Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 239 (2024) 483–490

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2023

A Method for Extracting BPMN Models from Textual Descriptions Using Natural Language Processing

Josip Tomo Licardo, Nikola Tanković*, Darko Etinger

Juraj Dobrila University of Pula, Faculty of Informatics, Rovinjska 14, Pula 52100, Croatia

Abstract

Business Process Model and Notation (BPMN) is a standard for formally modeling complex business processes. Manual creation of BPMN models can be time-consuming and error-prone, prompting a need for automation. Existing approaches, such as rule-based methods, machine learning, and machine translation, have progressed but face accuracy and real-world applicability challenges. In this research paper, we propose a novel method for automated extraction of BPMN models from textual descriptions using natural language processing (NLP) tools and deep learning models, including the spaCy library for text processing, a fine-tuned BERT model, and state-of-the-art large language models like GPT-3.5-Turbo and GPT-4. We utilize Graphviz, an open-source graph visualization software, to visualize the extracted processes. Our method supports representing tasks, exclusive gateways, parallel gateways, and start and end events in the generated BPMN models. The evaluation of 31 textual descriptions shows that our method generates process models with 96% accuracy using GPT-4 and 80% accuracy using GPT-3.5-Turbo large language models. Although subject to certain limitations, such as occasional inaccuracies in model outputs and reliance on well-formed input text, our approach offers a valuable contribution to the growing body of research on automating BPMN model generation.

Keywords: Business Process Model and Notation (BPMN); natural language processing (NLP); deep learning; information extraction; process modeling.

* Corresponding author.
E-mail address: nikola.tankovic@unipu.hr

10.1016/j.procs.2024.06.196

1. Introduction

Business Process Model and Notation (BPMN) is a widely used standard for modeling business processes. It enables the representation of complex processes to be visually and easily understandable. However, creating BPMN models can take time and effort, especially when dealing with large and complex processes. Natural language processing (NLP) techniques have emerged as a promising approach to automate the generation of BPMN models from textual descriptions, thereby improving the efficiency and accuracy of the process.

Past research has investigated NLP techniques for generating BPMN models, employing machine learning, rule-based approaches, and machine translation, with varied success. However, these approaches lack accuracy and real-world application. This paper proposes a novel method for extracting BPMN models from textual descriptions using NLP techniques, incorporating a pipeline of NLP tools and deep learning models, and employing Graphviz for process visualization.

The paper is structured as follows: first, we review related works in the field of NLP and BPMN in section 2. Next, we present our proposed method in section 3. In section 4, we evaluate the effectiveness and accuracy of our proposed method. Section 5 discusses the limitations of our approach and potential directions for future research. Finally, we conclude the paper in section 6 by summarizing our contributions and discussing the implications of our findings.

2. Related work

This section reviews relevant literature on the automated generation of BPMN process models from textual descriptions using NLP techniques. Approaches for extracting BPMN models from textual descriptions using NLP can be grouped as follows:

Table 1. Methods for extracting BPMN models from the text

Method	Works
Rule-based approaches	[1], [2], [3], [4], [5], [6], [7], [8]
Machine learning and deep learning-based approaches	[9], [10], [11], [13]
Machine translation approaches	[15]
Constrained natural language approaches	[16]

2.1. Rule-based approaches

Rule-based approaches for generating BPMN process models from textual descriptions primarily employ NLP techniques and mapping rules to transform natural language elements into formal models. Fundamental studies, including Friedrich et al. [1], Honkisz et al. [2], and Sholiq et al. [3], utilize NLP tools for syntax parsing and semantic analysis, employing various techniques such as anaphora resolution and Subject-Verb-Object extraction.

Schumacher & Minor [4] emphasize structure-aware knowledge representation and non-sequential control-flow structures, while Ferreira et al. [5] focus on mapping rules and a prototype tool for process element identification. Van der Aa et al. [6] extract declarative process models using constraints, and Quishpi et al. [7] use a tree-based pattern query language for annotation extraction. Finally, Nasiri et al. [8] propose using Prolog language rules and Python code to generate UML activity diagrams from user stories.

