Unveiling Business Processes Control-Flow: Automated Extraction of Entities and Constraint Relations from Text

Diogo de Santana Candido[1,2][a], Hilário Tomaz Alves de Oliveira[1][b] and Mateus Barcellos Costa[1][c]

[1]*Postgraduate Program in Applied Computing (PPComp), Instituto Federal do Espírito Santo, Serra, Brazil*

[2]*Senado Federal, Brasília, Brazil*

Keywords: Business Process Modeling, Natural Language Processing, Named Entity Recognition, Relation Classification, Machine Learning.

Abstract: Business process models have increasingly been recognized as critical artifacts for organizations. However, process modeling, i.e., the act of creating accurate and meaningful models, remains a significant challenge. As a result, many processes continue to be informally described using natural language text, leading to ambiguities and hindering precise modeling. To address these issues, more formalized models are typically developed manually, a task that requires substantial time and effort. This study proposes a transcription approach that leverages Natural Language Processing (NLP) techniques for the preliminary extraction of entities and constraint relations. A dataset comprising 133 documents annotated with 5,395 expert labels was utilized to evaluate the effectiveness of the proposed method. The experiments focused on two primary tasks: Named Entity Recognition (NER) and relation classification. For NER, the BiLSTM-CRF model, enhanced with Glove and Flair embeddings, delivered the best performance. In the relation classification task, the RoBERTa$_{Large}$ model achieved superior results, particularly in managing complex dependencies. These findings highlight the potential of NLP techniques to automate and enhance business process modeling.

1 INTRODUCTION

Textual descriptions of business processes are widely utilized across diverse domains, including regulation, engineering, healthcare, and education. Such widespread adoption can be attributed to various advantages, including their accessibility, which facilitates understanding for non-specialist audiences (van der Aa et al., 2018). However, the inherent ambiguity of natural language descriptions poses substantial challenges for organizations, including misunderstandings, lack of optimization, undesired flexibility, and standardization issues in process execution. This scenario highlights the need for more formalized models that mitigate ambiguity while preserving key quality attributes, such as legibility, clarity, and consistency with business requirements.

Business Process Management (BPM) provides a conceptual and technical framework for improving organizational activities through the identification, analysis, and monitoring of processes (Grohs et al., 2023). Within BPM, the critical discipline of process discovery draws upon various sources of information, including observations, event logs, interviews, and document analysis (Dymora et al., 2019). Among these sources, textual documents stand out for their information richness, yet they present significant challenges due to their unstructured nature and inherent complexity (Bellan et al., 2020).

Analyzing natural language descriptions for process modeling requires addressing aspects from both the syntactic, semantic, and pragmatic levels of information representation while managing ambiguities and incomplete structures (Bellan et al., 2022b). Additionally, identifying constraints and dependencies within processes often relies on domain experts who may lack modeling expertise (Van der Aa et al., 2019).

This work introduces an approach that leverages Natural Language Processing (NLP) and Machine Learning (ML) techniques to extract core entities and relationships from textual business process descriptions. By emphasizing declarative modeling, this approach simplifies process representation, making it

[a] https://orcid.org/0009-0000-2733-2046
[b] https://orcid.org/0000-0003-0643-7206
[c] https://orcid.org/0000-0002-4235-5411

Candido, D. S., Alves de Oliveira, H. T. and Costa, M. B.
Unveiling Business Processes Control-Flow: Automated Extraction of Entities and Constraint Relations from Text.
DOI: 10.5220/0013435600003929
In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 2*, pages 771-782
ISBN: 978-989-758-749-8; ISSN: 2184-4992

easier to understand and model efficiently. Following a minimalist strategy, it prioritizes key components, such as participants and dependence relations, to enhance adaptability across various contexts and ensure consistency in process documentation (Obendorf, 2009).

The proposed methodology employs Named Entity Recognition (NER) to identify essential entities and employs relation classification to analyze dependencies between them. The resulting declarative models provide a clear representation of process constraints, serving as a bridge toward the automated generation of imperative models, thereby reducing ambiguities and improving efficiency.

The structure and organization of this article are as follows: Section 2 provides foundational concepts crucial to understanding the proposed database. Section 3 reviews prior studies and approaches relevant to the domain of business process modeling from textual data. Section 4 outlines the methodology employed to construct the dataset, detailing data collection, annotation, and preprocessing steps. Section 5 presents and analyzes the experimental results for the NER and Relation Classification tasks, offering insights into the performance and challenges encountered. Section 7 demonstrates the practical application of the developed model by generating declarative and BPMN models from textual data, showcasing its utility in real-world scenarios. Lastly, Section 8 summarizes the key findings of this study, discusses its contributions to the field, and highlights potential avenues for future research.

2 BACKGROUND

2.1 Business Process Control-Flow Modeling

The control-flow perspective refers to the formal specification of dependencies that govern the sequence of activities within a process (Fionda and Guzzo, 2020). Therefore, its modeling provides the specification of rules that determine the behavior of the process or its operational semantics. Languages and notations for such purposes can be categorized as declarative or constraint-based and procedural or imperative. While declarative notations aim to provide boundary constraints for process execution, procedural notations aim to establish the specific traces (execution paths) that the process should follow. Procedural notations are almost all inspired by Process Algebra, whose operators enable the specification of order (.), choice (+), and parallelism (|). The following sentence illustrates the use of Process Algebra to describe a simple Order Processing model:

$$\begin{aligned} Proc = {} & \text{“New Order”}.\text{“Register Order”}.\text{“Check Stock”}. \\ & \left(\begin{array}{l} (\text{“Product Available”}.(\text{“Send Product”}|\text{“Charge”})) \\ + (\text{“Product Not Available”}.\text{“Cancel Order”}) \end{array} \right) \end{aligned}$$

Declarative process languages and notations are inspired by logic-based languages, such as Linear Temporal Logic (LTL) (Fionda and Guzzo, 2020). LTL is based on the formulation of sentences that logically and temporally constrain the behavior of the considered variables. In this manner, it is possible to impose the condition of the existence (execution) of activities based on the existence of other activities. The language incorporates boolean operators and temporal modal operators such as X for next, G for always (globally), F for finally, R for release, W for weak until, and M for mighty release. A process can be modeled with LTL by creating a set of LTL sentences. The following sentences describe the Order Processing in terms of LTL.

