Extracting Process Models from Textual Descriptions: Do We Really Need LLMs?

Han van der Aa

SummerSOC 2026

About Me - Etymology

- My first name is “Han” (or technically Johannes), last name “van der Aa”;
- The Aa is a river (or actually there are many with that name in the Netherlands and Germany), so the last name means “from the river Aa”
- Fun fact: in the Netherlands we do not consider the *intersertion* (e.g., ”van der”) when sorting alphabetically, so it’s hard to beat me on that count.

Liste der Gewässer namens Aa

13 Sprachen

Artikel Diskussion Lesen Bearbeiten Quelltext bearbeiten Versionsgeschichte Werkzeuge

Aa, im Alpenraum auch Ache (althochdeutsch *Aha* „Ache“, siehe dort, vgl. schwedisch und dänisch *Å*, „Wasser, Fluss“, verwandt mit dem lateinischen *aqua*) ist ein häufiger Flussname im deutsch-niederländischen Sprachraum und in der Deutschschweiz für Flüsse oder Bäche. In Deutschland ist dieser Name auf Westfalen konzentriert. Bei Flüssen in Niedersachsen heißt der entsprechende Name oder Namensbestandteil zumeist *Aue* oder *-au*. Sehr viele Flüsse und Bäche in Schleswig-Holstein heißen *Au* mit irgendeiner Zusatzbezeichnung, in der dänischen Namensversion ist das dann *Å* mit Zusatzbezeichnung. Die altfriesische Entsprechung lautete *ē*, in Ostfriesland ist daher die Bezeichnung Ehe für viele Wasserläufe gebräuchlich.

Einige Seen wurden nach Flüssen mit dem Namen *Aa* benannt (siehe Aasee).

Auch der Name der Stadt Aachen ist auf diesen althochdeutschen Wortstamm zurückzuführen.

Outline

- Part 1 — What is text-to-process extraction, and why is it hard?

- Part 2 — Do we really need LLMs?

  CAISE 2026 paper: „Enabling Small Language Models for Text-to-Process Extraction: Balancing Accuracy and Efficiency through Distant Supervision“

Part 1 — Text-to-Process Extraction

What it is, why it matters, and how it is done

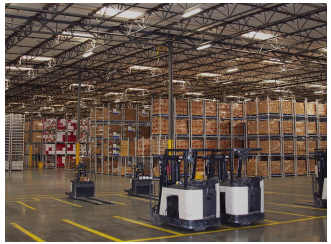
universität wien

Context: Business Process Management

Business Process Management

- Discipline that analyzes **how work is performed** in organizations
- Ensuring **consistent outcomes** and using **improvement opportunities**


