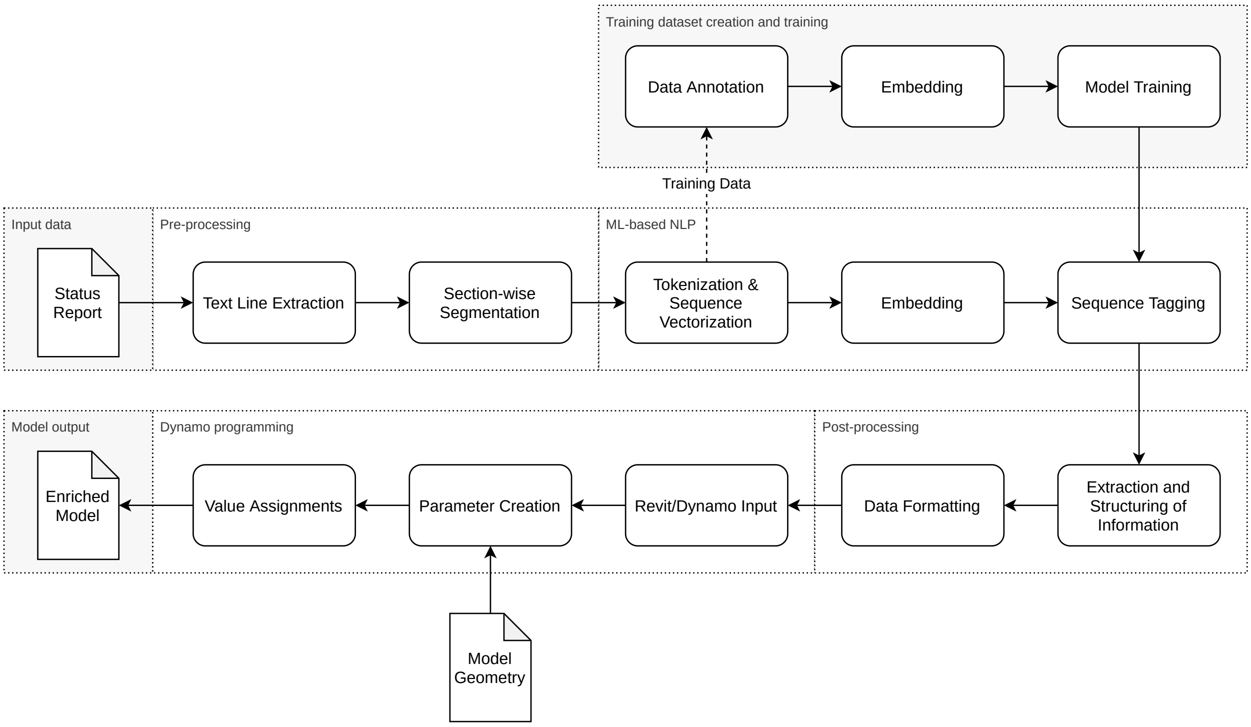

Figure 1. Process diagram of the presented end-to-end extraction and enrichment process.

document to document and as slight imperfections in the formatting are an issue. A traditional programming approach based on regular expression will therefore require an infeasible number of rules and rule exceptions to extract keys and values. Thus, this particular document structure emphasizes the need for a flexible text interpreter.

3.1 Training dataset

To train the machine learning model discussed in Section 3.3, a sufficiently large training dataset is necessary. Therefore, a total of 38 status reports are gathered, pre-processed and annotated manually to be suited for training the neural network and validating the method. The reports cover road and highway bridges in the state of North-Rhine Westfalia in Germany. In total, 98 distinct labels are to be differentiated. Each label represents a parameter name to be extracted with its respective value. Within this study, it is assumed that the documents are free of contradictions, i.e., each parameter is assigned a single value.
It is noted that, depending on the types of bridges involved (e.g., truss or beam bridge) and the used construction material (e.g., steel or concrete), a suitable set of training data files has to be selected to include training cases for all the desired pieces information from the reports.