- New Order $\rightarrow$ $\mathbf{X}$(Register Order)
- Register Order $\rightarrow$ $\mathbf{X}$(Check Stock)
- Check Stock $\rightarrow$ $\mathbf{X}$(Product Available $\vee$ Product Not Available)
- Product Available $\rightarrow$ $\mathbf{X}$(Send Product $\vee$ Charge)
- Product Not Available $\rightarrow$ $\mathbf{X}$(Cancel Order)
- $\neg$(Send Product $\wedge$ Cancel Order)
- $\neg$(Charge $\wedge$ Cancel Order)

Note that, unlike the procedural approach, the LTL sentences may be unordered and not related to each other. In this way, such models have the advantage of being able to be constructed in a fragmented manner, for instance, at different times or in different organizational spaces. They also have the disadvantage of being susceptible to consistency errors. This latter aspect has led to the need for the application of verification methods, such as model checking, in declarative languages (Schützenmeier et al., 2021). It can be said that the process description in Process Algebra is compliant with the LTL specification. Furthermore, other descriptions in Process Algebra can meet this specification.

2.2 Situation-Based Modeling Notation

Given that declarative models can effectively capture the essential constraints of a process and serve as a foundation for deriving specific procedural models, this work explores their use in extracting process models from textual descriptions. The chosen notation for this purpose is the Situation-Based Modeling Notation (SBMN) which is originally discussed by Costa and Tamzalit (Costa and Tamzalit, 2017) in the context of business process modeling recommendation patterns. In SBMN, the concept of a situation plays a central role in designing the execution logic according to the process objectives. Formally, a situation is defined as a binary relation in which the operands are sets of Active Flow Objects (AFOs) within the process. The set of AFO flows in a process corresponds to the set of its elements representing activities or events.

The proposed situation catalog considered a few sets of situation types that have proven enough to represent the control flow of linear processes. For this work, it was considered the following set of types:

Dependence. A set of AFOs with a temporal execution dependence among them. Dependencies are sub-classified into two types: Strict ($\lhd$, DEP) and Circumstantial ($\unlhd$, DEPC). In a strict dependence relation, if *b* depends on *a*, then *b* can be executed in a flow only, and only if *a* has been executed before. In a circumstantial dependence relation, if *b* depends on *a*, then *b* can be executed in a flow where *a* was executed before or in a flow where *a* is not executed any time.

Non-coexistence. A set of flow objects with a non-coexistence relation ($\otimes$, XOR)at the same execution flow, generally mapped to a XOR relation in procedural notations.

Union. A set of AFOs with a union relation ($\oplus$, UNI) at the same execution flow.

Situations can also use the *or* logical connector ($\vee$) to relate active flow objects. For example, the dependence situation $c \lhd (a \vee b)$ indicates that *c* depends on *a* or *b*.

2.3 Natural Language Processing for Business Process Modeling

Natural Language Processing (NLP) focuses on enabling machines to understand, interpret, and generate human language. Among its key tasks, relation extraction (RE) aims to identify and classify semantic relationships between entities mentioned in unstructured text. This task plays a critical role in transforming raw textual data into structured information, enabling applications such as knowledge graph construction, question answering, and information retrieval.

The foundation for RE was established during the Message Understanding Conferences (MUC), a series of events organized by the U.S. Defense Advanced Research Projects Agency (DARPA) from 1987 to 1998. These conferences presented practical challenges, encouraging researchers to develop systems capable of extracting specific information from text, such as entities, events, and relationships (Detroja et al., 2023). The seventh edition (MUC-7) formalized RE as a distinct task, contributing to the creation of annotated corpora and benchmarks that continue to guide research in this area (Chinchor, 1998).

RE typically integrates Named Entity Recognition (NER) and relation classification. Two main approaches to RE are widely used: pipeline and joint methodologies. The pipeline approach processes NER and relation classification sequentially, while the joint approach combines both tasks within a unified framework. The latter reduces error propagation and captures dependencies between entities and their relationships more effectively, making RE a crucial step in structuring unstructured data into actionable insights (Zhao et al., 2024).

3 RELATED WORK

The extraction of business process models from textual descriptions continues to grow as a crucial research area. This field addresses the challenge of translating the inherent ambiguities of natural language into formalized models, thereby enhancing accessibility, precision, and scalability. Current methodologies range from traditional rule-based systems to cutting-edge machine learning and NLP techniques, with Pre-trained Language Nodels (PLMs) gaining prominence.

A seminal contribution by (Bellan et al., 2023) provides a comprehensive overview of the domain, tracing the evolution of methods for extracting process models from text. Their survey highlights the predominance of rule-based approaches, frequently paired with machine learning models for tasks such as NER and relation classification. These traditional methods set the foundation for process extraction but faced challenges in generalization and scalability. Recent advancements, particularly PLMs and deep learning frameworks, have significantly improved the ability to handle linguistic complexity and domain adaptability.

Declarative process models have garnered significant attention due to their flexibility. For instance, (López et al., 2021) introduced a hybrid approach combining rule-based methods with BERT-based NER models to extract Dynamic Condition Response (DCR) graphs. The proposed model achieved an f1-score of 0.71, demonstrating the potential of deep learning integrated with domain-specific rules. Similarly, (Van der Aa et al., 2019) developed a method for extracting declarative constraints using linguistic normalization, semantic parsing, and rule-based templates. Despite high precision in identifying activities and dependencies, this approach struggled with complex relationships and domain generalization.

Imperative models like BPMN have also been extensively explored. Studies by (Friedrich et al., 2011) and (Honkisz et al., 2018) developed two-step pipelines that combined syntactic and semantic analysis to transform textual descriptions into structured models. While demonstrating potential in specific use cases, these methods often lacked large-scale evaluations and faced difficulties in handling ambiguous or incomplete inputs.