Manufacturing


Logistics


Services

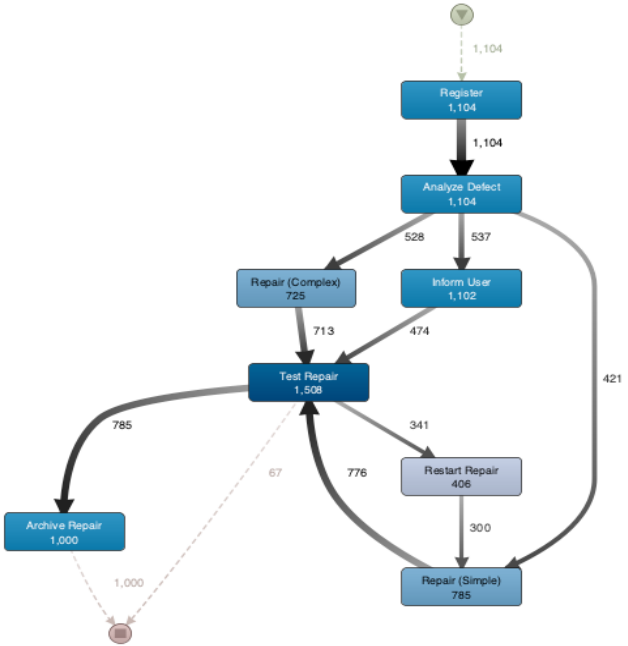

Healthcare

Business Process

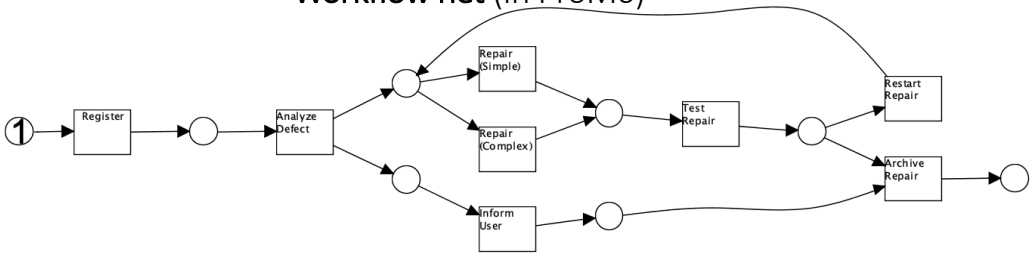
- A chain of **related activities, events and decisions**
- Leads to an **outcome that is of value** to an organization or its customers

Modeling Business Processes

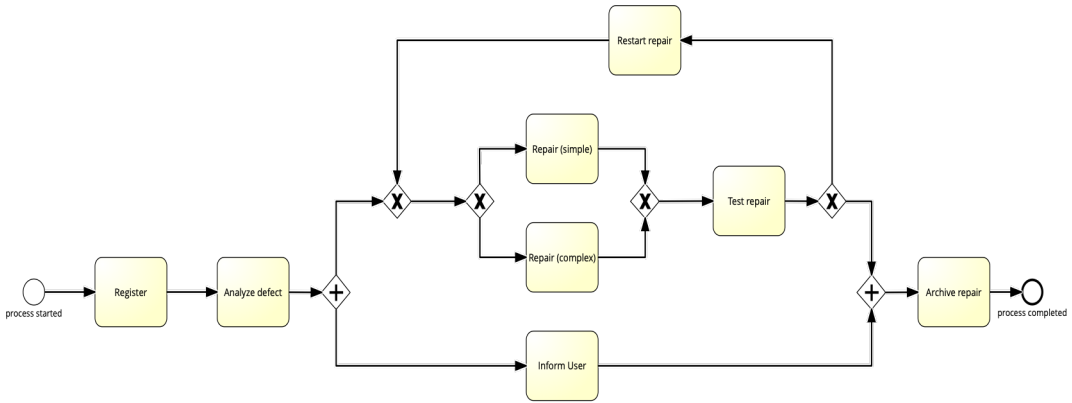
Process Map (in *DISCO* tool)



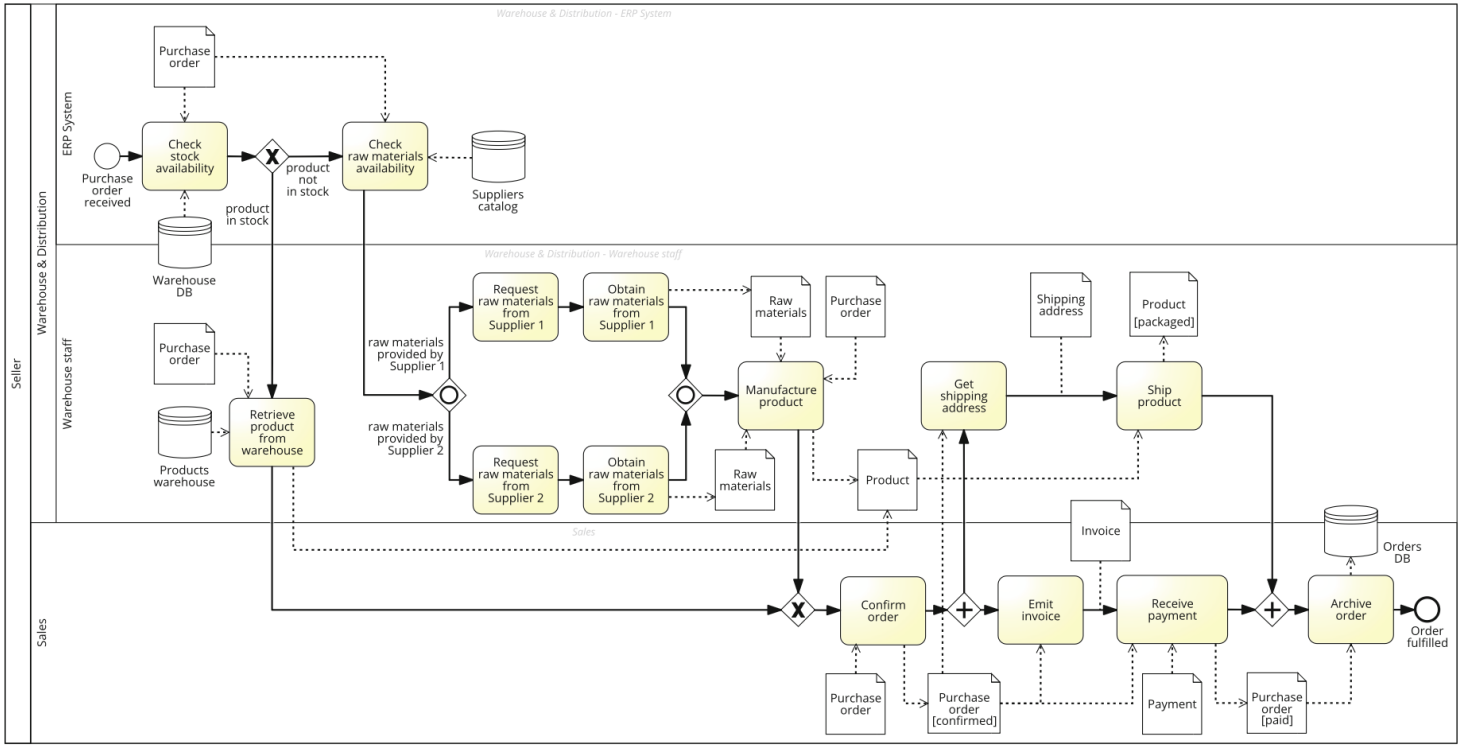
Workflow net (in ProM6)



