Software and Systems Modeling
https://doi.org/10.1007/s10270-025-01305-1

SPECIAL SECTION PAPER

Repeat, reorder, rephrase: data augmentation for process information extraction

Julian Neuberger[1] · Lars Ackermann[2] · Stefan Jablonski[1]

Received: 30 November 2024 / Revised: 29 May 2025 / Accepted: 4 June 2025

Abstract

Automatic retrieval of formal business process models from their natural language descriptions is a well-established way to facilitate the time- and cost-intensive modeling procedure. Yet, a lack of data usable for developing and training new retrieval methods is impeding progress in this field of research. This issue can be overcome by either using methods less reliant on high-quality data, such as large language models, or creating bigger datasets. The latter is often preferable in the context of business process modeling, especially when internal workflows of organizations have to be treated confidentially. It is the more data-intensive solution, though, which is costly. Data augmentation techniques aim to improve both quality and quantity of existing datasets, by deliberate perturbations resulting in new, synthetic data. In this article, we present a collection of data augmentation techniques, which are specifically selected for the task of improving data quality in the context of process information extraction. We show why data augmentation techniques from the wider field of natural language processing are often not applicable to process information extraction, and how the resulting data differ in terms of linguistic variety, structure, and feature space coverage. In our experiments, data augmentation results in an absolute improvement in the F_1 measure of 5.7% for extracting process-relevant entities from text and 4.5% for extracting relations between those entities. We make all code available at https://github.com/JulianNeuberger/pet-data-augmentation and results for our experiments at https://zenodo.org/doi/10.5281/zenodo.10941423.

Keywords Business Process Extraction · Data Augmentation · Natural Language Processing

1 Introduction

In recent years, many systems for extracting process-relevant information from natural language process descriptions have been proposed to expedite process discovery, i.e., the initial modeling of an as-is process [2, 7, 16, 29, 30, 33, 37, 41]. Interest in this research topic is founded in the fact that discovering such an as-is process is known to require up to 60% of time planned for new business process management projects [17]. Generally, generating process models from natural language descriptions is done in two phases. First, process-relevant information is extracted from the text, which is then used to synthesize a formal business process model. Note that there are direct transformation approaches, as well as conversational agents to generate process models from text [13, 25, 27].

While there are unsupervised approaches to the first phase [6, 19, 26, 34], many recent approaches to business process information extraction are still based on supervised machine learning, which requires tuples of input (process descriptions) and expected output (e.g., process-relevant actors, activities, etc.). Such tuples are expensive to create and require deep knowledge about processes, as well as natural language, which is why even the largest collection of data for process information extraction is still comparatively small, containing just 2,000 examples of process-relevant entities [7], while datasets for other information extraction tasks contain multiple orders of magnitude more examples. Datasets for extraction of named entities and their relations, for example, the DocRed dataset, contain more than 1,500,000 examples of entities and relations between them [42].

Communicated by Han van der Aa and Rainer Schmidt.

✉ Julian Neuberger
julian.neuberger@uni-bayreuth.de

[1] University of Bayreuth, Universitätsstraße 30, Bayreuth, Germany

[2] Hof University of Applied Sciences, Alfons-Goppel-Platz 1, Hof, Germany

Published online: 02 July 2025

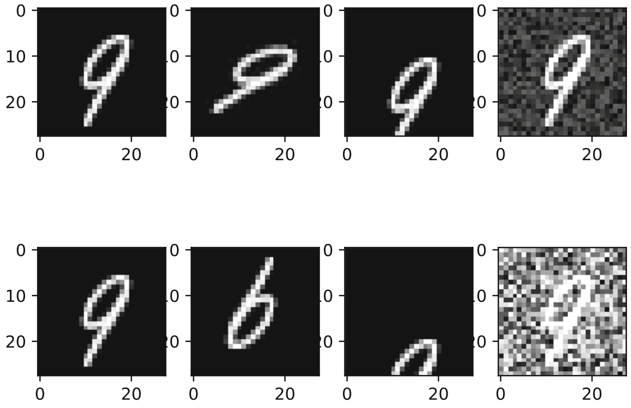


Fig. 1 Motivating example for data augmentation in computer vision

Other domains of research use data augmentation (DA) to improve both quality and quantity of training data. This entails creating new data tuples, by applying targeted perturbations on the input, while preserving the underlying semantics, and therefore maintaining the validity of the expected output, e.g., extracted information. One of the earliest adopters of data augmentation was the field of computer vision. For example, in image classification, images are perturbed by operations including, but not limited to, rotation, translation, or addition of noise. Figure 1 shows these operations applied to an image of a handwritten *9*. Note how the intensity of operations dictates the usefulness of data augmentation in this example. Rotating the image by a few degrees keeps its *semantics*, i.e., it is still an image of the digit 9, but rotating it by 180 degrees yields an image of a 6. Similarly, translating the image too much makes the image ambiguous, and introducing too much noise makes it hard to decipher, even for humans. Still, when configured properly, data augmentation enables more efficient use of valuable training data, creation of more robust models, and better generalization capabilities of those [39, 40].

Despite its popularity and success in other fields of research, data augmentation is used in very few areas of business process management [1, 22, 23]. In this article, we apply 26 data augmentation techniques, specifically designed for information extraction tasks, to process data. We analyze how these data augmentation techniques change the data on a linguistic level, how the resulting dataset covers the feature space, and present configurations of augmentations, which results in data that allow us to train machine learning process information extraction approaches with a total, absolute increase in performance of up to 5.7%. We find that simple to use and computationally cheap perturbations rival the use of large language models in terms of extraction quality improvement, for the current state of the art in process information extraction approaches.

This article is an extension of our previous work in [35], where we used a black-box evaluation to determine the feasibility of data augmentation in a process information extraction context. This work expands on this in two major ways. *(1)* We adjust all augmentation techniques used in the experiment as well as the experiment itself, to improve their applicability to the process information extraction task. We describe these modifications in more detail in Sect. 4.4. Furthermore, we investigate five additional augmentation techniques and two oversampling techniques. *(2)* Section 5 discusses the effects of data augmentation on the data itself. This includes changes in the process description text (Sect. 5.1), common errors introduced by data augmentation (Sect. 5.2), and visualizations of changes in meaning through data augmentation (Sect. 5.3).

The rest of this article is structured as follows. Section 2 defines important notions of data augmentation and the process information extraction task. Section 3 covers work related to this article. We then describe our experiment setup in Sect. 4, starting with our leading research questions (Sect. 4.1), the selection of data augmentation techniques (Sect. 4.2), a classification of these (Sect. 4.3), how we adjusted some techniques for the use with process information extraction data (Sect. 4.4), and finally our strategy for finding optimal configurations for data augmentation techniques (Sect. 4.5). Section 5 analyzes the effects of data augmentation on the text of process descriptions, common errors introduced by data augmentation (Sect. 5.2), visualizations of the changes in meaning (Sect. 5.3), and vocabulary and relation direction (Sect. 5.4). Section 6 discusses the results of a black-box evaluation using data augmented with the selected techniques and answers our research questions. We conclude the article in Sect. 8.

2 Background

Data augmentation describes a suite of techniques originally popularized in computer vision [40], where operations, such as cropping, rotating, or introducing noise into images greatly improved performance of machine learning algorithms used for classification of images. Operations such as these usually preserve the semantics of input data, meaning that an image containing an object will still depict the same object after its data have been augmented, for example, they have been overlaid with noise. This property is called *invariance* [43] and is harder to hold for natural language data [15]. An example for this fact is depicted in Fig. 2. Changing random tokens (e.g., words) in a sentence may alter semantics to a point, where relevant elements or relations between those elements are no longer present after augmentation. Additionally, annotations may be lost, if techniques are applied and afterward these changes cannot be traced. This might hap-

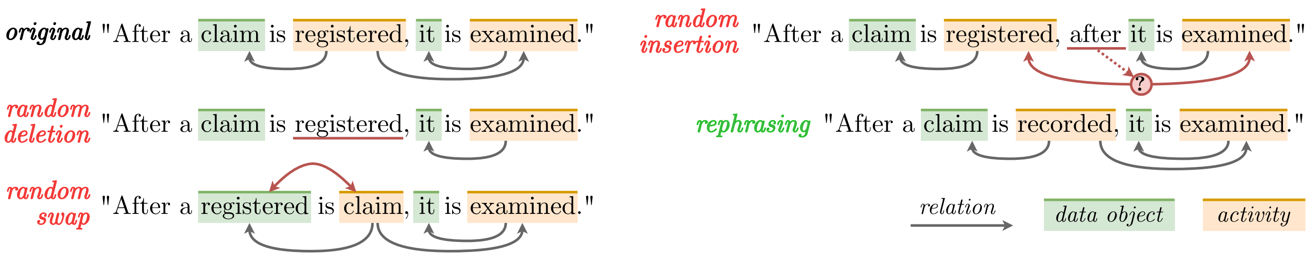

original "After a claim is registered, it is examined."
random deletion "After a claim is registered, it is examined."
random swap "After a registered is claim, it is examined."
random insertion "After a claim is registered, after it is examined."
?
rephrasing "After a claim is recorded, it is examined."
relation
data object
activity

Fig. 2 Examples for four different data augmentation techniques. *Random deletion*, *random swap*, *random insertion* (all written in red) are all not preserving the semantics of a sample and its label. *Rephrasing* (green) is an example for a technique that does

pen, when, for example, an entire sentence is translated into another language and then is back-translated typically leading to a rephrased version of the original text. Since it is not clear, which parts of the new sample correspond to the original one, annotations of process-relevant elements may not apply to the new sample. For this reason, research on data augmentation techniques has been conducted, which are specifically designed for information extraction tasks [14, 20, 28]. These techniques use additional resources, such as pre-trained large language models, to augment training samples, while keeping their semantics intact.

Process-Relevant Information Extraction from natural language is a research field immediately relevant for information systems, as business process models are often a central part for process-aware information systems. Discovering and creating these process models are an expensive task [17], and a lot of work has been done on extracting them from natural language text directly [2, 6, 16, 17, 41]. These texts describe a business process in natural language as technical documentation, maintenance handbooks, or interview transcripts. Sequences of words (spans) in these texts contain information that is relevant to the business process, such as *Actors* (persons or departments involved in the process), *Activities* (tasks that are executed), or *Data Objects* (physical or digital objects involved in the process). Extracting this information is therefore a sequence tagging task and can be framed as *Mention Detection* (MD). Mentions relate to each other, e.g., defining the order of execution for two Activities, or which Actor executes the Activity. Predicting and classifying these relations is called *Relation Extraction* (RE). Refer to Fig. 3 for an example of this process. It shows a fragment of a larger description of a process from the insurance domain, where insurance claims have to be registered in a system and subsequently examined by an employee. The spans *claim* and *it* are annotated as Data Objects (the claim in question, in green). Activities executed by a process participant are marked in orange. These four spans can now be transformed into business process model elements for a target notation language (here BPMN[1]). How these elements interact with each other can also be extracted from the text fragment, e.g., the *Flow* of activity execution between the mentions *registered* and *examined*, depicted as an orange arrow.