3.2 Pre-processing

Since the documents originally are in PDF format, the textual data is first detached from the typeset document with the use of a PDF manipulation library. Since there is no inherent table structure in the documents, it is converted from PDF to TXT format line-by-line. Excess white space is removed. Having each report present in TXT format, they are segmented in accordance with the highest hierarchy level of the table of contents, which is the only structure of the document relied upon. This segmentation shortens the length of the individual sequences being fed to the model, and with it the number of unique labels a single model has to differentiate. This is because separate models are trained for each of the document sections (c.f. Section 3.3). After the the segmentation, the training text files are annotated manually, labeling the values of the key-value pairs with the name of the key (see Fig. 2. This label choice facilitates the post-processing, as a mapping from a predicted label to a word sequence will directly represent a key-value pair. Moreover, the labels are given prefixes according to the CoNLL 2002 benchmark format [20], i.e., the label of the first token in an entity is extended by the prefix “B-”, all the following tokens of the entity receive the prefix “I-”. This doubles the number of distinct labels to 196.

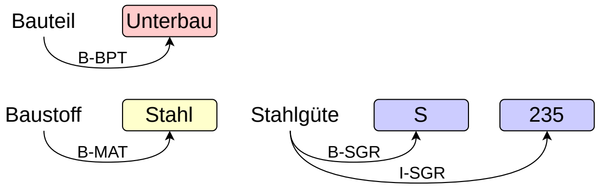

Figure 2. Example of the labeling scheme in a document excerpt. Key names are the label names for the values: Bauteil (building part) → BPT, Baustoff (material) → MAT, Stahlgüte (steel grade) → SGR. Prefixes are in accordance with CoNLL 2002 [20] format.

3.3 Extraction Algorithm

Without all the imperfections to the document structure mentioned in Section 3, a hard-coded approach based on regular expressions would be the ideal way to extract the needed information, however, it simply does not come with the flexibility needed in order to deal with the status reports at hand. Therefore, a more tolerant, data-driven approach is adopted, enabling the processing of documents with slightly different structure, and even for possible future changes to the standard structure.

A fitting candidate approach are recurrent neural networks (RNNs), which are well-suited for sequence processing and have proven useful in all sorts of natural language processing tasks. Hochreiter and Schmidhuber [21] introduced Long-Short-Term-Memory (LSTM) RNNs, which are, in contrast to simple RNNs, able to deal with much longer sequences. Since the bridge record chapters typically include hundreds of words, LSTMs are chosen for the task. Furthermore, bi-directional LSTMs (Bi-LSTMs) may provide an additional advantage over LSTMs, since they can infer a word's label not only with the knowledge of all words before it, but also those after the particular words. This may be of use especially in those situations, in which a certain *key* word does not have an associated *value* following it, if the information is missing. A one-directional LSTM may tend to falsely interpret the following word as a value whatsoever, where a Bi-LSTM may handle the situation better, supposedly. Another possible extension to the model is to append a conditional random field (CRF) layer, which may improve the performance as well [22]. A CRF infers token labels based on the conditional probability that a label occurs given the neighboring labels, thus, it takes the context into account. In the course of this study, multiple model architectures are tested for the sequence tagging: (1) a CRF, (2) an LSTM, (3) an LSTM + CRF layer, (4) a Bi-LSTM, (5) a Bi-LSTM + CRF layer, and (6) a Bi-LSTM followed by another LSTM and a CRF layer. As a classification layer, a fully connected (FC) layer is inserted after the LSTM layers. It serves as a hidden layer in the models which include a CRF for classification. Also, to transform tokens to a vector representation, each model includes an embedding layer, mapping words to vectors of size d_e. All included LSTM layers have the hidden dimension h_d.

To find the set of best hyper-parameters for each of the model architectures, a Bayesian optimization was run with the help of the KerasTuner library [23] to find concrete values for the hidden dimension d_h and the embedding dimension d_e. The optimal hyper-parameters found for the models being applied to the dataset at hand are summarized in Table 1. All the listed models are trained and evaluated in Section 4.