2.2. Machine learning and deep learning-based approaches

In machine learning and deep learning, relevant studies can be categorized into task classification and process model extraction approaches. Leopold et al. [9] propose a machine learning-based method for task classification using

linguistic features and support vector machines. In contrast, Qian et al. [10] developed a deep learning-based hierarchical neural network for multi-grained text classification, capturing procedural knowledge.

Pyrtek et al. [11] employ neural networks for generating process models from text, utilizing a deletion-based compression model. This approach potentially reduces the cognitive effort of process modelers. Bellan et al. [12] discuss automatic process discovery, highlighting the need for representative datasets and detailed annotations. They suggest a two-step transformation approach with intermediate representation and explore in-context learning and GPT-3 for extracting information conversationally [13]. The PET dataset [14] addresses the lack of standardized corpora of business process descriptions, enabling objective comparison of extraction approaches.

2.3. Machine translation approaches

Machine translation approaches, such as those by Sonbol et al. [15], provide a novel perspective on generating BPMN models from textual descriptions through two main phases: natural language analysis and BPMN diagram generation. Inspired by semantic transfer-based machine translation techniques, this approach emphasizes capturing and preserving the semantic content while transforming the text into a graphical process representation. Although the results show an 81% similarity to manually created models, it is important to acknowledge the need for further research and refinement to address potential limitations.

2.4. Constrained natural language approaches

Constrained natural language approaches involve using a restricted subset of natural language with specific grammar rules and vocabulary to simplify parsing and interpretation of textual descriptions for generating BPMN models. Ivanchikj et al. [16] exemplify this with their tool, BPMN Sketch Miner, which balances expressiveness and usability for rapid BPMN process model sketching. However, this approach has limitations, including inflexibility, as it relies on a unique syntax that must be learned for generating BPMN models.

3. Proposed method

Our method differs from the related works in several ways. Firstly, we employ large language models for process description analysis, enabling more nuanced information capture and accurate process model generation. Although Bellan et al. [13] use GPT-3 for conversational information extraction from process descriptions, our approach leverages advanced models like GPT-3.5-Turbo and GPT-4 for enhanced accuracy and efficiency, and further generates graphical visualizations of the process. Additionally, our method employs deep learning models for token classification and information extraction, improving accuracy and minimizing manual intervention compared to rule-based or semi-automatic approaches.

The proposed method processes the input text through a pipeline, during which we progressively gather more information on the structure of the process.

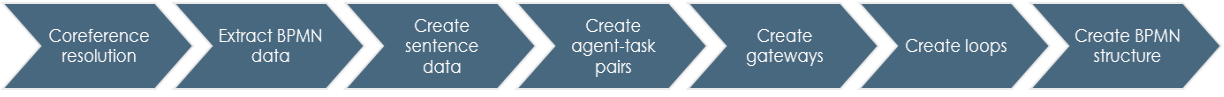

Fig. 1. Overview of the text analysis pipeline.

3.1. Coreference resolution

The initial phase of the method involves processing user input, which begins with coreference resolution. Coreference resolution is a critical aspect of natural language understanding, as it identifies expressions within a text that refer to the same entity or concept.