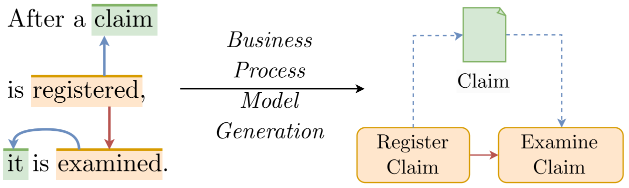


Fig. 3 Example for a fragment of a natural language business process description and its corresponding business process model fragment in BPMN

Developing approaches toward automated extraction of process-relevant information requires data to test performance, and train models, if applicable. The currently largest collection of human-annotated process descriptions is called PET [7]. It contains 45 natural language process descriptions and is annotated with 7 types of process-relevant entities (e.g., Actors, Activities, Data Objects), as well as 6 types of relations between them (e.g., Flow between Activities). In total, the dataset contains less than 2,000 examples for both relations and entity mentions. For comparison, typical datasets for related tasks, like knowledge graph completion, contain more than 200 times as many. For example, the popular *FB15k* dataset comprises more than 500,000 relation examples [10]. Datasets for extraction of named entities and their relations have similar extents, e.g., the DocRed dataset, which contains more than 1,500,000 relation examples [42]. This fact makes PET a prime candidate for data augmentation techniques, in order to make the most out of the limited amount of training examples. We show this in our experiments using PET for the tasks MD and RE in process information extraction. To our knowledge, our work is the first to attempt applying NLP data augmentation to the process information extraction task.

3 Related work

Data augmentation techniques applied in this paper are largely based on the ones available in the *NL-Augmenter*

[1] See specification at https://www.bpmn.org/.

framework [12]. NL-Augmenter provides a list of more than 100 data augmentation techniques, which are suitable for varying tasks like text classification, sentiment analysis, and even tagging. We discuss how we adapted these techniques to the PET data format in more detail in Sect. 4. Not all techniques are relevant for this work, and we have to exclude most of them, as they are not fitting for process information extraction. Details of our exclusion criteria are found in Sect. 4.2.

This paper is not the first work that applies data augmentation to a business process management task. In [22], the authors evaluate nine simple data augmentation techniques (e.g., random deletion) on a total of seven event logs, using seven different models. These event logs are then used for predictive process monitoring tasks. Our paper follows a similar line of thought, but instead of predictive process monitoring, we apply data augmentation to data for process information extraction from text. The techniques we employ differ significantly from theirs in two core aspects. First, techniques used in the paper at hand are more complex, owing to the more complex character of natural language. While their work focused on reordering events in a log of a process execution, our work uses techniques that are concerned with replacing, extending, or modifying sequences of text, while preserving any annotations present in the data. Second, techniques used in our work often require external resources. These resources can be explicit, i.e., databases like WordNet [31], which contains lexical information such as synonyms, antonyms, or hypernyms of words. They can also be implicit, such as large language models, which contain knowledge about natural language, obtained by unsupervised training on huge amounts of textual data [11].

The techniques we present in our paper mainly benefit work that already exists in the field of process information extraction. Therefore, approaches based on machine learning are related to this work. These approaches can be separated into two main fields of research: *(1)* learning approaches, which use the data to train a machine learning models, e.g., a neural network [2], conditional random fields [7], or decision trees [33]; *(2)* prompting-based approaches that use the data for engineering input for large language models (e.g., GPT) [21, 24, 34], or use the data for so called *in-context learning*, by providing examples in the input itself [6].

Automated extraction of information relevant to business processes from natural language text descriptions can be seen as a special case of automated knowledge graph construction or completion [5]. We therefore consider techniques for automated knowledge graph construction and completion as distantly related work that could still benefit from the augmentation techniques we analyze in this paper. Nonetheless, we focus on methods of process information extraction in this paper, as potential solution for this field's small datasets.

4 Experiment setup

This article answers four leading research questions, defined in Sect. 4.1. To this end, we use data augmentation techniques from the domain of natural language processing. The NL-Augmenter framework [12] provides a total of 117 of such data augmentation techniques, but not all of them are applicable to the task at hand. We therefore define four criteria for exclusion in Sect. 4.2. We group the remaining, selected data augmentation techniques into five classes in Sect. 4.3 and discuss how we modify them to be better applicable to process descriptions in Sect. 4.4. Finally, in Sect. 4.5 we discuss our approach toward finding optimal configurations for the parameters of data augmentation techniques.

4.1 Research questions

Following the intuition from Sect. 1, our first two research questions 4.1 and 4.1 are focused on answering how much knowledge about natural language and process information extraction is needed to augment data for the process information extraction task.

While *simple data augmentation*, i.e., randomly deleting, swapping, or inserting words into a text, is easy to implement and require no additional knowledge about natural language or process information extraction, they are also prone to breaking semantics of process descriptions and introducing unnatural noise. Still, related work in business process management has been shown to benefit from such simple data augmentation techniques, e.g., predictive business process monitoring [22, 23]. These considerations lead us to raising research question 4.1.

On the other hand, using large-scale language modeling, e.g., through the use of pretrained language models, such as BERT [11], or GPT [38], is highly demanding in terms of resources (hardware), and time. We therefore are interested in how useful these methods are compared to smaller, rule-based methods, i.e., if the investment of hardware and time is worth it through significantly higher data quality, measured by the performance gain of models trained with it. We therefore pose research question 4.1.

One popular technique to improve classification of rare classes in a classification task is called *oversampling* [32]. Here, samples that contain rare classes are shown multiple times during training. Preliminary experiments showed that the RE method presented in [33] can already benefit from this technique. Transformations that yield improvements lower than oversampling could even be considered detrimental to model performance. For this reason, a third question, 4.1, is focused on finding data augmentation techniques that are better than simply repeating data.

The fourth and final one question, 4.1, is aimed at the *apparent* effects of data augmentation on the text of process descriptions. This question is aimed at understanding how process descriptions change as a result of applying data augmentation. Since this is a highly qualitative question, we aim to answer it by exploring process descriptions altered by data augmentation, finding common errors, and visualizing changes in characteristics.

In summary, our research questions are as follows.

RQ1 Can simple data augmentation techniques, including swapping, deleting, or randomly inserting words into sentences, increase the performance of machine learning methods for process information extraction, measured as the harmonic mean (F_1 score) of precision and recall?

RQ2 Does the use of deep learning models, especially (large) neural language models, in data augmentation, provide an advantage measurable using the F_1 score over simpler, rule-based methods?

RQ3 Does data augmentation outperform oversampling of documents during training?

RQ4 What characteristics of the natural language text data are changed by augmentations?

4.2 Selection of techniques

The NL-Augmenter framework provides a total of 119 data augmentation techniques, but not all of them are applicable to the task at hand. We therefore define four criteria for exclusion.

EC1 Language: The technique does not apply to the English language, i.e., we exclude techniques targeted at all other languages. The dataset we use for our experiments, PET, is in English, techniques targeting other languages are therefore not relevant for this paper.

EC2 Spelling: The data augmentation technique alters the spelling of tokens, i.e., misspelling perturbations, which allows for evaluating robustness to spelling mistakes. This issue is not present in the PET dataset, but may be relevant for future work, if less perfect data sources (e.g., notes taken by employees) are analyzed.

EC3 Supervised Training Data: The data augmentation technique does not work for supervised data, i.e., perturbations would corrupt labels present in the PET dataset and we cannot adjust it to preserve labels.

EC4 External Resources: The technique uses task-, and/or domain-specific resources, such as dictionaries, or databases, which do not exist for processes represented in PET and prevent the technique to be task-agnostic.

Applying these criteria results in 20 data augmentation techniques relevant for the task of generating business process models from natural language text. We have listed the exact number of data augmentation techniques excluded by each criterion in Fig. 4. Two of the 20 relevant techniques had errors in their original code, which we fixed. Note that one of the 20 selected techniques had two modes of operation, which we split into two separate techniques, resulting in a total of 21 data augmentation techniques at this stage. We then added five more data augmentation techniques, specifically designed to help us answering our four research questions, which are as follows.

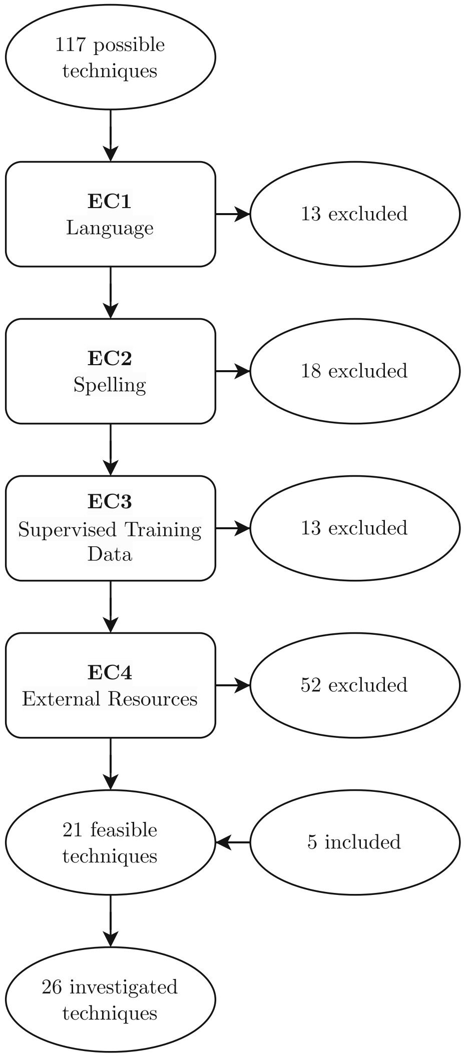


Fig. 4 Application of exclusion criteria, and inclusion of five additional techniques, leading to 26 investigated data augmentation and oversampling techniques

Random Swap. Randomly selects two tokens in the document and swaps them.

Random Insert. Uniformly samples tokens from the dataset vocabulary and inserts them at random positions in the document. Together with augmentation technique B.79 (Random Deletion) from the NL-Augmenter framework, Random

Swap, and Random Insert are the three techniques the *simple data augmentation techniques* referenced in research question 4.1.

Large Language Model Rephrasing. Rephrase text segments via a Large Language Model (LLM). We utilize a local *Llama 3.1* model with 70B parameters for this augmentation technique and use the results to supplement those of other, existing techniques in answering research question 4.1. This augmentation technique is the most time-intensive one by far (see Sect. 2), but should help us get a good idea of the usefulness of generative artificial intelligence for data augmentation.