More recently, Large Language Models (LLMs) such as GPT-4 and Gemini have revolutionized the extraction of process models by enabling the direct generation of BPMN and declarative models from text. Studies by (Grohs et al., 2023) and (Kourani et al., 2024) highlighted the promise of LLMs in automating process modeling through advanced prompt engineering. However, challenges remain, including dependence on high-quality prompts and limited control over generated outputs. Similarly, (Bellan et al., 2022a) demonstrated the adaptability of GPT-3 and in-context learning for process element and relationship extraction but noted difficulties in capturing complex control-flow relationships.

Datasets have played a vital role in advancing this field. (Bellan et al., 2022b) introduced the PET dataset, annotated for BPMN element extraction, providing a benchmark for process extraction methodologies. This dataset comprises manually annotated entities and relations such as activities, actors, and dependencies, facilitating the evaluation of various extraction techniques. Additionally, (Ackermann et al., 2021) proposed UCCA4BPM, leveraging semantic parsing and graph neural networks for process annotation, underscoring the importance of annotated corpora in this domain.

Several studies have further expanded the scope of process model extraction. For instance, (Qian et al., 2020) employed a neural network-based approach for sentence classification in procedural texts, achieving high precision in distinguishing activity-related sentences. (Ferreira et al., 2017) utilized rule-based methods for extracting BPMN elements, demonstrating the effectiveness of syntactic analysis combined with predefined patterns. Similarly, (Epure et al., 2015) developed methods for extracting models from domain-specific texts such as archaeological reports, highlighting the importance of contextual adaptation.

Despite these advancements, key challenges persist. Many approaches rely heavily on BPMN and declarative notations, often resulting in partial representations that fail to capture hierarchical structures and interdependencies comprehensively. Moreover, the scarcity of large, high-quality datasets limits the application of data-intensive techniques such as PLMs. Addressing these challenges will require improved dataset creation, hybrid methods combining rule-based systems with PLMs, and scalable architectures that focus on extracting essential entities and relations.

This study builds on prior work by introducing a simplified yet robust methodology that prioritizes key entities and relationships in business process descriptions. By leveraging state-of-the-art techniques, this research aims to bridge the gap between traditional and modern approaches, contributing to the advancement of automated business process modeling.

4 DATASET CONSTRUCTION

This section presents the steps carried out to construct the dataset used in the experiments performed in this work.

4.1 Data Acquisition

Publicly annotated datasets for business process modeling are scarce. Existing collections are often derived from the dataset introduced by (Friedrich et al., 2011). Another dataset, proposed by (Qian et al., 2020), includes instruction manuals and food recipes, but these lack the complexity of typical business processes.

To address this, we systematically collected English-language texts describing business processes. The primary source was the work of (Friedrich et al., 2011), which is widely used in research on extracting information from natural language business process descriptions. Additional texts were sourced from:

- **Klievtsova et al. (2023):** 24 newly proposed business process descriptions (Klievtsova et al., 2023).
- **Dumas et al. (2018):** 48 texts of exercises and examples from (Dumas et al., 2018).
- **Class Exercises:** 16 descriptions collected from an undergraduate course on business process modeling.

Table 1 summarizes the sources, resulting in a dataset of 133 texts.

Table 1: Sources of Texts in the Dataset.

Source	Number of Texts
Friedrich et al. (2011)	45
Klievtsova et al. (2023)	24
Dumas et al. (2018)	48
Class Exercises	16
Total	**133**

4.2 Annotation

The dataset annotation was based on SMN (see Section 2.2), identifying logical-temporal relations such as strict dependence, circumstantial dependence, union, and non-coexistence. Relationships of responsibility were also included when the actor entity was introduced.

Flow objects were categorized into activities, triggers, and catches. A conditional entity was defined to represent conditions linked to circumstantial dependencies or non-coexistence. Tables 2 and 3 summarize the named entities and relation types considered during the annotation process.

Table 2: Named entity types used in annotation.

Name	Description
Actor	Responsible for actions in the process.
Activity	Tasks or operations in the process.
Trigger	Events that start a process.
Catch	Events that capture conditions.
Conditional	Conditions tied to dependencies.

Table 3: Relation types considered in annotation.

Name	Description
Strict Dependence	Logical dependencies.
Circumstantial Dependence	Condition-based dependencies.
Union	Merging of entities or actions.
Non-Coexistence	Entities that cannot coexist.
Perform	Actor responsibility for an activity.

Two human experts performed the annotation using the Doccano tool[1] to ensure consistency and reliability. A detailed annotation guide was created to standardize procedures and ensure reproducibility.

[1]https://github.com/doccano/doccano

To illustrate the annotation, consider the following purchase order process: *"When a new product is requested, the supplier checks the inventory. If available, the product is labeled and sent to the dock. Otherwise, it is ordered from the manufacturer, and a delayed delivery notice is issued to the requester."*. The annotated entities and relations are shown in Figure 1.

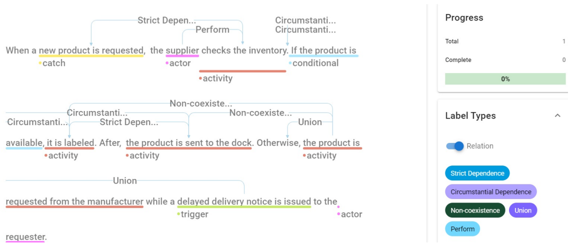

Figure 1: Example of an annotation of a purchase order process.

At the end of the annotation process, 1,361 sentences were analyzed. Table 4 summarizes the annotated entities and relations.

Table 4: Annotated Entities and Relations: Summary.