BPMN model (in Signavio)



A More Complex Model



Source: Dumas et al. - Fundamentals of Business Process Management

What Are Process Models For?

Communication
A shared language for business, analysts and IT to agree on how work is done.

Analysis
Locate bottlenecks, risks and compliance issues before they cause problems.

Redesign & improvement
Compare as-is and to-be processes to make work faster and cheaper.

Automation
Serve as blueprints for workflow engines and information systems (BPMS).

The Downside: Who Wants to (or Can) Model?

Establishing process models **is a time-consuming and complex task**, which requires modeling expertise that are is often **not available with domain experts**

Therefore, processes are **often initially described in more accessible formats**, such as textual process descriptions (or in the age of AI: chats), whereas organizations often already have a wealth of process-related information in textual documents.

However, **such descriptions are not suitable for most downstream use cases**, such as precise communication, formal analysis, etc.

What is Text-to-Process Extraction?

- **Goal:** turn an unstructured *textual process description* into a *process model*.
- **Input:** natural-language text (manuals, SOPs, regulations, e-mails, …)
- **Output:** an imperative or declarative process model
- Bridges the gap between how processes are *described* and how they can be *analyzed and executed*.

(1) A small company manufactures customized bicycles.

(2) Whenever the sales department receives an order, a new process instance is created.

(3) A member of the sales department can then reject or accept the order for a customized bike.

(4) If the order is is accepted, the order details are entered into the ERP system.

(5) Then, the storehouse and the engineering department (S&E) are informed.

(6) The storehouse immediately processes the part list of the order.

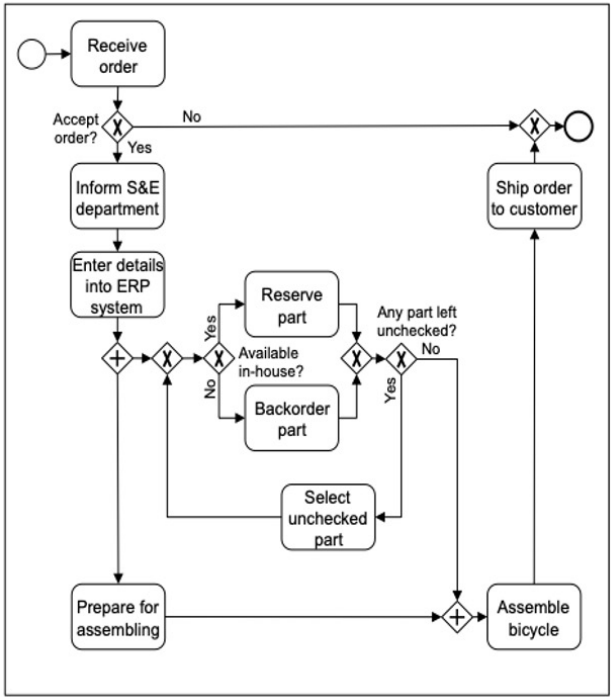
(7) If a part is available, it is reserved.