Inverse Type Frequency Oversampling. Oversample, i.e., repeat documents in the dataset according to the rarity of their contained entity and relation types.

Uniform Oversampling. Randomly oversample documents from the dataset. These two oversampling techniques serve as a baseline of improvement, to answer research question 4.1.

4.3 Classifying data augmentation techniques

Many data augmentation techniques are similar in their inner workings, i.e., they use similar approaches toward perturbing documents. In this section, we present the four major categories of data augmentations we identified by running a given data augmentation and analyzing the changes it made. If an augmentation changed several characteristics of the original text, we chose the most prominent one as its category. An overview of the mapping between data augmentations and their corresponding categories are found in Table 1.

C1 *Rephrasing*. This class of data augmentation techniques targets spans of one or more tokens and replaces them with synonymous wording. It also includes augmentations that replace words by their abbreviations or vice versa. Techniques in this class are not guaranteed to keep syntactic and semantic integrity, but do so more often than techniques in other classes.

C2 *Reordering*. Techniques that change the order of tokens, without introducing new tokens fall into this class. The scope of such techniques varies and ranges from reordering the tokens of a single mention, reordering sentences (without reordering tokens inside these sentences), or randomly swapping tokens in a document. Unlike *Rephrasing*, techniques of this class do not change words, only their ordering. Techniques in this class regularly break linguistic syntax.

C3 *Repeating*. When techniques use existing data, we categorize them as repetition techniques. These include data augmentation that concatenate documents, repeat whole documents (oversampling), or randomly insert token spans sampled from the dataset. Contrary to the classes *Rephrasing* and *Reordering*, techniques in this class do not change vocabulary, nor the ordering of phrases, but repeat existing structures in the text.

C4 *Noise*. Data augmentation techniques in this class randomly insert, delete, or replace tokens in the data. Inserted and replaced tokens are not part of the original dataset and include speaker phrases ("*uhm*," "*er*") or inserting (not substituting) synonyms of words.

4.4 Adjustment for process information extraction

We have to adjust some of the augmentation techniques slightly, to make them applicable to the process information extraction task. Our adjustments fall into the following categories.

Label Preservation. This adjustment applies mainly to augmentations that replace spans of tokens, such as rephrasing techniques. If the technique would rephrase an entire sentence, such as "*After **a claim** is registered, it is examined.*," it could result in "*Following registration of a claim, a review follows.*" Even though the rephrased sentence has the same number of tokens, naively assuming the positions of labels have not changed would result in ***registration of*** as an extraction target for training extraction models, impacting the performance of models trained with such data. Instead, we segment the text into sequences with the same corresponding entity type, i.e., '*After*,' '*a claim*,' '*is registered*,' '*it*,' '*is examined*,' and '..' We then rephrase the segments in isolation and replace the entire original sequence with the resulting rephrased sequence. This way, we can preserve labels, but may break the semantics of the sample (see Sect. 5.2).

Runtime Optimizations. Some augmentation techniques had obvious bottlenecks adversely affecting runtime, such as loading language models multiple times throughout their lifetime. One example of an augmentation technique implemented in this way is *Lost in Translation* (B.58), which instantiates for every perturbation a pipeline to encode and decode text. Loading times resulting from this lead to excessive runtimes, which can be avoided by instantiating the pipeline once, and reusing it for subsequent perturbations. We optimized the code for this and similar cases, allowing us to run augmentations repeatedly. This becomes important for the experiments in Sect. 4.5, where the huge number of repetitions required for finding optimal configurations would otherwise not be feasible in acceptable time frames.

Forcing Variety. When we augment a document with many times, a single augmentation technique potentially has to produce more than one augmented document, e.g., two augmented documents for the augmentation factor 2. Naively implementing this by re-running the augmentation may lead to identical augmented documents, especially for deterministic techniques. Were possible, we changed existing

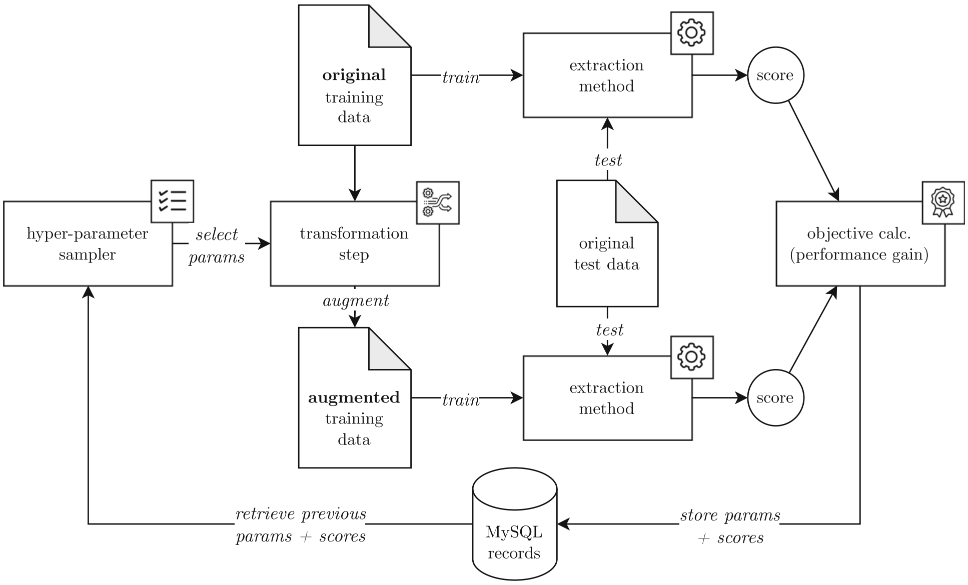


Fig. 5 Choosing optimal configurations for data augmentation techniques

techniques in such a way, that they return a list of augmented documents that are different from each other.

4.5 Finding optimal configurations

Each of the data augmentation techniques we selected can potentially be adjusted by several parameters, which control how augmented samples are synthesized. A typical example for such a parameter is the number of inserted tokens. Increasing this number would result in a sample, which is more perturbed compared to a sample where fewer tokens are inserted. We consider optimally choosing such parameters for a given technique a *hyper-parameter optimization* problem. Hyper-parameter optimization is defined as finding a configuration of parameters so that a given objective (metric to optimize) is minimal or rather maximal, depending on the case. Here, we want to maximize the performance gain that the application of a data augmentation technique has. To that end, we run a 5-fold cross-validation of the extraction step (MD, RE) with the original, unaugmented data. We then select a configuration for the given technique and run the same fivefold cross-validation, but augment the training data of each fold with the data augmentation technique. We define the difference between the scores of these two models on the (unaugmented) test dataset as the *performance gain* and use it as maximization objective for our hyper-parameter optimization. Each data augmentation technique is optimized in 25 runs (*trials*) using Optuna [4] and a Tree-Structured Parzen Estimator for selecting parameter values [9]. We depict this process in Fig. 5.

We use the best parameter configurations of a given data augmentation technique as its default configuration in later sections and experiments. You can find values for the parameters in each augmentation technique in Appendix 9.

5 Data augmentation effects

In this section, we will describe the classes of data augmentations. Table 1 shows a compact overview of all data augmentation techniques. It also summarizes the information found in this section, which is structured as follows. First, we start by analyzing the process descriptions themselves, i.e., how the surface form of the original text is changed by select data augmentation techniques in Sect. 5.1. We derive common classes of errors from this analysis and discuss how these errors impact the quality of synthesized data in Sect. 5.2. Next, to generalize the analysis of how augmentation changes the text, and more importantly, meaning of process descriptions, we visualize the text of whole process descriptions, sentences, and process entity mentions as scatter plots in Sect. 5.3. Finally, we discuss three more characteristics of textual process descriptions, linguistic variability, vocabulary size, and relation direction, in Sect. 5.4.

5.1 Surface form changes

In this section, we will discuss how data augmentation changes the wording (surface form) of process descriptions. We will also discuss how these changes affect the annotations (labels) of data, such as mentions and relations. Throughout this section, we will show examples of augmented process descriptions base on the running example shown in Fig. 6. To improve clarity, we omitted all relations with the exception

Table 1 Overview of augmentation techniques considered in this article. Column *Id* refers to identifier used in [12], column *category* is one of the categories defined in Sect. 4.3, while column *Errors* references one of the common error classes defined in Sect. 5.2, i.e., Semantics (SEM), Syntax (SYN), Punctuation (PUN), violations of Annotation Guidelines (GL), and deleted Annotations (AN)

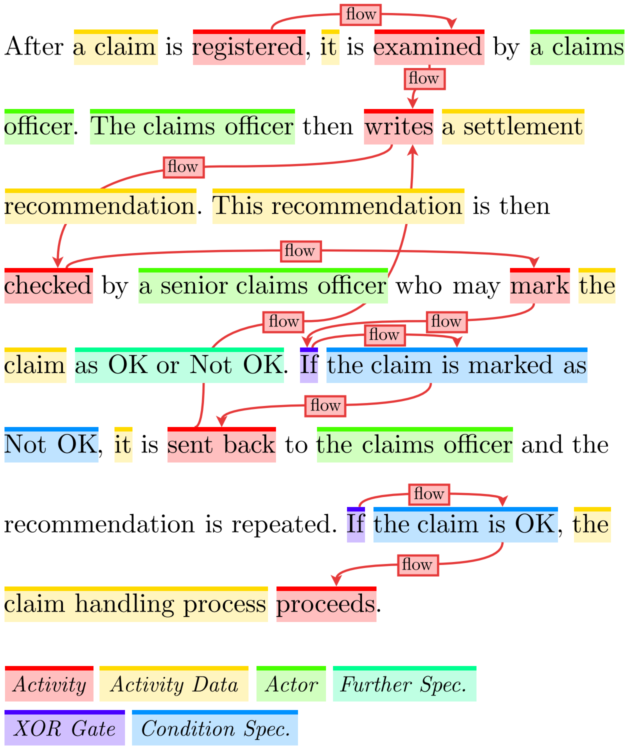
Technique	Id	Description	Category	Errors
Adjectives Antonyms Switch	B.3	use antonyms of adjectives	Noise	SEM
AntonymsSubstitute (Double Negation)	B.5	substitute even number of words with antonyms	Rephrasing	SEM
Auxiliary Negation Removal	B.6	remove negated auxiliaries	Noise	SEM
BackTranslation	B.8	translate to German, then back to English	Rephrasing	SEM, SYN, PUN
Concatenate Two Random Sentences	B.24	remove PUN between sentences	Repeating	PUN
Contextual Meaning Perturbation	B.26	replace words with use of pretrained language model	Rephrasing	SYN, SEM, GL
Contractions and Expansions Perturbation	B.27	Contract phrases where common contractions exist, e.g., *"I am"* to *"I'm"* and vice versa	Rephrasing	—
English Mention Replacement for NER	B.39	replace mention with one of the same types in document	Repeating	SYN, SEM
Filler Word Augmentation	B.40	introduce "uhm," "I think," ...	Noise	GL
Lost in Translation	B.58	repeatedly translate text segment into other languages and finally translate it back	Rephrasing	SYN, SEM, PUN, GL
Multilingual Back-Translation	B.62	see B.8, language is parameter	Rephrasing	SEM, SYN, PUN
Random Word Deletion	B.79	delete random words	Noise	AN, SYN, SEM
Replace Abbreviations and Acronyms	B.82	replace acronyms with full length expression and v.v.	Rephrasing	SEM
Hypernym Replacement	B.86	replace token with its hypernym (super term), e.g., Mountain Bike with vehicle	Rephrasing	SEM
Hyponym Replacement	B.86	replace token with its hyponym (super term), e.g., Mountain Bike with Bicycle	Rephrasing	SEM
Sentence Reordering	B.88	reorder sentences	Reordering	SEM
Shuffle Within Segments	B.90	shuffle tokens in mentions	Reordering	SYN, SEM, PUN
Synonym Insertion	B.100	insert synonym before word	Noise	SYN, SEM
Synonym Substitution	B.101	substitute word with synonym	Rephrasing	SEM
Subsequence Substitution for Sequence Tagging	B.103	replace sequence with another sequence with same POS tags	Repeating	SEM
Transformer Fill	B.106	replace tokens using language model	Rephrasing	SYN, SEM
Random Insert		insert random tokens	Noise	SYN, SEM, GL, PUN
Random Swap		swap position of tokens	Reordering	SYN, SEM, GL, PUN
Large Language Model Rephrasing		use an LLM to rephrase text segments, while considering the entire surrounding process description as context	Rephrasing	SYN
Inverse Type Frequency Oversampling		oversample documents containing rare mention / relation types	Repeating	—
Uniform Oversampling		uniformly oversample documents	Repeating	—