Type	Category	Count	% of Total
Entities	Actor	1,279	36.4%
	Activity	1,297	36.9%
	Trigger	234	6.7%
	Catch	233	6.6%
	Conditional	474	13.5%
Relations	Strict Dependence	553	29.4%
	Circumstantial Dependence	424	22.6%
	Perform	773	41.2%
	Union	69	3.7%
	Non-Coexistence	59	3.1%
Total		**5,395**	**100%**

The annotation process yielded a total of 5,395 annotations, divided between entities and relations. The most frequently annotated entities were *Actor* (1,279) and *Activity* (1,297), collectively representing nearly half of the annotations. Among relations, *Perform* (773) and *Strict Dependence* (553) were most common, reflecting the dataset's emphasis on capturing key roles, actions, and dependencies within business processes.

On the other hand, categories such as *Trigger* (234) and *Catch* (233) were less frequently annotated, indicating either limited representation in the dataset or lower priority during the annotation process. Similarly, the relations *Union* (69) and *Non-Coexistence* (59) appeared infrequently, suggesting their lower relevance within the annotated corpus. These patterns highlight the dataset's focus areas while pointing to potential gaps for future improvement.

4.3 Divergence and Problems Resolutions

After the annotation process, a more experienced evaluator was responsible for evaluating each of the annotations made. This process was based on the criteria proposed by Yuan et al. (Yuan et al., 2021), which were added with the criteria Absence and Ambiguity, which respectively identify the absence of annotation and the presence of ambiguity in the annotation. Based on these criteria, the annotations were analyzed to identify the following types of problems: Absence, Ambiguity, Factual Plausibility, Appropriateness, and Formatting. Table 5 describes each type of annotation problem that is searched in the curating process.

Table 5: Types of problems inspected in annotations.

Type of Problem	Description
Absence	Missing expected annotations.
Ambiguity	Unclear or multiple interpretations.
Factual Plausibility	Conflicts with known facts or logic.
Appropriateness	Unsuitable or irrelevant annotations.
Formatting	Structural or formatting issues.

5 EXPERIMENTAL METHODOLOGY

Two experiments were performed to evaluate the proposed approach and dataset. The first experiment focused on evaluating different NER approaches to identify entities of interest. In the second experiment, different classification algorithms based on the BERT model were evaluated to identify the relationships annotated in the database.

5.1 Named Entity Recognition

Named Entity Recognition (NER) is a fundamental NLP technique used to identify and classify entities such as activities, actors, triggers, and conditions in business process descriptions. Conditional Random Fields (CRF) is a widely used method for modeling the sequential nature of text and predicting entity labels based on context (Li et al., 2020).

To enhance the semantic representation of words, embeddings like GloVe, Flair, and BERT were employed. GloVe provides context-independent word vectors based on co-occurrence statistics (Pennington et al., 2014), while Flair embeddings offer contextual representations by incorporating surrounding text (Akbik et al., 2019). BERT further improves this by generating deep contextualized embeddings using a transformer-based architecture (Devlin et al., 2018). The BiLSTM-CRF architecture combines BiLSTM layers to capture temporal dependencies with a CRF layer for structured prediction, leveraging both past and future contexts for precise entity extraction (Lample et al., 2016).

5.2 Relation Classification

Relation classification is a pivotal task in transforming unstructured textual descriptions into structured representations by identifying semantic relationships between entities. This process involves categorizing relationships such as "non-coexistence" and "dependencies", utilizing contextual information embedded in the text. Document-level annotations were employed to effectively capture both local and long-range dependencies, ensuring that relationships spanning multiple sections of a text were accurately identified.

To construct the dataset, positive and negative examples were systematically generated from annotated entity pairs. Positive examples consisted of entity pairs with valid relationships explicitly annotated in the dataset, each labeled with the corresponding relationship type. Negative examples, labeled as "O", represented entity pairs without valid relationships. This distinction enabled the dataset to comprehensively capture both the presence and absence of relationships, ensuring a balanced and diverse analysis.

For the relation classification task, examples were generated following these guidelines:

- **Positive Examples:** For entity pairs (e_i, e_j) with valid relationships $(e_i, e_j) \in R$, positive examples were created. These represent true connections explicitly annotated in the dataset. For instance:

  In the sentence "Einstein developed the theory", the relationship $(Einstein, perform, developed\ the\ theory)$ was labeled as positive.

- **Negative Examples:** For entity pairs (e_i, e_j) where $(e_i, e_j) \notin R$, negative examples were generated. These pairs lack valid relationships and were assigned the label "O". This represents the absence of a connection between entities. For example:

  In a dataset with the entities "passengers" and "boarding pass", the relationship

$(passengers, O, boarding\ pass)$ would be labeled as negative if no valid relationship exists.

- **Dataset Balance:** To prevent an overrepresentation of negative examples—which could bias the model during training—the total number of negative examples was limited to approximately match the number of positive examples. This balance ensured that the model received an unbiased mix of both types, facilitating effective training and evaluation.

Table 6 presents some examples of texts containing the annotations made to enable the training and testing of the relationship classification models in this work.

Table 6: Examples of positive and negative relationships between entities.

Type	Text and Relationship
Positive	Once the boarding pass has been received, [E1]passengers[/E1] [E2]proceed to the security check[/E2]. Here, they need to pass the personal security screening and the luggage screening. Afterwards, they can proceed to the departure level. (perform)
Negative	Once the boarding pass has been received, [E1]passengers[/E1] proceed to the security check. Here, they need to pass the personal security screening and the luggage screening. Afterwards, they can [E2]proceed to the departure level[/E2]. (O)
Negative	Once the boarding pass has been received, passengers [E1]proceed to the security check[/E1]. Here, they need to pass the personal security screening and the luggage screening. Afterward, [E2]they[/E2] can proceed to the departure level. (O)

The final dataset comprised 3,889 examples, with 1,878 positive and 2,011 negative instances. To ensure robust training and evaluation, a 5-fold cross-validation methodology was applied. Document-level splitting was performed to prevent overlap between training and testing data, thereby guaranteeing the integrity and generalization of the evaluation process.

This balanced and comprehensive dataset enabled a detailed analysis of the model's ability to distinguish between valid and invalid relationships, providing a solid foundation for evaluating the effectiveness of relation classification methodologies.