(8) If it is not available, it is back-ordered.

(9) This procedure is repeated for each item on the part list.

(10) In the meantime, the engineering department prepares everything for the assembling of the ordered bicycle.

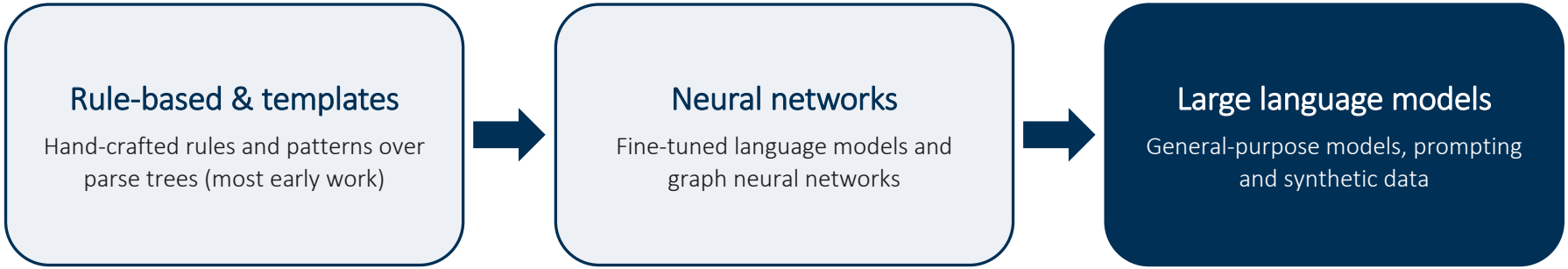
(11) If the storehouse has successfully reserved or back-ordered every item of the part list and the preparation activity has finished, the engineering department assembles the bicycle.



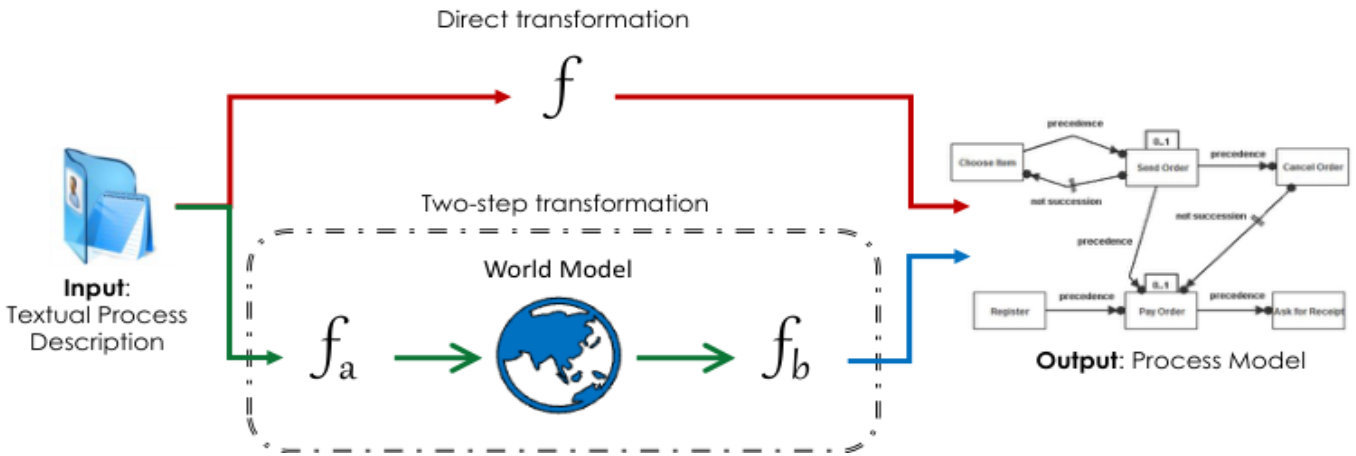
A Well-Studied but Hard Problem

- **Text-to-Process Extraction is hard.** Texts can occur in all kinds of shapes and formats (process oriented or not), have a wide range of writing styles, be ambiguous or incomplete, etc.