of the *Flow* relations, which we will use in some augmented examples.

5.1.1 Rephrasing augmentations

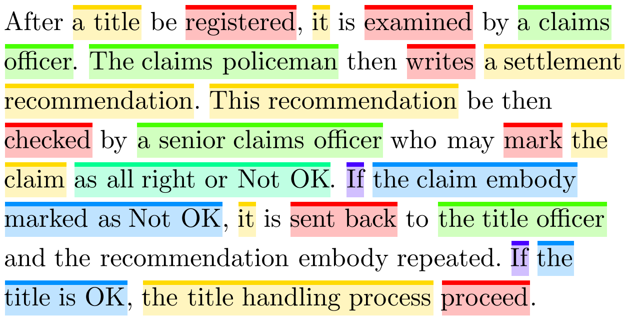
Consider, for example, the technique Synonym Substitution (B.101) as representative for augmentations that rephrase text segments in process descriptions, with results as shown in Fig. 7.

Synonym Substitution changes the form of verbs heavily. While it properly selects synonyms, it fails to inflect them according to the original verb. Synonym substitution also fails to take context into account and as such uses synonyms that are contextually wrong, e.g., replacing *settlement* with *colony* in *a settlement recommendation*, when settlement is used in a financial context. These problems can be subsumed as breaking both syntax and semantics to some degree. Rephrasing augmentations, that only replace a single



After a claim is registered, it is examined by a claims officer. The claims officer then writes a settlement recommendation. This recommendation is then checked by a senior claims officer who may mark the claim as OK or Not OK. If the claim is marked as Not OK, it is sent back to the claims officer and the recommendation is repeated. If the claim is OK, the claim handling process proceeds.

Fig. 6 Original surface form of document *doc-3.3* of the PET dataset



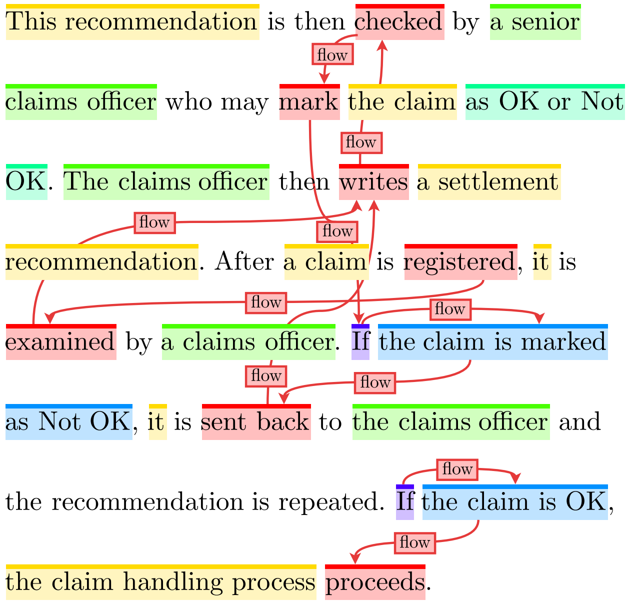
After a title be registered, it is examined by a claims officer. The claims policeman then writes a settlement recommendation. This recommendation be then checked by a senior claims officer who may mark the claim as all right or Not OK. If the claim embody marked as Not OK, it is sent back to the title officer and the recommendation embody repeated. If the title is OK, the title handling process proceed.

Fig. 7 Effects of synonym substitution (e.g., policeman for officer) on surface form. Relations are unchanged by this augmentation technique, so we omit them for brevity

token at a time did not break annotations in our experiments, i.e., they did not invalidate training targets of the dataset.

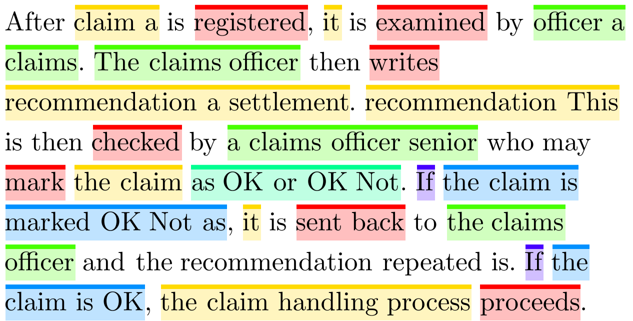
Reordering Augmentations. For reordering augmentations, we use the Sentence Reordering (B.88) technique as an example, shown in Fig. 8.

This augmentation does not change the surface form of mentions, but is one of the few augmentations that change the direction of relations. Depending on the role of a relation, this augmentation is problematic for process information data. Take, for example, the *flow* between *mark* and *writes* in the original data shown in Fig. 6 and the augmented one in Fig. 8. Reordering sentences leads to a different flow when following the description in the text, but the *flow* relation is not updated to reflect that.



This recommendation is then checked by a senior claims officer who may mark the claim as OK or Not OK. The claims officer then writes a settlement recommendation. After a claim is registered, it is examined by a claims officer. If the claim is marked as Not OK, it is sent back to the claims officer and the recommendation is repeated. If the claim is OK, the claim handling process proceeds.

Fig. 8 Effects of reordering sentences on the direction of relations. Note that many relations are now wrong, as the text actually describes the order of execution differently

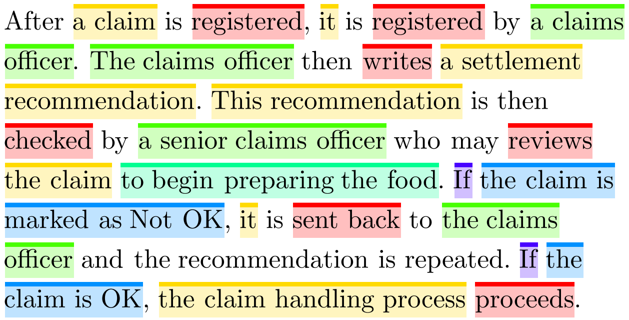


After claim a is registered, it is examined by officer a claims. The claims officer then writes recommendation a settlement. recommendation This is then checked by a claims officer senior who may mark the claim as OK or OK Not. If the claim is marked OK Not as, it is sent back to the claims officer and the recommendation repeated is. If the claim is OK, the claim handling process proceeds.

Fig. 9 Effects of shuffling tokens within segments. This augmentation does not affect the direction of relations

Other augmentations in this class change the surface form more dramatically, such as the *Shuffle Within Segments* augmentation, which splits the process descriptions into segments based on mention types (or lack thereof) and shuffles the tokens of randomly sampled segments. Figure 9 shows this, where, for example, the phrase *a recommendation settlement* becomes *recommendation a settlement*.

Note that this class of augmentation breaks syntax regularly and may even border on breaking semantics, e.g., when a phrase like *as OK or Not OK* becomes *as OK or OK Not*, like in Fig. 9. Since the punctuation is also subject to shuffling, these augmentation techniques in this class sometimes produce invalid punctuation.



After a claim is registered, it is registered by a claims officer. The claims officer then writes a settlement recommendation. This recommendation is then checked by a senior claims officer who may reviews the claim to begin preparing the food. If the claim is marked as Not OK, it is sent back to the claims officer and the recommendation is repeated. If the claim is OK, the claim handling process proceeds.

Fig. 10 Effects of replacing entity mentions in the process description with those of the same type, e.g., *examined (activity)* with *registered (activity)* from the same document, or *as OK or Not OK (further specification)* with *to begin preparing the food (further specification)*, from a different document. This augmentation does not affect the direction of relations

5.1.2 Repeating augmentations

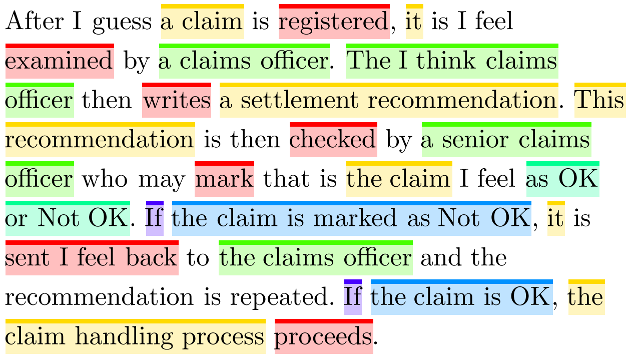
Augmentations that repeat entire documents, such as the oversampling strategies *Inverse Mention Frequency Sampler*, or *Uniform Repeat*, do not change the surface form of documents. Similarly, the *Merge Documents* augmentation strategy merges two random documents to produce a new one, which technically changes the surface form, but not in any conceptually interesting way.

For this reason, we will focus on augmentations that insert and replace parts of a document with repeated phrases from other documents, such as "*English Mention Replacement for NER*." Figure 10 shows an example for this augmentation technique. Since the replacements are sampled randomly, this technique and others like it tend to break both syntax and semantics of process descriptions. Still, since it respects the type of the replaced entity mention, it is potentially useful for the mention detection and relation extraction tasks, as we show in our experiments in Sect. 6.