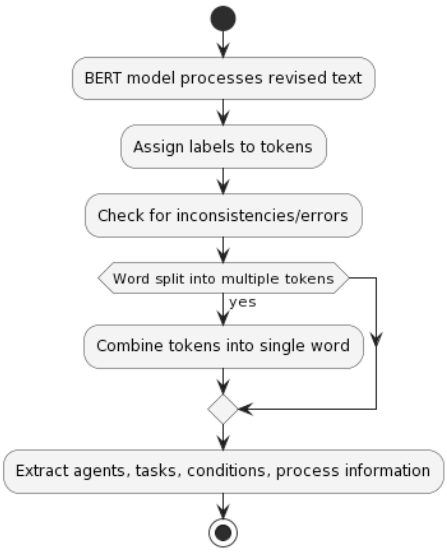

Fig. 2. Extraction of BPMN data using the BERT model

To integrate coreference resolution into our approach, we utilize a spaCy module that is still in its experimental phase at the time of writing. The module utilizes a transformer-based model, which enables the generation of a revised text version by resolving coreferences. The revised text is then supplied to the subsequent stages of the pipeline, ensuring that our method operates on a more coherent and semantically rich input.

3.2. BPMN data extraction

The next phase of our pipeline involves extracting the BPMN data from the revised text. This is achieved using a fine-tuned token classification BERT model hosted on the Hugging Face platform. The model has been fine-tuned on approximately 100 textual process descriptions. The dataset comprises five target labels, namely *agent*, *task*, *task information*, *process information*, and *condition*.

The BERT model processes the revised text obtained after coreference resolution and assigns the appropriate label to each token. After the data is extracted, we check for any inconsistencies or errors in the output. If the model that extracts BPMN data splits a word into multiple tokens, a function is employed to fix the production by combining the tokens into a single word. Once the data has been cleaned and prepared, we extract the agents, tasks, conditions, and process information from the list of extracted entities.

3.3. Creation of sentence data and agent-task pairs

We then create the sentence data using a function that creates a list of dictionaries containing the sentence text, start index, and end index. This function parses the revised text and organizes the data in a structured format.

Next, we create the agent-task pairs by combining agents and tasks based on the sentence they appear in. This is achieved through a dedicated function that analyzes the extracted entities and groups them accordingly. The function ensures that each agent is associated with the correct task, thereby maintaining the integrity of the process model.

3.4. Handling of parallel and exclusive gateways

We then proceed to check for parallel keywords in the text. If the text contains parallel keywords, we extract parallel gateways and paths from the process description using OpenAI's large language models. To

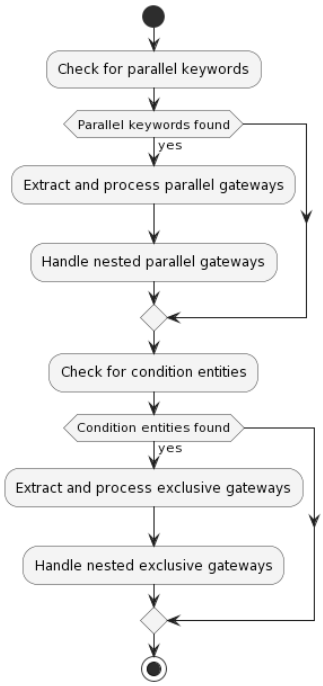

Fig. 3. Handling of parallel and exclusive gateways

accomplish this we employ a prompt designed to extract parallel gateways. To provide context to the model, we also include some examples to demonstrate the desired output.

Our primary objective in this step is to extract the text corresponding to a specific parallel gateway. Once the parallel gateway text is extracted, we locate it within the original process description. By doing so, we can determine the start and end indices of the parallel gateway. For each gateway, we then check if it contains any parallel keywords. If parallel keywords are found within a gateway, we repeat the entire extraction process for that gateway. This is done to account for nested parallel gateways, which may be present in more complex process descriptions.

If the text contains condition entities, we proceed to extract the exclusive gateways. This approach is analogous to the extraction process employed for parallel gateways. Upon obtaining the indices of the exclusive gateway, we employ another prompt to link each condition to a specific exclusive gateway.

Next, we focus on detecting nested exclusive gateways in the text. If any nested gateways are detected, we update the indices of the exclusive gateway paths accordingly. As demonstrated, a key aspect of our proposed method is identifying and tracking indices, which mark the beginning and end points of various elements within the process description.