- Therefore, there is a range of research on the development of extraction techniques



Two Ways to Extract a Model: Direct vs. Two-Step



- **Direct (*f*):** map text straight to a process model — powerful but hard to generalise.
- **Two-step (*f*a ∘ *f*b):** first extract elements into a *world model*, then build the diagram — modular, and the stronger performer.

Figure adapted from Bellan et al. (2023), Process Extraction from Text.

Two-Step Transformation: Text Extraction/Annotation

- **Named Entity Recognition (NER):** find the process elements in the text — activities, actors, activity data, gateways, conditions.

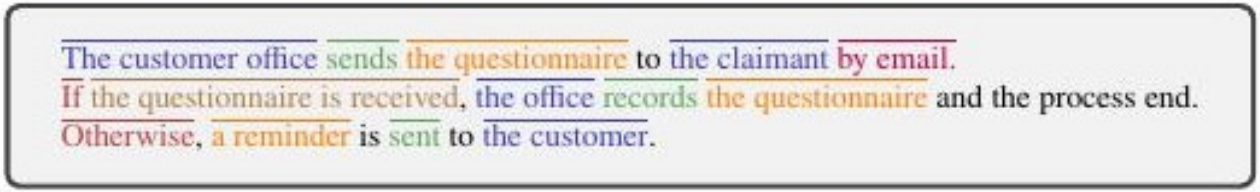


Fig. 2: The example shows an example of a procedural text fragment with process elements annotated as follows: *Activity*, *Activity Data*, *Actor*, *Further Specification*, *Gateway*, *Condition Specification*.

- **Relation Extraction (RE):** link those elements — who performs an activity (performer), what data it uses (uses), and the control flow (flow).
  - *Use*: Sends -> The questionnaire
  - *Actor performer:* The customer office -> sends
    Actor receiver: sends -> the claimant

Benefits of Two-Step Transformation

The two-step route has several benefits:

- **It uses language models for what they are good at:** analyzing text, while leaving the model generation step to dedicated algorithms
- **It makes text extraction agnostic to the target modeling language**, allowing for better generalization (and accuracy)
- **It breaks up the problem into well-defined NLP tasks** (i.e., Named Entity Recognition, Relation Extraction, Co-reference resolution)
- Crucially, this **lets us evaluate each sub-task separately against a gold standard**, making it possible to compare extraction approaches

Part 2

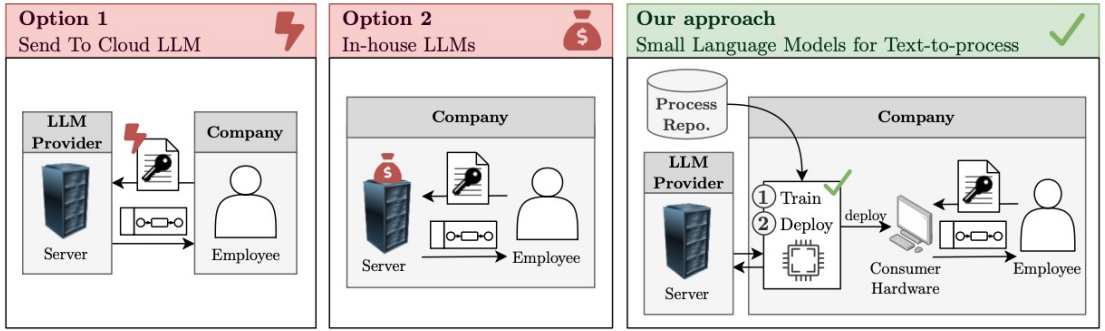
Enabling Small Language Models for Text-to-Process Extraction: Balancing Accuracy and Efficiency through Distant Supervision

Julian Neuberger[1], Han van der Aa[2], Ivan Khrop[3], and Hugo A. López[1]