5.1.3 Augmentations adding noise

Augmentations that add random tokens into the text can help to make models more robust [12]. These augmentation techniques are very likely to break both semantics and syntax of textual data. Still, they can be useful to simulate noisy input, which becomes increasingly useful, e.g., in the environment of chat bots, especially when the input is a transcript of human speech.

Figure 11 shows the effects of inserting so called speaker phrases into the process description. Speaker phrases are utterances of uncertainty, filler phrases, or other exclamations that do not contain any information, such as "*err*," or "*uhm*."



After I guess a claim is registered, it is I feel examined by a claims officer. The I think claims officer then writes a settlement recommendation. This recommendation is then checked by a senior claims officer who may mark that is the claim I feel as OK or Not OK. If the claim is marked as Not OK, it is sent I feel back to the claims officer and the recommendation is repeated. If the claim is OK, the claim handling process proceeds.

Fig. 11 Effects of inserting filler words, simulating uncertainty in a speaker

5.2 Common errors

We identified five types of errors, reoccurring in many of the data augmentation techniques. In this section, we present these errors and discuss their implications.

5.2.1 Violating annotation guidelines

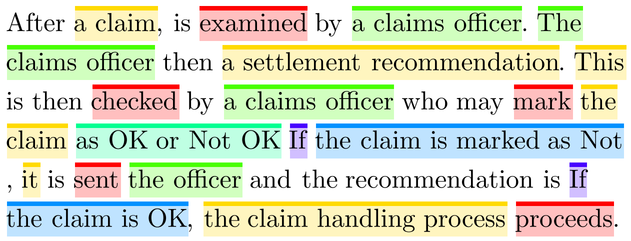
When inserting tokens into the process description, such as filler words (Fig. 11), or tokens sampled from the remaining dataset (Fig. 10), the boundaries of mentions may be expanded. This is the case, when a token is inserted in the middle or directly after the span of tokens making up a mention. For example, inserting "*I think*" into the second sentence in Fig. 11 expands the mention "*The claims officer*." A better way of annotation would be using non-continuous spans of text (only *The* and *claims officer*), but the annotation guidelines of PET are not designed for this, and instead would ask annotators to only annotate "*claims officer*." When we automatically augment data, we cannot decide which parts of a mention to keep, and which to discard, should an inserted token bisect a mention.

5.2.2 Removing annotations

Randomly deleting tokens may remove mentions completely, if its only token is deleted. This has a cascading effect, as we also have to delete relations that use that mention as an argument. Compare, for example, the text that results after we ran the *Random Deletion* augmentation technique, shown in Fig. 12, with the original shown in Fig. 6. Many *activity* mentions are now missing, and as a result the corresponding *flow* relations as well.

5.2.3 Breaking syntax

It can be argued that breaking syntax to some degree can be useful for improving the robustness of information extrac-



After a claim, is examined by a claims officer. The claims officer then a settlement recommendation. This is then checked by a claims officer who may mark the claim as OK or Not OK If the claim is marked as Not , it is sent the officer and the recommendation is If the claim is OK, the claim handling process proceeds.

Fig. 12 Random deletion augmentation deleting many tokens, mentions, and relations

tion approaches. This becomes clear, when a human reads the sentence "*After a claim registered is, [...].*" The intention can still be transported, even though there are grammatical inaccuracies. Yet, this intuition can also show how syntax can be broken beyond a point, where compensation is possible. Randomly ordering the tokens from the previous example leads to "*Is after claim a registered, [...],*" which becomes hard to understand, or even ambiguous (is the claim registered after something, or is something happening after a claim is registered). Reordering is not the only operation that broke syntax in our experiments; we also observed this when replacing or inserting tokens.

5.2.4 Breaking semantics

Unlike syntax, linguistic semantics are harder for extraction approaches to exploit, as it requires modeling language [8]. In recent years, large pretrained language models have made this easier, but the amount of data available for developing process information extraction approaches is impeding their use [33]. In turn, this means while breaking semantics appears egregious for humans, it is less important for the shallow learning approaches we use in our experiments. As a result, approaches that use large amounts of compute, to preserve semantics, are only marginally (if at all) better in our black-box evaluation. Nonetheless, we argue that avoiding semantic-breaking errors will become more and more important, when larger models are used for process information extraction. Figure 7 preserves syntax properly, but changes the semantics of "The claim officer" so much that it is even hard for human readers to recognize its other mentions (*claims policeman* and *title officer*).

5.2.5 Invalid punctuation

When augmentations remove or insert tokens, they may remove or insert punctuation. The latter is especially the case for all back-translating augmentations, as they translate only small, incomplete fragments of text and often end the translated fragment with punctuation. Invalid punctuation is generally not an issue for state-of-the-art extraction methods, but can be a problem for the relation extraction method used in this article, as it uses the sentence index as a feature for predicting relations between arguments [33].

5.3 Variations in meaning

The aim of this section is to give a high level overview of the changes in meaning introduced by augmentation. To this end, we embedded both the original and augmented documents, sentences, mentions, and relations using the Jina Embedding model [18]. This embedding model is based on a BERT architecture [11], i.e., a transformer model that can embed sequences of up to 8192 input tokens and is therefore very well suited to embed even long texts, such as process descriptions. Since relations are not pure text and cannot directly be embedded using a text embedding model, we formatted them as "*text of head mention –> text of tail mention*" and embedded the resulting string. The resulting embedding vectors have 768 dimensions, which cannot be visualized directly. Instead, we use openTSNE [36], a dimension reducing transformation based on TSNE, which allows us to learn this transformation on the vectors of the original documents, and then apply the same transformation on the vectors of the augmented documents.

Plotting the result as a scatter plot lets us visualize how a given class of augmentation changes the meaning of entire process descriptions (documents), sentences, mentions, or relations. Figure 13 shows such as scatter plot for the entire text of documents in the PET dataset. Augmentations that do not change the semantics, such as *reordering*, or *adding noise*, are hardly visible in these scatter plots, as their embeddings barely change. *Repeating* augmentations on the other hand change them enough, so that they cover previously empty space in the visualization. Similarly, rephrasing augmentations introduce small variations in the text, which are minute, but visible in our visualizations.

Visualizing how the content of sentences changes (Fig. 13) results in similar patterns; *Reordering* and *Adding Noise* barely change the vectors of sentences, i.e., their overall content and meaning. *Repeating* and *Rephrasing* augmentations produce more varied text. Visualizations of sentences already show a property that is intensified, when we visualize mention texts in Fig. 13—since sentences are smaller, text variations have a larger effect on their overall embedding. This leads to reordering and noise augmentations being visible in the visualization, compared to visualizations on document level, where they are invisible.

Mentions are usually quite short and only comprised of a few tokens. Similar to sentences, even small alterations of their text have effects on the resulting embedding vectors and their position in the scatter plot. Still, rephrasing has the most pronounced effect on the text (see Fig. 13).

Embedding and visualizing relations shows similar results as sentences and mentions. Rephrasing shows the most pro-

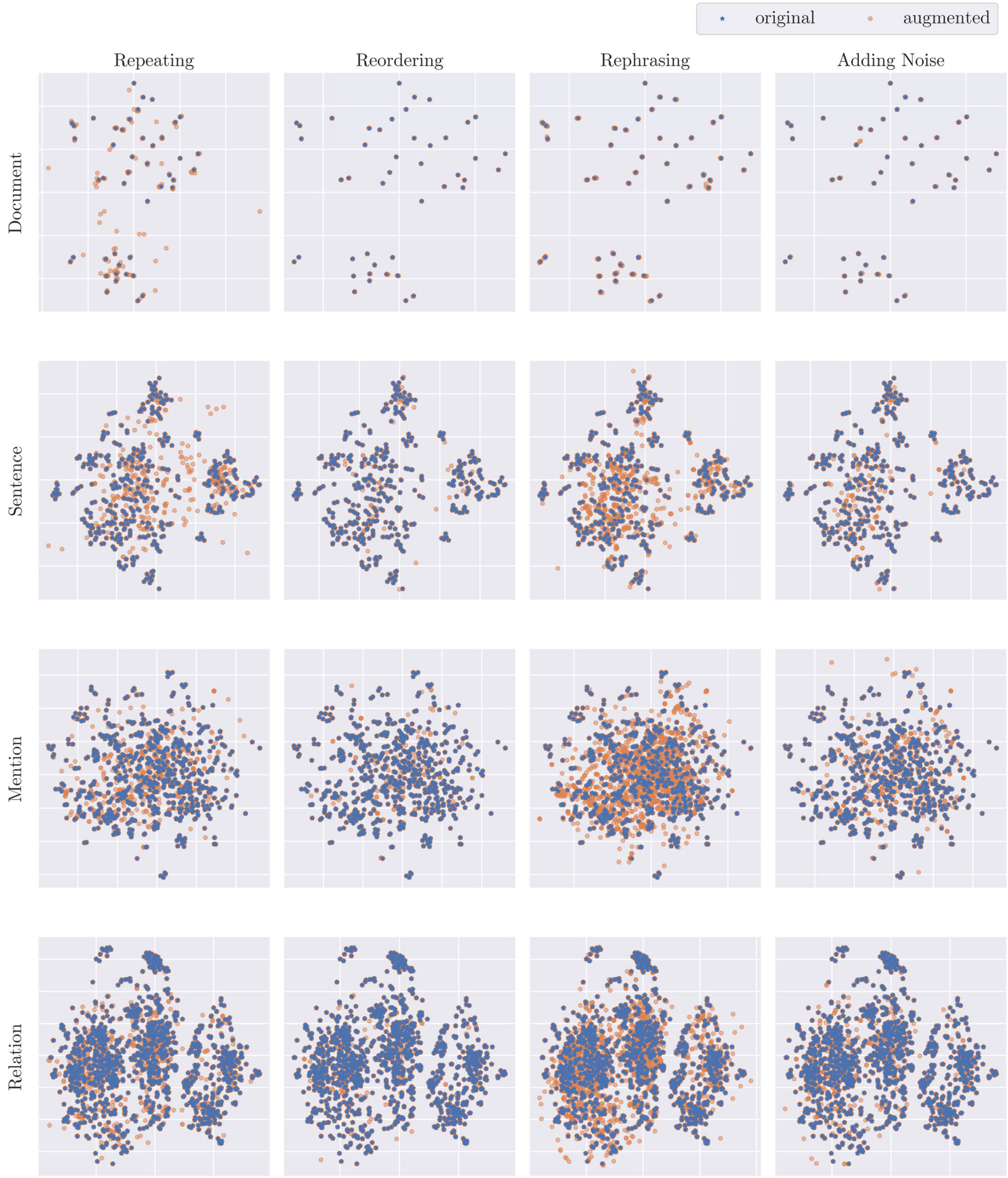


Fig. 13 Text embeddings of original documents (blue) and those augmented (orange) with a given class of techniques (columns). Shown are embeddings of the entire document text (first row), individual sentences (second row), mentions (third row), and relations (last row)

nounced effect on embeddings. Surprisingly, the relation embeddings of augmented documents are quite close to their originals, as shown in Fig. 13. We suspect this might be a limitation with our approach of embedding relations and less so with the augmentations themselves. In future work, one could try to embed relations differently, e.g., by including their type, or building natural language sentences from them, such as "*registered uses a claim*" from the *uses* relation *registered (Activity)* → *a claim (Activity Data)*.