Table 1. Model hyper-parameters found with Bayesian Optimization, by model design.

Model design	d_h	d_e
LSTM	256	928
Bi-LSTM	320	1024
LSTM + CRF	800	896
Bi-LSTM + CRF	512	1024
Bi-LSTM + LSTM + CRF	32	768

3.4 Post-processing

As the model's output is simply a tagged sequence, it needs to be cast into a tabular form for further use. Therefore, consecutive tokens with matching tags are simply concatenated and condensed to a single key-value entry in the output table. The output is saved as a CSV file to serve as input to the desired BIM software.

It is noted that incorrect predictions can lead to contradictory value assignments. Therefore, after the automatic processing of the documents, an engineer still has to examine the output file and resolve potential conflicts.

3.5 Model Enrichment

Dynamo for Revit is a Python-based visual programming tool that is available to Revit users to allow visually constructed scripts and logic. It allows communication with the Revit API in Python which is a great advantage for developers and enables the automatic enrichment of the BIM model as follows:

One step in the process of BIM modeling is the creation of parameters in Revit. However, it is also one of the most time-consuming processes since the creation of the parameters and assigning their values are typically done manually. The complexity goes to great heights when the

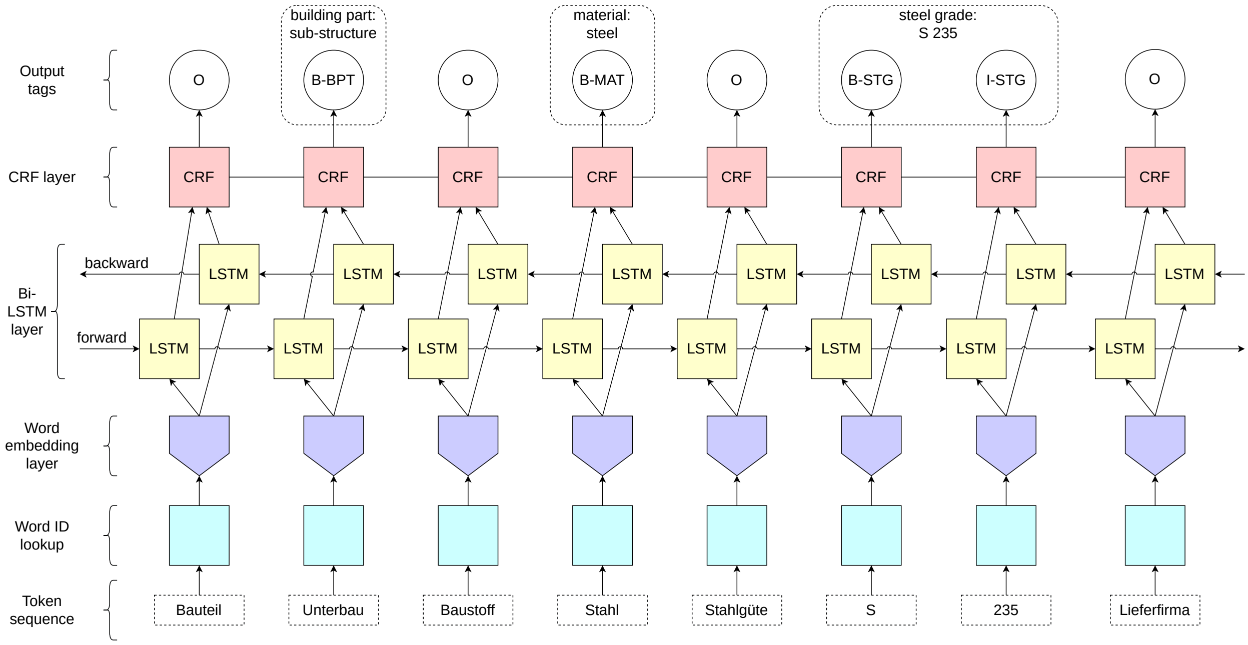

Figure 3. Schematic view of the best performing model design. The example sequence translates to: "Building part substructure building material steel steel grade S 235 supplier"