5.3 Experimental Setup

The experiments performed for the NER and the relation classification aimed to evaluate different approaches for these tasks. The codes used in the experiments are available in a GitHub repository[2]. The 133 annotated documents were converted to CoNLL format and divided using 5-fold cross-validation to ensure that sentences from the same document were in the same subset.

Three approaches were evaluated for the NER task. The first approach used CRF, relying on linguistic features such as part-of-speech tags and capitalization. The $sklearn\_crfsuite$[3] library was employed, and hyperparameters, including regularization coefficients (c_1 and c_2), were optimized for f1-score.

The second approach utilized a BiLSTM-CRF architecture, combining BiLSTM layers with a CRF layer for structured prediction. Word embeddings such as GloVe, Flair, BERT, and DistilBERT were evaluated individually and in combination. Training, performed using the Flair framework[4], employed a batch size of 32, a learning rate of 0.1, and early stopping after 10 epochs without improvement, continuing for up to 100 epochs.

The third approach fine-tuned transformer-based models, including DistilBERT, BERT (Base and Large), and RoBERTa, using the transformers library[5]. Training was conducted for up to 100 epochs with a batch size of 32, a learning rate of 2×10^{-5}, and a weight decay of 0.01, with early stopping after 20 epochs of no improvement.

For the relation classification task, transformer-based models, including DistilBERT, BERT (Base and Large), and RoBERTa, were employed. These models are particularly effective due to their ability to capture bidirectional contextual information, which is crucial for identifying nuanced relationships in text. The training process involved fine-tuning pre-trained models using the transformers library. A maximum of 20 epochs was set, with early stopping applied after five epochs without improvement. The AdamW optimizer was used with a learning rate of 5×10^{-5} and a batch size of 16. The models were evaluated using the macro f1-score to ensure balanced performance across all classes.

[2]https://github.com/laicsiifes/bpm_dataset
[3]https://sklearn-crfsuite.readthedocs.io/en/latest/
[4]https://github.com/flairNLP/flair
[5]https://huggingface.co/docs/transformers/index

6 RESULTS AND DISCUSSION

Table 7 presents the experimental results of the NER task based on the micro average f1-score evaluation measure computed at the complete entity level (exact match). An extraction is only considered correct if all the words that form the entity are identified.

Table 7: Results of the experiments for the NER task using the *f1-score* micro metric.

Mod./Lab.	Activity	Actor	Catch	Condition	Trigger	Micro Avg.
C	0.377	0.723	0.099	0.664	0.241	0.522
Bb	0.431	0.784	0.120	0.602	0.225	0.543
Bl	0.435	0.805	0.147	0.594	0.209	0.548
Di	0.414	0.776	0.111	0.578	0.204	0.526
Rb	0.476	0.810	0.192	0.659	0.229	0.584
Rl	0.470	0.804	0.178	0.638	0.251	0.580
G	0.427	0.750	0.077	0.613	0.249	0.553
F	0.487	0.801	0.130	0.688	0.281	0.606
D	0.457	0.791	0.157	0.625	0.283	0.577
G+F	**0.498**	**0.816**	0.196	**0.692**	**0.289**	**0.617**
G+D	0.462	0.796	0.173	0.625	0.253	0.577
G+B	0.472	0.800	**0.197**	0.658	0.271	0.583
G+F+B	0.489	0.805	0.172	0.680	0.309	0.604
G+F+D	0.424	0.762	0.105	0.586	0.176	0.540

Legend:

- **C**: CRF (Conditional Random Fields).
- **Bb**: BERT Base.
- **Bl**: BERT Large.
- **Di**: DistilBERT.
- **Rb**: RoBERTa Base.
- **Rl**: RoBERTa Large.
- **G**: BiLSTM-CRF + Glove embeddings.
- **F**: BiLSTM-CRF + Flair embeddings.
- **D**: BiLSTM-CRF + DistilBERT embeddings.
- **G+F**: BiLSTM-CRF + (Glove/Flair).
- **G+D**: BiLSTM-CRF + (Glove/DistilBERT).
- **G+B**: BiLSTM-CRF + (Glove/BERTBase).
- **G+F+B**: BiLSTM-CRF + (Glove/Flair/BERTBase).
- **G+F+D**: BiLSTM-CRF + (Glove/Flair/DistilBERT).

The analysis of NER results highlights the strengths and limitations of the evaluated models. The Actor entity consistently achieved f1-scores higher than 0.7 across all models, benefiting from its shorter length and high frequency. In contrast, Catch and Trigger entities demonstrated the lowest scores due to their infrequent annotations and semantic overlap with other entities, indicating the need for more balanced dataset.

BiLSTM-CRF models combined with Glove and Flair embeddings (G+F) outperformed others, achieving the highest micro-average f1-score of 0.617. The RoBERTa model showed competitive results, particularly for entities requiring complex contextual understanding. However, challenges such as misclassification of boundary tokens and semantic confusion between overlapping entity types, such as Catch, Trigger, and Activity, were observed.

Seeking a better understanding of the model results, Table 8 presents the size statistics of the annotated entities. By analyzing the results and statistics of the annotated entity sizes, relationships emerge between the number of annotations, the average size, and the complexity of the entities with the f1-score results, revealing how these variables influence the performance of the models.

Table 8: Statistics of annotated entity sizes (measured in characters).

Entity	Mean	Standard Deviation	Maximum	Minimum
Trigger	28.71	17.50	117	5
Condition	36.00	19.36	129	4
Activity	31.14	16.70	149	3
Actor	9.66	8.04	50	1
Catch	29.86	21.10	144	4

Overall, integrating multiple embedding techniques enhances model performance, and transformer models hold promise for further improvements with more balanced and enriched datasets. These findings underscore the importance of addressing entity-specific challenges to improve NER in business process descriptions.

In Figure 2, the confusion matrix of the BiLSTM-CRF+(Glove/Flair) model illustrates its strengths and limitations. The model excels in the “I-activity” and “O” classes, with high true positive rates. However, “B-activity” often misclassified as “O”, leading to moderate performance for the Activity entity. Similarly, significant confusion among “B-catch”, “B-trigger”, “I-catch”, and “I-trigger” classes impacts the performance of the Catch and Trigger entities.