5.4 Linguistic variability, mention length, and relation direction

To supplement the visualizations from Sect. 5.3, we define three additional characteristics of textual process descriptions that are changed by the data augmentation techniques we selected. These characteristics are as follows.

5.4.1 Linguistic variability

Increased linguistic variability, i.e., augmented text, uses a larger vocabulary to describe the same, or at least, a similar business process[2]. The most prominent examples for such techniques are the *Back-Translation* techniques. These use a large language model, e.g., BERT [11] to translate the process description to a different language and subsequently translate it back to the original language—here English. Since data augmentation techniques must not alter the annotations of entities, we only translated spans of text, not the entire document at once. Take, for example, the running example *After a claim is registered, it is examined.* Here four spans are annotated as entities—*a claim*, *registered*, *examined*, and *it*. Additionally, there are three remaining spans that do not correspond to entities: *After a*, *is*, *is*. By back-translating these seven spans separately, we obtain variation in their wording (*surface form*), but are still able to preserve annotations. Samples synthesized in this way are especially useful for making methods for the MD task generalize better and more robust.

5.4.2 Span length variations

Many spans of a given entity type, e.g., Actors, are very uniform in length across examples. This is a result of several factors, but most apparent actors are often identified by their job title, e.g., *the clerk*, or the department, e.g., *the secretary office*. These titles and departments are very short phrases, and longer ones are abbreviated, reducing their length to two or less tokens, e.g., *the MPOO*. Even though their expanded form may not be known, expanding some of these spans to suitable phrases, e.g., *Manager, Post Office Operations*, creates samples with longer surface forms. This, in turn, may improve the robustness of the MD extractors, as well as the generalization capabilities of RE methods.

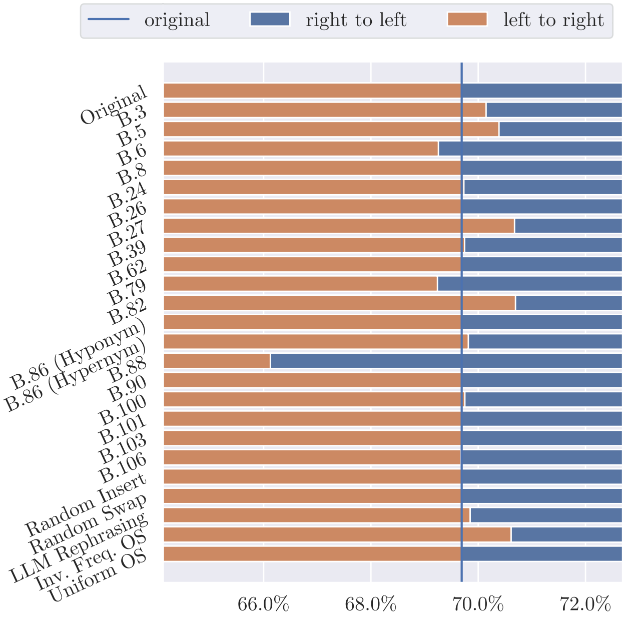


Fig. 14 Effects of techniques on relation direction. Techniques that do not change the ratio between relation directions are omitted for clarity. Oversampling is abbreviated to *OS*

5.4.3 Direction of relations

The order of appearance for mentions that form a relation is very uniform in the current version of PET. This is especially apparent, when looking at the baseline extraction rules defined by the original authors of PET: Here the order of appearance of Activities and Actors is exploited, to form the *Actor Performer* and *Actor Recipient* relations [7]. These relations define the Actor that performs an Activity, and the Actor, on which an Activity is performed. The Actor left of an Activity is assigned the former, while the Actor right of that Activity is assigned the latter. In this example, order uniformity can lead to less robust models, as they rely on this and subsequently make wrong predictions given different linguistic constructs. Synthesizing samples with a different order may encourage models to consider linguistic features (context) rather than just the order of mentions in a sentence during prediction.

These characteristics are visualized in Figs. 14 and 15. Figure 15 shows the "landscape" of data augmentation techniques evaluated in this paper. Three groups of techniques emerge. The first one is a group of techniques that only marginally increase the number of tokens in mentions and keep the size of the vocabulary roughly the same. These

[2] The augmentation technique might change information in the text, which changes the process overall, e.g., by replacing original actors with new, artificial ones.

techniques mainly change the context (i.e., the text that does not contain immediately process-relevant information) or the structure of the text (i.e., modify punctuation, or change the order of tokens). Techniques in the second group do not modify the vocabulary, but have a significant impact on the number of tokens in a given mention. These augmentations can theoretically be useful for the robustness of MD extraction models, but only have a moderate impact in our experiments, using the PET dataset. We count *Random Insertion*, *Filler Word Augmentation*, but also *Random Word Deletion* toward this group, see Fig. 2 for an example taken from the augmented data. The final group of techniques increases the size of the vocabulary, while keeping mention lengths roughly the same. These techniques are paraphrasing, aimed at preserving semantics and the structure of textual data. Techniques using WordNet to insert or substitute synonyms (*B.100*, *B.101*, as well as back translation methods (*B.62*, *B.26*), fall in this group.

6 Results

In this section, we will discuss results for our experiments and answer the research questions from Sect. 4.1. Table 2 lists the differences of all data augmentation techniques compared to a run on unaugmented data. All differences are measured as the micro-averaged F_1 score. Concluding from our results, we find that both the MD and RE tasks significantly benefit from some of the data augmentation techniques we selected and tested. Data augmentation results in improvements of up to +5.7% (absolute percentage points) in mention detection performance and up to +4.5% (absolute percentage points) in relation extraction performance.

4.1—Simple data augmentation. With the exception of *Random Word Deletion*, simple data augmentation techniques, i.e., *Random Insert*, and *Random Swap*, are not useful for improving mention detection results, meaning they are worse than or equal to oversampling documents. We expand on the notion of usefulness later in this section, when answering research question 4.1. This observation is even clearer for the relation extraction task, where none of the simple data augmentation techniques is able to outperform oversampling. For this reason, we answer our leading research question 4.1 with *No*. While it seems that simple data augmentation techniques improve the performance of both the MD and RE tasks, we attribute that improvement to the implicit oversampling instead of higher data quality.

4.1—Complex data augmentation. Looking at the remaining, more complex data augmentation techniques, only some of them outperform oversampling in both mention detection and relation extraction. For mention detection, we find that especially *Rephrasing* (e.g., *Antonyms Substitute (Double Negation)*) and *Repeating* (e.g., *Subsequence Substitution for Sequence Tagging*) techniques are most useful. Notably, techniques focused on acronyms, i.e., *Replace Abbreviations and Acronyms* and *Contractions and Expansions Perturbation* are worse than oversampling. This can be explained by their reliance on lists of possible acronyms, which are from many different domains, and not always applicable to the domain of a process description. For example, document *doc-10.6* of the pet dataset contains sentences like "*The MSPN sents a dismissal to the MSPO.*" These acronyms are undefined in the lists of the acronym-focused data augmentation techniques. In these cases, they operate like *Uniform Oversampling*, explaining their poor performance. Many of the based on large language models, especially back translation techniques, like *Multilingual Back Translation*, which translate a sentence fragment twice, and rephrasing techniques based on LLMs, are very time-intensive, as Table 2 shows. Yet, improvements in relation extraction performance are not significant, when comparing them to more lightweight approaches, e.g., *Synonym Substitution*, which uses WordNet to rephrase text sequences and runs several orders of magnitude faster. Juding from the observations made during our experiments, using these large language model-based methods is not worth the increase in computing power and time. While the MD task can still benefit from all data augmentation techniques, it does so to a lesser extent when compared to the RE task. This indicates a model that is already more stable and generalizes better. Transformations that alter the amount of tokens in mentions, such as *Random Word Deletion*, *Synonym Insertion*, or *Subsequence Substitution for Sequence Tagging*, result in lesser improvements, compared to paraphrasing methods, such as *AntonymsSubstitute*, *BackTranslation*, or *Synonym Substitution*. Similar to the RE task, the MD task does not benefit significantly more from resource- and time-intensive, large language model- based augmentation techniques for paraphrasing, compared to their simpler counterparts, answering our second research question 4.1. Based on our observations, we do not recommend using complex data augmentation techniques with classical supervised approaches to process information extraction. If they are useful for more complex extraction models (e.g., deep learning), it needs further research in future work.

4.1—Oversampling. Many of the data augmentation techniques we selected outperform the two oversampling strategies we investigated. For mention detection most of these techniques fall into the *Rephrasing* category, while *Adding Noise* does not seem to be useful. For the relation extraction task, a lot less data augmentation techniques are better than oversampling. We suspect this is due to a relation extraction model with sub-optimal hyper-parameter configuration. Since we kept this configuration fixed throughout our experiments and followed previous work [7, 33], oversampling may

Fig. 15 Effects on vocabulary size and the average length of mentions in tokens. Oversampling techniques are omitted, as they neither change the vocabulary size, nor the length of mentions

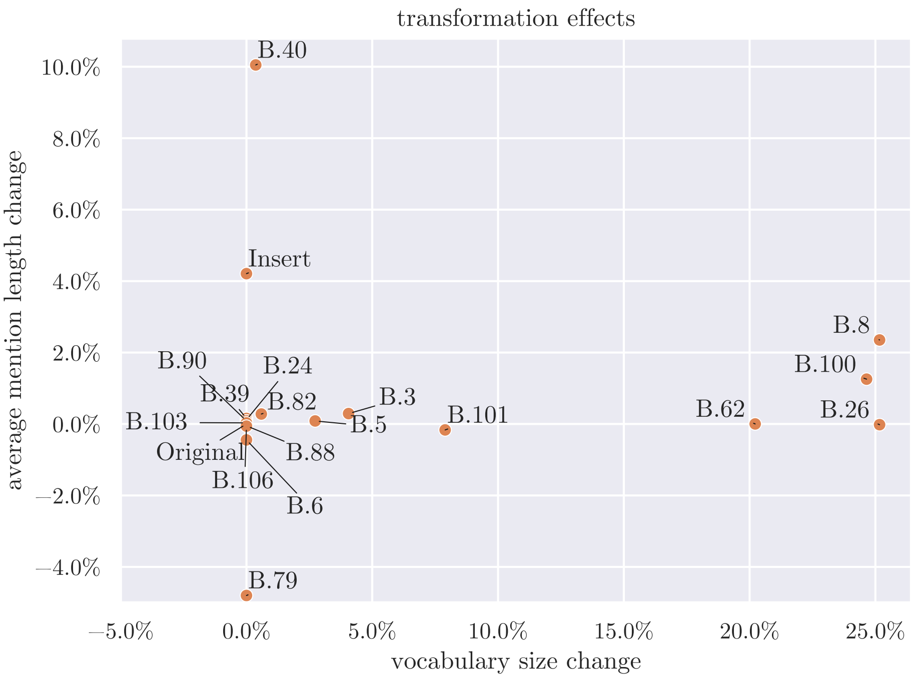


Table 2 Detailed results for all augmentation techniques. *Time* reports how long it takes each augmentation technique to augment a single document ten times. Performance improvements are listed for both the mention detection (MD) and relation extraction (RE) task. These results are the averages of a 5-fold cross-validation on the entire PET dataset, reported as the absolute increase in F_1 measure compared to a run on unaugmented training data