project has thousands of elements and many layers of classification. Whereas the Revit API provides a great value in automating processes, it does not contain methods of creating project parameters and for that, a workaround is performed through the use of shared parameters. The developed program reads the parameter names and values from the CSV output file of the RNN model. This data is further processed and grouped in Dynamo. The processing of parameters and values includes supplementing them with information to make them readable in Revit. These supplements include adding the data type of each value and choosing the element category for which the list of parameters will be created. After that, the program reads the values and links each parameter with the correct value in order to complete the process of semantic enrichment. In addition, a graphical user interface has been created for the program to be user-friendly and add a level of flexibility for the users. Moreover, the program is integrated with Revit as an add-in in the Revit ribbon to act as a complementary built-in tool to serve the automation, optimization, and ease of use.

4 Validation

To ensure the satisfactory performance of the chosen sequence tagging model, a 10-fold cross-validation experiment is carried out with the annotated corpus. In each of the 10 runs, the dataset is split into 10 parts, thereof 9 for training and one for testing. The performances are averaged to compute the final performance scores summarized in Table 2. The precision P and the recall R are defined by

$$P = \frac{\#\,\mathrm{TP}}{\#\,\mathrm{TP} + \#\,\mathrm{FP}}$$

$$\text{and} \quad R = \frac{\#\,\mathrm{TP}}{\#\,\mathrm{TP} + \#\,\mathrm{FN}},$$

where # TP is the number of true positives, # FP is the number of false positives and # FN is the number of false negatives in the sequence tagger's predictions. The F1 score is the harmonic mean of precision and recall. Since the Bi-LSTM-CRF shows the best performance among the tested model designs, it is selected for the information extraction pipeline.
The models are implemented in TensorFlow 2.7 [24], and all computations are executed under Ubuntu 20.04.3 LTS using an NVIDIA A100 GPU.

5 Demonstration

To test the proposed method on a real-world example, the highway bridge *Hachmannbrücke*, located in Hamburg, Germany, is modeled in Autodesk Revit and enriched with information from its respective bridge record. The

Table 2. Performance scores of the model variations in the 10-fold cross-validation experiments. Each number represents the averaged result from testing the model on the test set of each of the 10 folds.

Model design	P	R	F1
CRF	0.430	0.437	0.416
LSTM	0.970	0.966	0.965
LSTM + CRF	0.959	0.959	0.957
Bi-LSTM	0.967	0.964	0.963
Bi-LSTM + CRF	**0.970**	**0.969**	**0.967**
Bi-LSTM + LSTM + CRF	0.958	0.957	0.955

model comprises 1481 elements.

The user interface is depicted in Figure 4 and aids the engineer as follows: The information extraction pipeline takes a bridge record as input and outputs a CSV file. The user imports the file via the GUI (1) and proceeds with the selection of parameters and their respective values to be imported (2). The GUI snippets (4) and (5) depict the created and the enriched element parameters of the model (3), respectively. In the shown example, new parameters are assigned to elements of the category *Fundamente* (foundations), e.g., *Hauptbaustoff* (material) and *Baujahr* (year of construction). Without additional user input, they take the values as extracted from the original bridge record.

In this example, the method achieves an accuracy of 86%, given the parameter names listed in Figure 4. All of them are extracted correctly, except the parameter *Bemerkungen* (i.e., remarks). Presumably, this is because remarks can have arbitrary content, contrarily to parameters like steel grade or year of construction, which have a limited set of possible values and are, thus, more likely to be recognized by the model.

Without the trained extraction method and the designed Revit GUI, the user has to read into the PDF file, find the desired values, create the needed parameters for the relevant objects and assign the values to them one-by-one. For the selected parameters to be imported, the proposed method greatly facilitates the process compared to the manual editing of model file. Nonetheless, this semi-automatic model enrichment process still depends on a proficient user and their understanding of the Revit software.