The “B-actor” and “I-actor” classes show minimal errors, resulting in f1-scores exceeding 0.8. On the other hand, the Condition entity achieves satisfactory results due to its distinct linguistic patterns, despite occasional misclassification with “O”. Overall, challenges arise from semantic overlaps and dataset imbalances, particularly affecting the less frequent entities.

The results of the relation classification task, presented in Table 9, highlight the strengths and limitations of different models in identifying semantic relationships. Among the evaluated models, the RoBERTa architectures, especially $RoBERTa_{Large}$, stood out, achieving the highest macro-average F1-score of 0.770 and excelling in five of the six relationship classes. Such results demonstrate the effectiveness of transformer-based architectures, par-

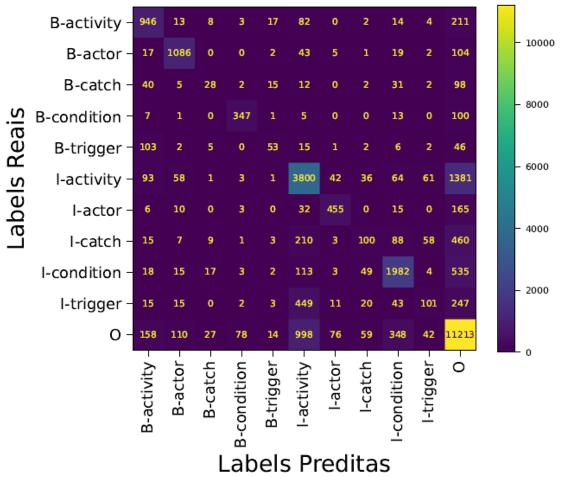

Figure 2: Confusion matrix of the BiLSTM-CRF+(Glove/Flair) model.

ticularly those enhanced with robust pretraining like RoBERTa, in capturing complex relational patterns.

Table 9: Results of Relation Classification experiments using the F1-score measure.

Mod./Lab.	O	Circum. Dep.	Non Coex.	Perform	Strict Dep.	Union
Bb	0.892	0.815	0.648	0.807	0.756	0.152
Bl	0.895	0.830	0.372	0.805	0.763	0.102
D	0.871	0.797	0.547	0.796	0.740	0.171
Rb	0.935	**0.890**	0.696	**0.894**	0.845	0.309
Rl	**0.940**	0.882	**0.710**	0.862	**0.861**	**0.365**

Legend:

- **Bb**: BERT Base.
- **Bl**: BERT Large.
- **D**: DistilBERT.
- **Rb**: RoBERTa Base.
- **Rl**: RoBERTa Large.

The evaluated models consistently performed well in identifying the absence of relationships ("O" class), with f1-score ranging from 0.871 (*DistilBERT*) to 0.940 ($RoBERTa_{Large}$). These results underline the models' ability to correctly classify non-relational instances, which are well-represented and clearly defined in the dataset. Similarly, the "Perform" class exhibited strong performance across most models, achieving a peak f1-score of 0.894 with $RoBERTa_{Base}$, likely due to the high number of annotations (773) and the relatively distinct semantic characteristics of this relationship.

Conversely, the "Union" class posed the greatest challenge, with f1-scores varying from 0.102 ($BERT_{Large}$) to 0.365 ($RoBERTa_{Large}$). The low frequency of annotations (69 examples) hindered the models' ability to generalize, emphasizing the need for more balanced datasets. The "Non-Coexistence" class also exhibited variability, with scores ranging from 0.372 ($BERT_{Large}$) to 0.710 ($RoBERTa_{Large}$), reflecting the semantic ambiguity and contextual dependence of this relationship.

For "Circumstantial Dependence", the best results were achieved by $RoBERTa_{Base}$ (0.890) and $RoBERTa_{Large}$ (0.882), indicating the effectiveness of these models in capturing contextual nuances. Similarly, "Strict Dependence," with 553 annotations, achieved satisfactory performance, with $RoBERTa_{Large}$ reaching an f1-score of 0.861. The larger dataset and the distinctive semantic features of this relationship contributed to its improved classification compared to other classes.

The confusion matrix for $RoBERTa_{Large}$ (Figure 3) provides additional insights into the classification process. The model demonstrated minimal confusion for the "O" class, effectively distinguishing negative examples. However, semantic overlaps between classes, such as "Circumstantial Dependence" and "Perform" led to occasional misclassifications. The "Union" class, with its low representation, exhibited significant confusion, particularly with classes like "Circumstantial Dependence."

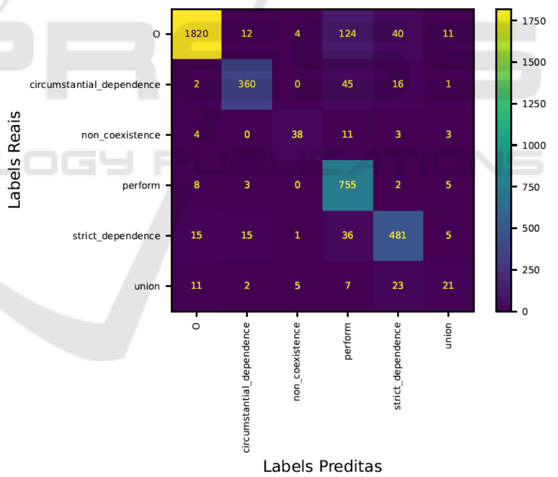

Figure 3: Confusion matrix of the $RoBERTa_{Large}$ model.