Technique	Time [*s*]	MD	RE
Unaugmented	—	69.5%	75.9%
Adjectives Antonyms Switch	0.008	+1.8%	+3.3%
AntonymsSubstitute (Double Negation)	1.374	+5.7%	+2.4%
Auxiliary Negation Removal	0.096	+2.2%	+3.5%
BackTranslation	13.933	+2.9%	+3.6%
Concatenate Two Random Sentences	0.009	+2.3%	+4.5%
Contextual Meaning Perturbation	6.040	+2.5%	+4.1%
Contractions and Expansions Perturbation	0.011	+0.6%	+1.9%
English Mention Replacement for NER	0.031	+2.0%	+3.8%
Filler Word Augmentation	0.078	+1.6%	+3.3%
Hypernym Replacement	2.514	+3.0%	+3.8%
Hyponym Replacement	3.406	+2.2%	+4.0%
Inverse Type Frequency Oversampling	0.011	+1.4%	+3.3%
Large Language Model Rephrasing	214.240	+3.4%	+3.1%
Lost in Translation	63.285	+1.0%	+1.3%
Multilingual Back-Translation	62.531	+1.7%	+4.1%
Random Insert	0.012	+1.3%	+3.3%
Random Swap	0.002	+2.2%	+2.6%
Random Word Deletion	0.016	+3.0%	+2.7%
Replace Abbreviations and Acronyms	0.018	+1.4%	+2.9%
Sentence Reordering	0.004	+2.4%	+4.0%
Shuffle Within Segments	0.046	+2.1%	+4.1%
Synonym Insertion	0.055	+1.6%	+4.4%
Synonym Substitution	0.057	+3.3%	+3.2%
Subsequence Substitution for Sequence Tagging	0.385	+5.3%	+3.6%
Transformer Fill	1.958	+2.1%	+3.3%
Uniform Oversampling	0.002	+2.2%	+3.5%

be compensating for a learning rate that is set too low. The experiment described in Sect. 4.5 could be expanded in future work, to consider the hyper-parameters of mention detection and relation extraction models, but was out of scope for this article.

4.1—Characteristics. Based on Sect. 5, presenting the effects of data augmentation techniques on the text of process descriptions, we see three major areas affected by different data augmentation techniques. **(1)** The embedding of entity mentions and relations between them is affected by *Rephrasing* and *Reordering* data augmentation techniques the most. This implies that the meaning of the corresponding process elements changes the most, if techniques from these categories are used. These changes sometimes also break the syntax and semantics of process descriptions, which does not appear to be a problem for the models used in our experiments, as the improvements are still outperforming oversampling strategies. **(2)** *Reordering* techniques are the only category of data augmentation that change the directions of relations. Figure 14 visualizes this fact, with technique *B.88* (*Sentence Reordering*) clearly standing out. While reordering sentences often breaks the semantics of the descriptions of the order of activity execution ("*Register claim. Then examine it*" vs. "*Then examine it. Register claim.*"), this does not seem to be an issue, as this data augmentation technique is one of the best ones for improving the performance of relation extraction. **(3)** The last characteristic changed by our selection of data augmentation techniques is the linguistic variability, i.e., the number of unique tokens (words) used to describe process elements. We observed an increase of up to 25% (Fig. 15). We cannot directly conclude from our experiments, if this is a beneficial effect or not, since the data augmentation techniques that increase the linguistic variability the most are the ones based on large language models, which do not have clear advantages compared to other techniques. Quantifying the effects of increased vocabulary in process descriptions would require further research, especially using larger deep learning models, and not just classical machine learning methods.

7 Limitations and future work

We judged the usefulness of data augmentation techniques using only one extraction pipeline based on classical (shallow) machine learning models. This mainly originates from the lack of supervised machine learning methods for the process information extraction methods. While there is previous work using deep learning models for extracting process information, the authors only investigated entity mention detection, not relation extraction [2]. It would also be interesting to investigate, if data augmentation can solve the problem of small datasets preventing the training of general purpose information extraction methods from natural language process [3], though for this article this investigation was considered out of scope.

Furthermore, our exploration of the changes in surface form of process descriptions is highly qualitative and focused on finding noteworthy occurrences of errors in the augmented data. Deeper analysis on a linguistic level could lead to more insights where data augmentation techniques introduce unwanted and unexpected perturbations, potentially detrimental to the process information extraction task.

Finally, we did not investigate the effects of data augmentation on models trained with augmented data. While we did do a black-box evaluation to judge how much certain techniques improve the extraction quality, this is merely descriptive and does not explain these improvements. In future work, we plan to analyze models trained with augmented data in terms of their robustness, resilience against adversarial examples, and generalization capabilities, compared to models trained with unaugmented data only.

Some future work we are currently interested in directly follows from these limitations, including testing augmented data for training deep learning models, analyzing augmented data on a linguistic level, and investigating why augmented data changes an extraction model's capabilities.

Additionally, we want to analyze how targeted data augmentation can be used to improve extraction of certain types of mentions or relations, tackling the problem of data imbalance. We also want to explore adaptive data augmentation, where samples are selected for augmentation by their value for model training, e.g., measured by the number of wrong predictions the cause during evaluation.

8 Conclusion

In this article, we evaluated established data augmentation techniques for use in the mention detection and relation extraction steps of extracting process-relevant information from natural language texts for use in the automated generation of business process models. To this end, we selected a total of 21 suitable methods from the NL-Augmenter framework [12]. We complemented this selection with two oversampling techniques and two simple augmentation techniques, to answer how much data augmentation can improve the quality of mentions and relations extracted from textual business process descriptions. We find that while not all data augmentation techniques are useful, i.e., improve performance more than repeating unaltered data (oversampling), many are and can improve the performance of detecting process-relevant entity mentions by up to +5.7% (absolute percentage points) and up to +4.5% (absolute percentage points) in relation extraction performance.

Additionally, we discuss several characteristics of the data that are changed by the investigated data augmentation techniques, including the change in vocabulary size of process descriptions, the direction of relations between process elements, and the change of meaning in the descriptions of processes. We complement this analysis with an extensive exploration of the surface form changes in process description texts and derive four common error classes, which are introduced by data augmentation techniques.

9 Parameters of data augmentation techniques

See (Tables 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, and 27)

Table 3 Parameters for augmentation technique *Adjective Antonyms Switch* (*B.3*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	4.70	3.60
Replaceprobability	Probability to replace an adjective with its antonym	93.1%	89.8%

Table 4 Parameters for augmentation technique *Antonyms Substitute (Double Negation)* (*B.5*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	0.98	1.52

Table 5 Parameters for augmentation technique *Auxiliary Negation Removal* (*B.6*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	1.56	4.77

Table 6 Parameters for augmentation technique *Back Translation* (*B.8*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	3.13	5.63
Replaceprobability	Probability to replace a text segment with its back-translation	98.5%	63.2%
Segment length	Minimum length of segment to be considered for translation	3	3

Table 7 Parameters for augmentation technique *Concatenate Two Random Sentences* (*B.24*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	6.52	9.80

Table 8 Parameters for augmentation technique *Contextual Meaning Perturbation* (*B.26*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	6.52	7.94
Replace probability	Probability to replace a text segment with its back-translation	31.4%	3.4%
Part ofspeech taggroups	Groups of part of speech tags to consider for back-translation, e.g., nouns, verbs, adjectives, ...	Nouns Adj. Adv.	Nouns Adj. Adv.

Table 9 Parameters for augmentation technique *Contractions and Expansions Perturbation* (*B.27*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	0.47	2.24
Replaceprobability	Probability to replace an abbreviation with its long form and vice versa	97.3%	91.9%

Table 10 Parameters for augmentation technique *English Mention Replacement for NER* (*B.39*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	8.81	6.51
Replaceprobability	Probability to replace an entity mention with one from another process description	10.7%	73.9%

Table 11 Parameters for augmentation technique *Filler Word Augmentation* (*B.40*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	5.13	7.63
Insert probability	Probability to insert any of the filler phrases	6.3%	9.1%
Insert filler phrases	Should phrases like *err*, *uhm*, *ahh* be inserted?	No	Yes
Insert speaker phrases	Should phrases like *I think*, *I mean*, *I would say* be inserted?	Yes	Yes
Insertuncertaintyphrases	Should phrases like *maybe*, *probably*, *possibly* be inserted?	No	No

Table 12 Parameters for augmentation technique *Multilingual Back Translation* (*B.62*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	3.47	4.24
Replaceprobability	Probability to replace an entity mention with its back-translation	22.3%	18.9%
pivot language	Language to translate the segment to and back	lo	ja

Table 13 Parameters for augmentation technique *Random Word Deletion* (*B.79*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	3.58	6.72
Delete probability	Probability to delete a token	12.2%	0.3%

Table 14 Parameters for augmentation technique *Replace Abbreviations and Acronyms* (*B.82*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	1.41	0.09
Replaceprobability	Probability to replace an abbreviation with its long form and vice versa	88.6%	99.6%

Table 15 Parameters for augmentation technique *Hypernym Replacement* (*B.86*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	4.30	6.01
Replaceprobability	Probability to replace a word with its hypernym	38.7%	98.0%

Table 16 Parameters for augmentation technique *Hyponym Replacement* (*B.86*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	3.20	6.87
Replaceprobability	Probability to replace a word with its hyponym	47.4%	91.7%

Table 17 Parameters for augmentation technique *Sentence Reordering* (*B.88*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	2.67	4.18

Table 18 Parameters for augmentation technique *Shuffle Within Segments* (*B.90*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	7.90	4.23
Replaceprobability	Probability to replace a segment with its shuffled version	46.2%	7.2%

Table 19 Parameters for augmentation technique *Synonym Insertion* (*B.100*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	7.90	6.23
Insert probability	Probability to insert the synonym of a word before that word	14.7%	25.7%

Table 20 Parameters for augmentation technique *Synonym Substitution* (*B.101*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	2.14	4.54
Replaceprobability	Probability to replace a word with its synonym	30.2%	10.6%

Table 21 Parameters for augmentation technique *Subsequence Substitution for Sequence Tagging* (*B.103*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	5.04	6.67
Replaceprobability	Probability to replace a segment with a segment with identical part-of-speech tags from a different document	30.2%	40.7%
Min length	Minimum length of segment to replace	1	4
Max length	Maximum length of segment to replace	5	8

Table 22 Parameters for augmentation technique *Transformer Fill* (*B.101*). Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	3.37	5.56
Replaceprobability	Probability to mask a word and fill it using a transformer model (BERT)	30.4%	38.4%
Part of speech tags	Groups of part-of-speech tags that are considered for masking and filling	Nouns	Nouns, Adj.