6 Conclusions

Few approaches in the research subject of retrospective BIM model creation deal with the semantic enrichment of bridge models, and if they do, their focus is mostly on inferring information from geometry. This paper, however, demonstrates the value of text documents as a source of information for semantic enrichment. The study's contributions are twofold: First, an ML-based information extraction algorithm is proposed for the processing of structural bridge records provided by German highway agencies. It shows promising performance in the cross-validation experiments and for the real-world example document. Second, an end-to-end pipeline is developed for the semantic enrichment of bridge BIM models, including an ML-based information extraction method and a Dynamo tool to incorporate the data into the model. The data integration tool is also available as a Revit add-in, which might promote industry acceptance.

However, the presented approach has three main drawbacks: First, it is limited to bridges and, moreover, to these very kind of documents, at least as far as no other training data is provided. Second, it has only been tested for Revit for now, but since the extracted data is stored in an open format, it can serve as input data for other BIM modeling software or for the enrichment of models saved in IFC format as well. Lastly, as the post-processing does not include any semantic processing to ensure the quality of the results, it is advisable for an engineer to check the exported file for errors before importing it to the model. In follow-up studies, it is anticipated to automate this check for the most apparent extraction errors.

Acknowledgements

This research is conducted as part of the BIMKIT project, funded by the German Federal Ministry for Economic Affairs and Climate Action. The authors would like to express their gratitude towards Schüßler-Plan Ingenieurgesellschaft, who generously provided the bridge model for the demonstration in Section 5, and towards Hamburg Port Authority, the owner of the bridge. Also, the authors thank both the Landesbetrieb Straßen NRW and the Autobahn GmbH for making available enough bridge status reports to train and test our developed algorithms thoroughly.

References

[1] Jack C. P. Cheng, Qiqi Lu, and Yichuan Deng. Analytical review and evaluation of civil information modeling. *Automation in Construction*, 67:31–47, 2016. doi:10.1016/j.autcon.2016.02.006.

[2] Rebekka Volk, Julian Stengel, and Frank Schultmann. Building Information Modeling (BIM) for existing buildings — Literature review and future needs. *Automation in Construction*, 38:109–127, 2014. doi:10.1016/j.autcon.2013.10.023.

[3] André Borrmann, Markus König, Christian Koch, and Jakob Beetz. Building Information Modeling: Why? What? How? In André Borrmann,