3.5. Classification of process information entities

Continuing with the proposed method, if the text contains *process information* entities, we further classify them into several possible subcategories. This classification step is crucial for accurately capturing the flow and structure of the process model. To achieve this classification, we employ the pre-trained BART model from Hugging Face. The *bart-large-mnli* model is a highly popular pre-trained language model designed for natural language inference (NLI) tasks, making it well-suited for zero-shot text classification. By leveraging the power of this model, we can effectively classify *process information* entities into their respective subcategories, further refining the extracted information from the process description.

3.6. Creation of loops

We then identify loops by examining specific keywords and assign task IDs only to tasks without loop keywords. Tasks with loop keywords are not assigned new IDs; instead, we locate the relevant previous task and add a *go_to* key in place of the current agent-task pair. We use the GPT-3.5-Turbo model to determine which task the current description refers to, then apply fuzzy string matching to compare the output to all preceding tasks. We identify the task with the highest similarity as the "previous task," obtain its ID, and add the *go_to* field to the "current" task, successfully creating a loop within our process model.

3.7. Creation of the BPMN structure and diagram

Finally, we create the BPMN structure, representing the process model with its elements and relationships. First, we assign agent-task pairs to corresponding gateways, if present, using gateway start and end indices.

Next, we nest gateways if required, to create a nested JSON-like structure representing the process, which can be parsed for visualization. The BPMN structure is then fed to a module that uses Graphviz to build the diagram (a directed graph). This graph can be rendered as a PDF or other image file types, visually representing the extracted BPMN process model.

4. Evaluation

To assess the effectiveness of our proposed method, we evaluated 31 textual descriptions representative of the types of input that the method was designed to handle. The evaluation process involved testing our method using GPT-3.5-Turbo and GPT-4 to compare their respective performance in generating accurate process models.

To measure accuracy, we utilized a relative graph edit distance (RGED) metric, which is derived from graph edit distance (GED). In our evaluation, we calculated GED using the NetworkX package in Python and compared the generated graph with the target graph, considering all the package's default parameters. To obtain RGED, we used the following formula:

$$RGED(g_1, g_2) = \frac{GED(g_1, g_2)}{GED(g_1, \emptyset) + GED(g_2, \emptyset)} \tag{1}$$

In this formula, g_1 and g_2 represent the generated and target graphs, respectively, and $\emptyset$ represents an empty graph. RGED normalizes GED by dividing it by the sum of GED for both graphs when compared individually with an empty graph.

Once we have the RGED value, we subtract it from 1 to compute a metric that reflects the correctness of the generated process models, similar to the concept of accuracy. With this approach, GPT-3.5-Turbo achieved an average accuracy of 80%, while GPT-4 reached a higher average accuracy of 96%.

These findings demonstrate the potential of our method in extracting accurate and comprehensible BPMN models from textual descriptions. GPT-4, in particular, yielded promising results, showcasing the advantage of leveraging state-of-the-art language models in information extraction and process model generation. However, it is important to note that the evaluation was limited to a specific set of textual descriptions, which may not fully capture the range of real-world applications and potential challenges the method may encounter.

The extracted BPMN models for the online exam and product development processes are shown in Fig. 4. These models were generated using the textual process descriptions presented in Appendix A.