In conclusion, the results underscore the effectiveness of advanced transformer models in identifying complex semantic relationships, with $RoBERTa_{Large}$ consistently outperforming other architectures. These findings align with prior research, such as (Grohs et al., 2023) and (Kourani et al., 2024), which highlighted the advantages of LLMs in automating process model extraction. However, the imbalance in class representation significantly impacts the models' ability to generalize, particularly for low-frequency relationships like "Union." This limitation was also observed in previous studies like (Bellan et al., 2022a), where capturing intricate process dependencies proved challenging due to data scarcity. Addressing this imbalance through techniques such as data augmentation and enriched annotation, as suggested by (Bellan et al., 2022b) and (Ackermann et al., 2021), can enhance model robustness and improve performance in underrepresented classes. Furthermore, hybrid approaches, such as the combination of rule-based and machine learning techniques explored in (Qian et al., 2020) and (Epure et al., 2015), could provide promising solutions to mitigate the impact of data imbalance while improving extraction accuracy. These findings provide a foundation for future research to further optimize relation classification in structured datasets.

7 APPLICATION OF THE DEVELOPED MODEL

To evaluate the practical application of the developed model, the text "Blizzard Online Character Generator" was selected as input. This text describes a character creation process for a computer game, detailing user interactions and system responses. The text, categorized under "Computer Games," is presented below:

Input Text: "Blizzard Online Character Generator"

Blizzard creates a cool online tool for creating characters for their new WoW expansion. When creating a World of Warcraft character, you can start doing two things: While you are setting up your account, you can already come up with good character names. The setup of your account starts with checking whether you have a battle.net account. If you do not have one yet, you enter the account information and click the link you receive in the confirmation mail. As soon as you have a battle.net account, you can check if you have an active WoW subscription. If not, you can select the payment method. If you choose credit card, enter your credit card information. If you choose your bank account, enter your IBAN and BIC numbers. After that you can log into the game and select realm, race and class of your character. Until now, you should have come up with some good names. You enter them one by one until a name is still available. You get a confirmation, and some selfies of your character, as soon as an expansion is released you get another message. (Klievtsova et al., 2023)

The text was processed using the BiLSTM-CRF+(Glove/Flair) model to identify and extract entities and relations. The identified activities and their respective labels are listed below:

- **T1:** Start creating the character.
- **T2:** Set up account information.
- **T3:** Enter payment details.
- **T4:** Log into the game.
- **T5:** Come up with character names.
- **T6:** Enter name suggestions.
- **T7:** Receive confirmation and selfies.

The extracted entities and relations were organized into a declarative model. Key dependencies, such as "T2 XOR T3 depends on T1" and "T4 depends on T2 and T3", were established to represent logical and sequential flows. These dependencies were then transformed into an imperative BPMN model, as shown in Figure 4.

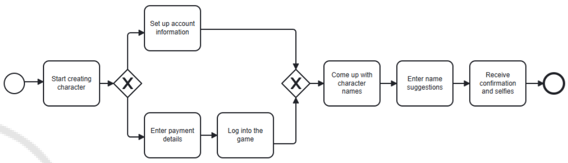

Figure 4: BPMN model generated from the text "Blizzard Online Character Generator".

The BPMN model captures the sequential and conditional flow of activities described in the input text. For instance, activities such as "Log into the game" depend on prior steps like account setup or payment completion, which are represented using exclusive gateways. The integration of conditions, such as "If no account exists yet" or "If a name is available", ensures the BPMN model aligns with the declarative constraints.

The generated BPMN model demonstrates high fidelity to the input text, accurately reflecting the extracted entities and relations. This process highlights the model's capability to transform unstructured text into structured process representations, supporting both declarative and imperative paradigms. The findings validate the utility of the developed approach in automating the modeling of business processes from natural language descriptions.

8 CONCLUSION

This study investigated the application of neural network architectures and machine learning algorithms to extract entities and constraint relations in business processes described in natural language. A structured methodology was developed, consisting of sequential stages: Named Entity Recognition (NER) and relation classification. This approach aimed to identify and categorize the key entities and relationships required for modeling the control-flow perspective of business processes.

As part of this research, a reference dataset was constructed, comprising 133 documents annotated with entities and relations relevant to business process modeling. These annotations, performed by domain experts, resulted in a dataset containing 1,361 sentences and 5,395 annotated elements, including entities and relations. This dataset provides a robust foundation for evaluating the proposed methodologies and benchmarking future advancements in the domain.

Two experiments were conducted to validate the proposed architecture. The first focused on the NER task, applied different models, including CRF, $BERT_{Base}$, $BERT_{Large}$, DistilBERT, $RoBERTa_{Base}$, $RoBERTa_{Large}$, and the BiLSTM-CRF architecture combined with word embeddings from Glove, Flair, DistilBERT, and $BERT_{Base}$. Combinations of these embeddings were also explored. The second experiment addressed the relation classification task, assessing models such as $BERT_{Base}$, $BERT_{Large}$, DistilBERT, $RoBERTa_{Base}$, and $RoBERTa_{Large}$.

The experimental results demonstrated the importance of advanced architectures in extracting entities and relations from natural language descriptions of business processes. For NER, the integration of representations such as Glove and Flair with the BiLSTM-CRF architecture proved highly effective, consistently outperforming BERT- and RoBERTa-based models across multiple entity categories. In the relation classification task, $RoBERTa_{Large}$ emerged as the most robust model, achieving the best overall performance, particularly for complex relationships such as Strict Dependence and Union.

The proposed approach demonstrates the feasibility of extracting entities and relations that capture essential constraints in business processes described in natural language. Despite these promising results, several limitations were identified. The performance of the NER task highlighted significant challenges, reflecting the inherent complexity of entity recognition. Furthermore, the limited size of the dataset and annotations may have impacted the generality and scope of the results.

The aforementioned limitations underscore the need for future work to address these issues, including expanding the corpus, refining annotations, and exploring more advanced models. Specifically, future research will focus on leveraging Large Language Models (LLMs) such as GPT-4 and Gemini for process model extraction. These models have shown significant potential to capture complex contextual dependencies, as highlighted by recent studies (Grohs et al., 2023; Kourani et al., 2024). Furthermore, incorporating semi-supervised learning techniques and active learning strategies will be explored to enhance annotation efficiency and model adaptability in low-resource settings.