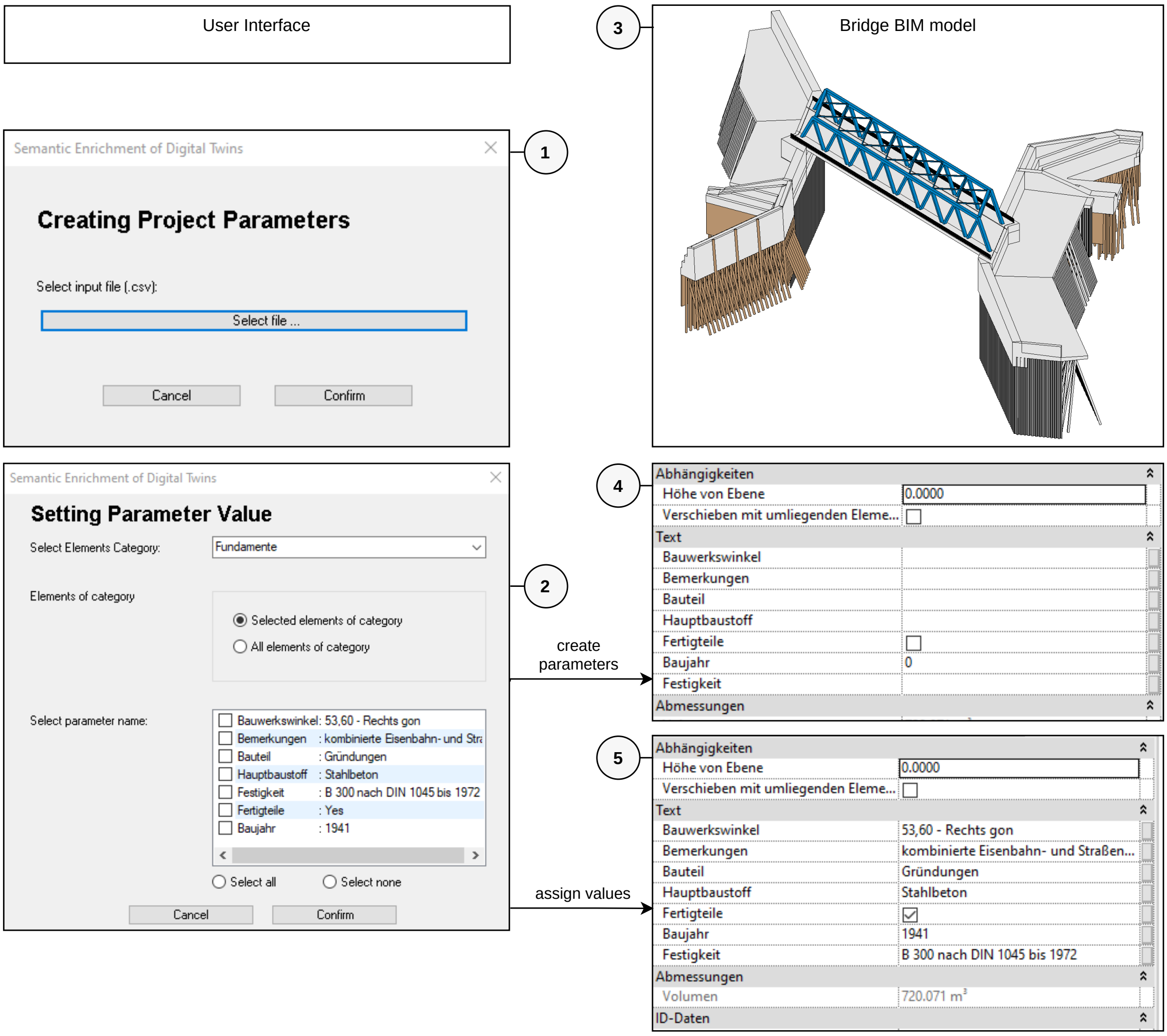

Figure 4. Illustration of the model enrichment user interface. 1: GUI to select the input CSV file, 2: GUI to accept user input to define which element categories should be assigned the respective extracted values, 3: BIM model image, 4: parameter table after parameter creation, 5: parameter table after value insertion.

Markus König, Christian Koch, and Jakob Beetz, editors, *Building Information Modeling: Technology Foundations and Industry Practice*, pages 1–24. Springer International Publishing, Cham, 2018. doi:10.1007/978-3-319-92862-3_1.

[4] Viorica Pătrăucean, Iro Armeni, Mohammad Nahangi, Jamie Yeung, Ioannis Brilakis, and Carl Haas. State of research in automatic as-built modelling. *Advanced Engineering Informatics*, 29(2):162–171, 2015. doi:10.1016/j.aei.2015.01.001.

[5] Lucile Gimenez, Jean-Laurent Hippolyte, Sylvain Robert, Frédéric Suard, and Khaldoun Zreik. Review: reconstruction of 3D building information models from 2D scanned plans. *Journal of Building Engineering*, 2:24–35, 2015. doi:10.1016/j.jobe.2015.04.002.

[6] Christian Koch and Markus König. Data Modeling. In André Borrmann, Markus König, Christian Koch, and Jakob Beetz, editors, *Building Information Modeling: Technology Foundations and Industry Practice*, pages 43–62. Springer International Publishing, Cham, 2018. doi:10.1007/978-3-319-92862-3_3. URL `https://doi.org/10.1007/978-3-319-92862-3_3`.

[7] Nouha Hichri, Chiara Stefani, Livio De Luca, and Philippe Véron. Review of the "as-built BIM" ap-

proaches. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XL-5-W1, pages 107–112, Trento, Italy, 2013. doi:10.5194/isprsarchives-XL-5-W1-107-2013.