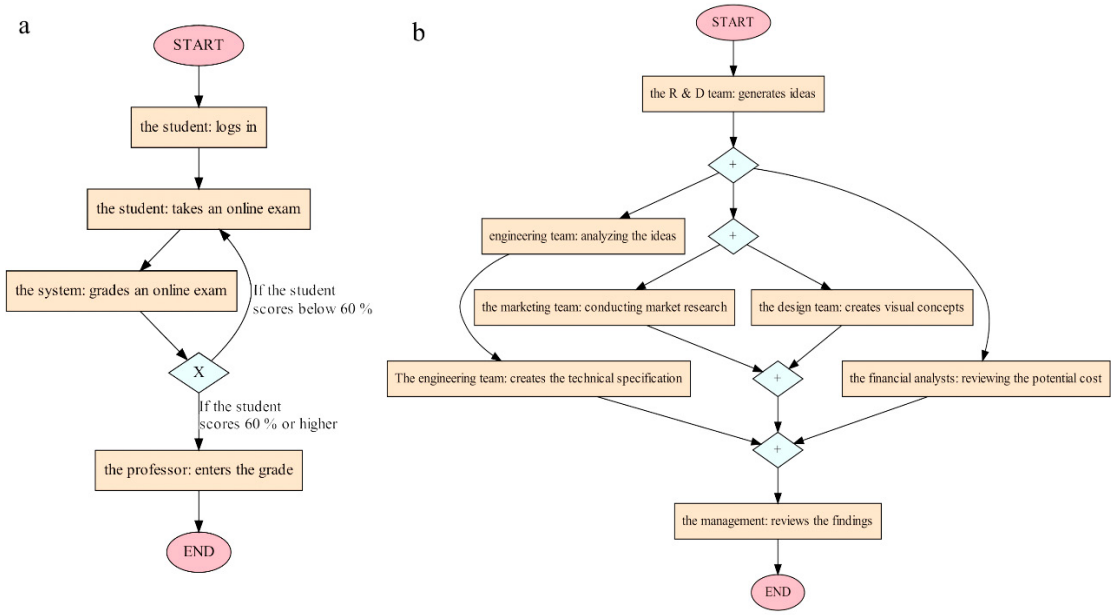

Fig. 4. (a) The online exam process; (b) The product development process.

5. Limitations

One of the limitations of our proposed method lies in the potential for faulty outputs by the BERT model or the OpenAI models, which despite their remarkable performance, are imperfect. Consequently, any inaccuracies in the outputs of these models could negatively impact the quality of the extracted process model.

Another limitation is the dependence of our method on the quality of the input text. Our method assumes that the input text is well-formed, adhering to consistent syntax and grammar. In real-world scenarios, however, textual descriptions may need to be more clear or well written, which could lead to inaccuracies in the extracted process model.

Lastly, our method currently supports the visualization of a limited set of BPMN elements. It does not cover more complex elements, including inclusive gateways and various events. This limitation may restrict the applicability of our method to a certain range of business processes.

In light of these limitations, several potential directions for future research emerge. One direction involves extending our method to support additional BPMN elements, such as inclusive gateways and various event types. This expansion could increase the method's applicability to a broader range of business processes and enhance its practical utility. Another direction for future research is investigating different ways of prompting the OpenAI models. Prompt engineering, which entails crafting effective prompts to guide the model's responses, could be a valuable avenue to explore. By experimenting with various prompt designs and structures, researchers can enhance the performance of these models when extracting information from textual descriptions of business processes.

6. Conclusion

This paper presents a novel approach to extracting BPMN models from textual descriptions using NLP techniques and deep learning models, such as GPT-3.5-Turbo, GPT-4, and a fine-tuned BERT model. Our method generates accurate and understandable process models with minimal manual intervention.

The practical implications of our research are noteworthy. For instance, the proposed method can streamline the process of creating and refining BPMN models for various users, regardless of their expertise level. It can also be employed as an intermediate step for defining formal models or as an educational aid for students studying BPMN.

Ultimately, our approach can enhance communication and collaboration between business and IT stakeholders during the analysis and design of complex processes.

Nevertheless, limitations exist, such as occasional inaccuracies, dependence on well-formed input text, and support for limited BPMN elements. Future research could address these limitations by expanding the range of supported BPMN elements and enhancing prompt engineering for large language models.

Appendix A. Textual process descriptions

A.1. The online exam process

The process begins when the student logs in to the university's website. He then takes an online exam. After that, the system grades it. If the student scores below 60%, he takes the exam again. If the student scores 60% or higher on the exam, the professor enters the grade.