Table 23 Parameters for augmentation technique *Random Insert*. Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	6.04	20.3
Insert probability	Probability to insert a random token from a different document at each original token	2.4%	0.6%

Table 24 Parameters for augmentation technique *Random Swap*. Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	2.81	2.03
Swap probability	Probability to swap any two tokens in the process description	0.7%	1.5%

Table 25 Parameters for augmentation technique *Large Language Model Rephrasing*. Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Augmentation rate	Number of augmented process descriptions created per one original	4.51	6.19
Replaceprobability	Probability to replace a text segment with a rephrased version	27.7%	30.6%

Table 26 Parameters for augmentation technique *Uniform Oversampling*. Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
oversampling rate	Number of uniform randomly sampled original documents	1.46	5.08

Table 27 Parameters for augmentation technique *Inverse Type Oversampling*. Columns *MD* and *RE* refer to the parameter value for the mention detection (MD) and relation extraction task (RE)

Name	Description	MD	RE
Oversampling rate	Number of randomly sampled original documents, weighed by the rarity of contained types	4.09	5.05

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Ackermann, L., Käppel, M., Marcus, L., Moder, L., Dunzer, S., Hornsteiner, M., Liessmann, A., Zisgen, Y., Empl, P., Herm, L.-V., et al.: Recent advances in data-driven business process management. arXiv preprint arXiv:2406.01786 (2024)
2. Ackermann, L., Neuberger, J., and Jablonski, S.: Data-driven annotation of textual process descriptions based on formal meaning representations. In CAiSE (2021)
3. Ackermann, L., Neuberger, J., Käppel, M., and Jablonski, S.: Bridging research fields: An empirical study on joint, neural relation extraction techniques. In CAiSE (2023)
4. Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 2623–2631 (2019)
5. Bellan, P., Dragoni, M., and Ghidini, C.: Assisted process knowledge graph building using pre-trained language models. In Proceedings of AIxIA 2022 - Advances in Artificial Intelligence (2022)
6. Bellan, P., Dragoni, M., and Ghidini, C.: Extracting business process entities and relations from text using pre-trained language models and in-context learning. In EDOC (2022)
7. Bellan, P., Ghidini, C., Dragoni, M., Ponzetto, S. P., and van der Aa, H.: Process extraction from natural language text: the PET dataset and annotation guidelines. In NL4AI (2022)
8. Bellegarda, J.R.: Exploiting latent semantic information in statistical language modeling. Proceed. IEEE **88**(8), 1279–1296 (2000)
9. Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B.: Algorithms for hyper-parameter optimization. Advances in Neural Information Processing Systems 24 (2011)
10. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O.: Translating embeddings for modeling multi-relational data. Advances in neural information processing systems 26 (2013)
11. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
12. Dhole, K. D., Gangal, V., Gehrmann, S., Gupta, A., Li, Z., Mahamood, S., Mahendiran, A., Mille, S., Shrivastava, A., Tan, S., et al.: Nl-augmenter: a framework for task-sensitive natural language augmentation. arXiv preprint arXiv:2112.02721 (2021)
13. Eldin, A. N., Assy, N., Anesini, O., Dalmas, B., and Gaaloul, W.: A decomposed hybrid approach to business process modeling with llms
14. Erdengasileng, A., Han, Q., Zhao, T., Tian, S., Sui, X., Li, K., Wang, W., Wang, J., Hu, T., Pan, F., et al.: Pre-trained models, data augmentation, and ensemble learning for biomedical information extraction and document classification. Database 2022, baac066 (2022)
15. Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E.: A survey of data augmentation approaches for NLP
16. Ferreira., R. C. B., Thom., L. H., and Fantinato., M.: A semi-automatic approach to identify business process elements in natural language texts. In ICEIS (2017)
17. Friedrich, F., Mendling, J., and Puhlmann, F.: Process model generation from natural language text. In CAiSE (2011)
18. Günther, M., Ong, J., Mohr, I., Abdessalem, A., Abel, T., Akram, M. K., Guzman, S., Mastrapas, G., Sturua, S., Wang, B., Werk, M., Wang, N., and Xiao, H.: Jina embeddings 2: 8192-token general-purpose text embeddings for long documents, (2023)
19. Grohs, M., Abb, L., Elsayed, N., and Rehse, J.-R.: Large language models can accomplish business process management tasks. In International Conference on Business Process Management, Springer, pp. 453–465 (2023)
20. Jiang, Z., Han, J., Sisman, B., and Dong, X. L.: Cori: Collective relation integration with data augmentation for open information extraction. arXiv preprint arXiv:2106.00793 (2021)
21. Kampik, T., Warmuth, C., Rebmann, A., Agam, R., Egger, L. N., Gerber, A., Hoffart, J., Kolk, J., Herzig, P., Decker, G., et al.: Large process models: Business process management in the age of generative ai. arXiv preprint arXiv:2309.00900 (2023)
22. Käppel, M., and Jablonski, S.: Model-agnostic event log augmentation for predictive process monitoring. In International Conference on Advanced Information Systems Engineering, Springer, pp. 381–397 (2023)
23. Käppel, M., Schönig, S., and Jablonski, S.: Leveraging small sample learning for business process management. Information and Software Technology (2021)
24. Klievtsova, N., Benzin, J.-V., Kampik, T., Mangler, J., and Rinderle-Ma, S.: Conversational process modelling: state of the art, applications, and implications in practice. In International Conference on Business Process Management, Springer, pp. 319–336 (2023)
25. Köpke, J., and Safan, A.: Introducing the bpmn-chatbot for efficient llm-based process modeling
26. Kourani, H., Berti, A., Schuster, D., and van der Aalst, W. M.: Process modeling with large language models. arXiv preprint arXiv:2403.07541 (2024)
27. Kourani, H., Berti, A., Schuster, D., and van der Aalst, W. M.: Promoai: Process modeling with generative AI. arXiv preprint arXiv:2403.04327 (2024)
28. Liu, J., Chen, Y., and Xu, J.: Machine reading comprehension as data augmentation: a case study on implicit event argument extrac-

tion. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 2716–2725 (2021)

29. López, H. A., Strømsted, R., Niyodusenga, J.-M., and Marquard, M.: Declarative process discovery: Linking process and textual views. In International Conference on Advanced Information Systems Engineering (2021)
30. López-Acosta, H.-A., Hildebrandt, T., Debois, S., and Marquard, M.: The process highlighter: From texts to declarative processes and back. In CEUR Workshop Proceedings, CEUR Workshop Proceedings, pp. 66–70 (2018)
31. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
32. Mohammed, R., Rawashdeh, J., and Abdullah, M.: Machine learning with oversampling and undersampling techniques: overview study and experimental results. In 2020 11th international conference on information and communication systems (ICICS), IEEE, pp. 243–248 (2020)
33. Neuberger, J., Ackermann, L., and Jablonski, S.: Beyond rule-based named entity recognition and relation extraction for process model generation from natural language text. In CoopIS (2023)
34. Neuberger, J., Ackermann, L., van der Aa, H., and Jablonski, S.: A universal prompting strategy for extracting process model information from natural language text using large language models. In International Conference on Conceptual Modeling, Springer, pp. 38–55 (2024)
35. Neuberger, J., Doll, L., Engelmann, B., Ackermann, L., and Jablonski, S.: Leveraging data augmentation for process information extraction. In International Conference on Business Process (2024)Modeling, Development and Support, Springer, pp. 57–70 (2024)
36. Poličar, P.G., Stražar, M., Zupan, B.: Opentsne: a modular python library for t-sne dimensionality reduction and embedding. J. Stat. Softw. **109**(3), 1–30 (2024)
37. Quishpi, L., Carmona, J., and Padró, L.: Extracting annotations from textual descriptions of processes. In BPM 2020 (2020)
38. Radford, A.: Improving language understanding by generative pretraining
39. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. J. Big Data **6**(1), 1–48 (2019)
40. Shorten, C., Khoshgoftaar, T. M., and Furht, B.: Text data augmentation for deep learning. Journal of big Data (2021)
41. van der Aa, H., Di Ciccio, C., Leopold, H., and Reijers, H. A.: Extracting declarative process models from natural language. In CAiSE (2019)
42. Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J., and Sun, M.: Docred: A large-scale document-level relation extraction dataset. arXiv preprint arXiv:1906.06127 (2019)
43. Zoran, D., and Weiss, Y.: Scale invariance and noise in natural images. In 2009 IEEE 12th International Conference on Computer Vision, IEEE, pp. 2209–2216 (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Julian Neuberger is a research assistant at the University of Bayreuth, where he obtained his PhD in computer science in April of 2025 for his work on extracting business process model information from natural language text descriptions. He currently works on improving and facilitating the collaborative collection of training data for information extraction. Among others, his research interests especially lie in the application of natural language processing for conceptual modelling tasks. Julian has published academic papers in many internationally recognized venues, including CaiSE, ER, BPM, and CoopIS.

Lars Ackermann is a full professor of Process Mining at the Department of Computer Science of Hof University of Applied Sciences. Prior to this, he held a postdoctoral position at the University of Bayreuth, where he also earned his doctorate in Computer Science in 2018. His research is situated at the intersection of Business Process Management and Artificial Intelligence, in particular using unstructured data in process mining, process model generation, and predictive business process monitoring. He has published in leading conferences such as BPM, CAiSE, CoopIS, and ER, serves as reviewer for top-tier venues including BPM, Neurocomputing, Business & Information Systems Engineering (BISE), and is an associate editor for Wirtschaftsinformatik (WI).

Stefan Jablonski is a Full Professor of Computer Science with the Institute for Computer Science at University of Bayreuth (Germany). He is the head of the chair for Databases and Information Systems since 2006, after having a full professorship at the University of Erlangen-Nürnberg since 1994. His major research interests include Business Process Management, flexible process enactment technologies and metamodeling. He contributed numerous publications in high-impact journals and conferences in these fields and participated in many national and international BPM projects, both in research and industry.