[8] Tanya Bloch and Rafael Sacks. Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. *Automation in Construction*, 91:256–272, 2018. doi:10.1016/j.autcon.2018.03.018.

[9] Fábio Matoseiro Dinis, João Poças Martins, Ana Sofia Guimarães, and Bárbara Rangel. BIM and Semantic Enrichment Methods and Applications: A Review of Recent Developments. *Archives of Computational Methods in Engineering*, 2021. doi:10.1007/s11831-021-09595-6.

[10] Michael Belsky, Rafael Sacks, and Ioannis Brilakis. Semantic Enrichment for Building Information Modeling. *Computer-Aided Civil and Infrastructure Engineering*, 31(4):261–274, 2016. doi:10.1111/mice.12128.

[11] Rafael Sacks, Ling Ma, Raz Yosef, Andre Borrmann, Simon Daum, and Uri Kattel. Semantic Enrichment for Building Information Modeling: Procedure for Compiling Inference Rules and Operators for Complex Geometry. *Journal of Computing in Civil Engineering*, 31(6):04017062, 2017. doi:10.1061/(ASCE)CP.1943-5487.0000705.

[12] Dušan Isailović, Vladeta Stojanovic, Matthias Trapp, Rico Richter, Rade Hajdin, and Jürgen Döllner. Bridge damage: Detection, IFC-based semantic enrichment and visualization. *Automation in Construction*, 112:103088, 2020. doi:10.1016/j.autcon.2020.103088.

[13] Kaijian Liu and Nora El-Gohary. Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports. *Automation in Construction*, 81:313–327, 2017. doi:10.1016/j.autcon.2017.02.003.

[14] Kaijian Liu and Nora El-Gohary. Semantic Modeling of Bridge Deterioration Knowledge for Supporting Big Bridge Data Analytics. In *Construction Research Congress 2016: Old and New Construction Technologies Converge in Historic San Juan*, pages 930–939, San Juan, Puerto Rico, 2016. doi:10.1061/9780784479827.094.

[15] Seonghyeon Moon, Sehwan Chung, and Seokho Chi. Bridge Damage Recognition from Inspection Reports Using NER Based on Recurrent Neural Network with Active Learning. *Journal of Performance of Constructed Facilities*, 34(6):04020119, 2020. doi:10.1061/(ASCE)CF.1943-5509.0001530.

[16] Tianshu Li and Devin Harris. Automated construction of bridge condition inventory using natural language processing and historical inspection reports. In *Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure, and Transportation XIII*, volume 10971, pages 206–213, 2019. doi:10.1117/12.2514006.

[17] Tianshu Li, Mohamad Alipour, and Devin K. Harris. Context-aware sequence labeling for condition information extraction from historical bridge inspection reports. *Advanced Engineering Informatics*, 49: 101333, 2021. doi:10.1016/j.aei.2021.101333.

[18] Ren Li, Tianjin Mo, Jianxi Yang, Dong Li, Shixin Jiang, and Di Wang. Bridge inspection named entity recognition via BERT and lexicon augmented machine reading comprehension neural model. *Advanced Engineering Informatics*, 50:101416, 2021. doi:10.1016/j.aei.2021.101416.

[19] Kaijian Liu and Nora El-Gohary. Semantic Neural Network Ensemble for Automated Dependency Relation Extraction from Bridge Inspection Reports. *Journal of Computing in Civil Engineering*, 35 (4):04021007, 2021. doi:10.1061/(ASCE)CP.1943-5487.0000961.

[20] Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In *COLING-02: The 6th Conference on Natural Language Learning*, pages 155–158, Taipei, Taiwan, 2002. URL `https://aclanthology.org/W02-2024`.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-term Memory. *Neural computation*, 9:1735–80, 1997. doi:10.1162/neco.1997.9.8.1735.

[22] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF Models for Sequence Tagging. 2015. arXiv: 1508.01991.

[23] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, and others. Keras Tuner, 2019. URL `https://github.com/keras-team/keras-tuner`.

[24] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, and others. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL `https://www.tensorflow.org/`.