ACKNOWLEDGMENTS

We would like to thank the Graduate Program in Applied Computing at Ifes–Serra (PPCOMP) and the Capixaba Open University Program (UnAC) of the Secretariat for Science, Technology, Innovation, and Professional Education (SECTI) of the Government of the State of Espírito Santo, Brazil, for their support in the development of this work.

REFERENCES

Ackermann, L., Neuberger, J., and Jablonski, S. (2021). Data-driven annotation of textual process descriptions based on formal meaning representations. In *International Conference on Advanced Information Systems Engineering*, pages 75–90. Springer.

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, pages 54–59.

Bellan, P., Dragoni, M., and Ghidini, C. (2020). A qualitative analysis of the state of the art in process extraction from text. *DP@AI*IA*, pages 19–30.

Bellan, P., Dragoni, M., and Ghidini, C. (2022a). Extracting business process entities and relations from text using pre-trained language models and in-context learning. In *International Conference on Enterprise Design, Operations, and Computing*, pages 182–199. Springer.

Bellan, P., van der Aa, H., Dragoni, M., Ghidini, C., and Ponzetto, S. P. (2022b). Pet: an annotated dataset for process extraction from natural language text tasks. In *International Conference on Business Process Management*, pages 315–321. Springer.

Bellan, P., van der Aa, H., Dragoni, M., Ghidini, C., and Ponzetto, S. P. (2023). Process extraction from text: Benchmarking the state of the art and paving the way for future challenges. *arXiv preprint arXiv:2110.03754*.

Chinchor, N. (1998). Overview of muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*.

Costa, M. B. and Tamzalit, D. (2017). Recommendation patterns for business process imperative modeling. In *Proceedings of the Symposium on Applied Computing*, pages 735–742.

Detroja, K., Bhensdadia, C., and Bhatt, B. S. (2023). A survey on relation extraction. *Intelligent Systems with Applications*, 19:200244.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A., et al. (2018). *Fundamentals of business process management*, volume 2. Springer.

Dymora, P., Koryl, M., and Mazurek, M. (2019). Process discovery in business process management optimization. *Information*, 10(9):270.

Epure, E. V., Martín-Rodilla, P., Hug, C., Deneckère, R., and Salinesi, C. (2015). Automatic process model discovery from textual methodologies. In *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, pages 19–30. IEEE.

Ferreira, R. C. B., Thom, L. H., and Fantinato, M. (2017). A semi-automatic approach to identify business process elements in natural language texts. In *International Conference on Enterprise Information Systems*, volume 2, pages 250–261. SCITEPRESS.

Fionda, V. and Guzzo, A. (2020). Control-flow modeling with declare: Behavioral properties, computational complexity, and tools. *IEEE Transactions on Knowledge & Data Engineering*, 32(05):898–911.

Friedrich, F., Mendling, J., and Puhlmann, F. (2011). Process model generation from natural language text. In *Advanced Information Systems Engineering: 23rd International Conference, CAiSE 2011, London, UK, June 20-24, 2011. Proceedings 23*, pages 482–496. Springer.

Grohs, M., Abb, L., Elsayed, N., and Rehse, J.-R. (2023). Large language models can accomplish business process management tasks. In *International Conference on Business Process Management*, pages 453–465. Springer.

Honkisz, K., Kluza, K., and Wiśniewski, P. (2018). A concept for generating business process models from natural language description. In *Knowledge Science, Engineering and Management: 11th International Conference, KSEM 2018, Changchun, China, August 17–19, 2018, Proceedings, Part I 11*, pages 91–103. Springer.

Klievtsova, N., Benzin, J.-V., Kampik, T., Mangler, J., and Rinderle-Ma, S. (2023). Conversational process modelling: State of the art, applications, and implications in practice. *arXiv preprint arXiv:2304.11065*.

Kourani, H., Berti, A., Schuster, D., and van der Aalst, W. M. (2024). Process modeling with large language models. In *International Conference on Business Process Modeling, Development and Support*, pages 229–244. Springer.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Li, J., Sun, A., Han, J., and Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.

López, H. A., Strømsted, R., Niyodusenga, J.-M., and Marquard, M. (2021). Declarative process discovery: Linking process and textual views. In Nurcan, S. and Korthaus, A., editors, *Intelligent Information Systems*, pages 109–117, Cham. Springer International Publishing.

Obendorf, H. (2009). *Minimalism: designing simplicity*. Springer Science & Business Media.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Qian, C., Wen, L., Kumar, A., Lin, L., Lin, L., Zong, Z., Li, S., and Wang, J. (2020). An approach for process model extraction by multi-grained text classification. In *Advanced Information Systems Engineering: 32nd International Conference, CAiSE 2020, Grenoble, France, June 8–12, 2020, Proceedings 32*, pages 268–282. Springer.

Schützenmeier, N., Käppel, M., Petter, S., and Jablonski, S. (2021). Upper-bounded model checking for declarative process models. In *The Practice of Enterprise Modeling: 14th IFIP WG*, pages 195–211. Springer.

van der Aa, H., Carmona Vargas, J., Leopold, H., Mendling, J., and Padró, L. (2018). Challenges and opportunities of applying natural language processing in business process management. In *COLING 2018: The 27th International Conference on Computational Linguistics: Proceedings of the Conference: August 20-26, 2018 Santa Fe, New Mexico, USA*, pages 2791–2801. Association for Computational Linguistics.

Van der Aa, H., Di Ciccio, C., Leopold, H., and Reijers, H. A. (2019). Extracting declarative process models from natural language. In *Advanced Information Systems Engineering: 31st International Conference, CAiSE 2019, Rome, Italy, June 3–7, 2019, Proceedings 31*, pages 365–382. Springer.

Yuan, A., Ippolito, D., Nikolaev, V., Callison-Burch, C., Coenen, A., and Gehrmann, S. (2021). Synthbio: A case study in faster curation of text datasets. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Zhao, X., Deng, Y., Yang, M., Wang, L., Zhang, R., Cheng, H., Lam, W., Shen, Y., and Xu, R. (2024). A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Computing Surveys*, 56(11):1–39.