A.2. The product development process

The process starts when the R&D team generates ideas for new products. At this point, 3 things occur in parallel: the first thing is the engineering team analyzing the ideas for feasibility. The engineering team also creates the technical specification. The second path involves the marketing team conducting market research for the ideas. At the same time, the design team creates visual concepts for the potential products. The third path sees the financial analysts reviewing the potential cost of the ideas. Once each track has completed its analysis, the management reviews the findings of the analysis.

References

[1] Friedrich F, Mendling J, Puhlmann F. (2011) “Process Model Generation from Natural Language Text”. In: Mouratidis H, Rolland C, editors. Advanced Information Systems Engineering, Berlin, Heidelberg: Springer; p. 482–96.
[2] Honkisz K, Kluza K, Wiśniewski P. (2018) “A Concept for Generating Business Process Models from Natural Language Description”, p. 91–103.
[3] Sholiq S, Sarno R, Astuti ES. (2022) “Generating BPMN diagram from textual requirements”. Journal of King Saud University - Computer and Information Sciences; 34:10079–93.
[4] Schumacher P, Minor M. (2014) “Extracting control-flow from text”. Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014), p. 203–10.
[5] Ferreira RCB, Thom LH, Fantinato M. (2017) “A Semi-automatic Approach to Identify Business Process Elements in Natural Language Texts”: Proceedings of the 19th International Conference on Enterprise Information Systems, Porto, Portugal: SCITEPRESS - Science and Technology Publications; p. 250–61.
[6] van der Aa H, Di Ciccio C, Leopold H, Reijers HA. (2019) “Extracting Declarative Process Models from Natural Language”. In: Giorgini P, Weber B, editors. Advanced Information Systems Engineering, Cham: Springer International Publishing; p. 365–82.
[7] Quishpi L, Carmona J, Padró L. (2020) “Extracting Annotations from Textual Descriptions of Processes”. In: Fahland D, Ghidini C, Becker J, Dumas M, editors. Business Process Management, Cham: Springer International Publishing; p. 184–201.
[8] Nasiri S, Adadi A, Lahmer M. (2023) “Automatic generation of business process models from user stories”. International Journal of Electrical and Computer Engineering (IJECE); 13:809–22.
[9] Leopold H, van der Aa H, Reijers HA. (2018) “Identifying Candidate Tasks for Robotic Process Automation in Textual Process Descriptions”. In: Gulden J, Reinhartz-Berger I, Schmidt R, Guerreiro S, Guédria W, Bera P, editors. Enterprise, Business-Process and Information Systems Modeling, Cham: Springer International Publishing; p. 67–81.
[10] Qian C, Wen L, Kumar A, Lin L, Lin L, Zong Z, et al. (2020) “An Approach for Process Model Extraction by Multi-grained Text Classification”. Advanced Information Systems Engineering;12127:268–82.
[11] Pyrtek M, Hake1 P, Loos and P. (2021) “Using Artificial Neural Networks to Derive Process Model Activity Labels from Process Descriptions”
[12] Bellan P, Dragoni M, Ghidini C. (2021) “Process Extraction from Text: state of the art and challenges for the future”
[13] Bellan P, Dragoni M, Ghidini C. (2022) “Leveraging pre-trained language models for conversational information seeking from text”
[14] Bellan P, van der Aa H, Dragoni M, Ghidini C, Ponzetto SP. (2023) “PET: An Annotated Dataset for Process Extraction from Natural Language Text Tasks”. In: Cabanillas C, Garmann-Johnsen NF, Koschmider A, editors. Business Process Management Workshops, Cham: Springer International Publishing; p. 315–21.
[15] Sonbol R, Rebdawi G, Ghneim N. (2022) “A Machine Translation Like Approach to Generate Business Process Model From Textual Description”
[16] Ivanchikj A, Serbout S, Pautasso C. (2020) “From text to visual BPMN process models: design and evaluation”. Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, Virtual Event Canada: ACM; p. 229–39.