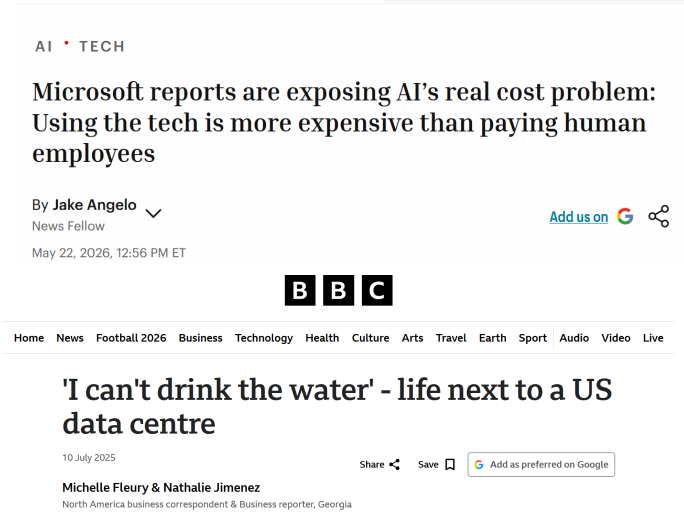
[1] Technical University of Denmark, Denmark
jtone@dtu.dk,hulo@dtu.dk
[2] University of Vienna, Vienna, Austria
han.van.der.aa@univie.ac.at
[3] University of Bayreuth, Bayreuth, Germany
ivan.khrop@uni-bayreuth.de

universität wien

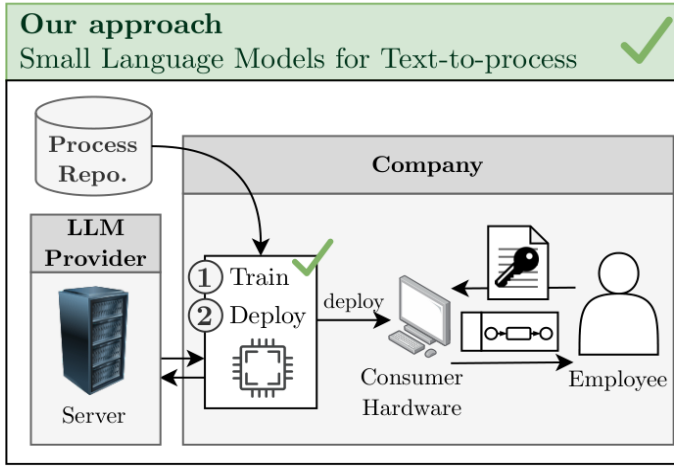
Why Not Just Use LLMs?



- LLMs reach high accuracy for text-to-process extraction — but three barriers block real-world adoption.
- **Privacy & compliance:** cloud LLMs require sending confidential process descriptions to external providers.
- **Hardware cost:** local LLMs that rival cloud models need expensive, specialized hardware; consumer-grade LLMs lag far behind.
- **Sustainability:** repeated LLM inference carries a real energy and water footprint.



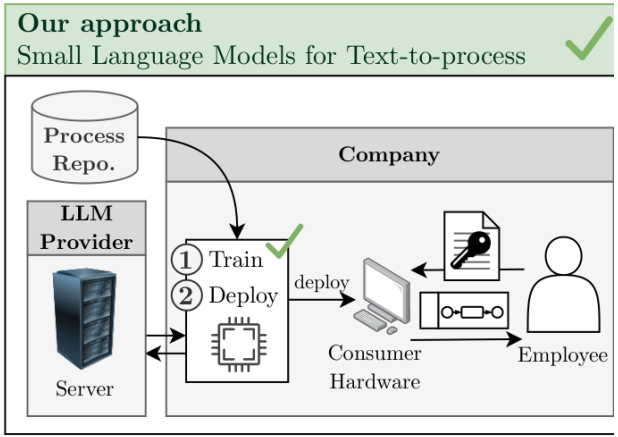
The Catch: Small Models Need Labeled Data, Which We Lack



- **Smaller machine-learning models would sidestep all three barriers — but they need labeled training data.**
- Creating that data demands scarce BPM expertise and is slow and costly.
- The popular PET dataset required three BPM experts to annotate just 45 texts.
- Thus, we need to train small models without hand-labelling thousands of documents.

Our Solution: Using Distant Supervision

- **Distant Supervision (DS):** Distant supervision is an ML technique that automatically generates large amounts of labeled training data by aligning existing knowledge bases with unstructured text.
- DS bypasses expensive manual labeling but introduces inherent "noise"
- Traditionally the source is a knowledge base; in our case, we turn to existing repositories of process models.
- **In summary:** We use LLMs to automatically turn process models into textual descriptions and then to annotate these texts with necessary labels. Creating training data at scale.
- **Key shift:** the LLM runs just once to create data, can be on non-sensitive descriptions — we never use LLMs for inference.



First Idea: Just Let an LLM Describe the Model

The obvious approach: have an LLM freely describe each BPMN model, then annotate that text.

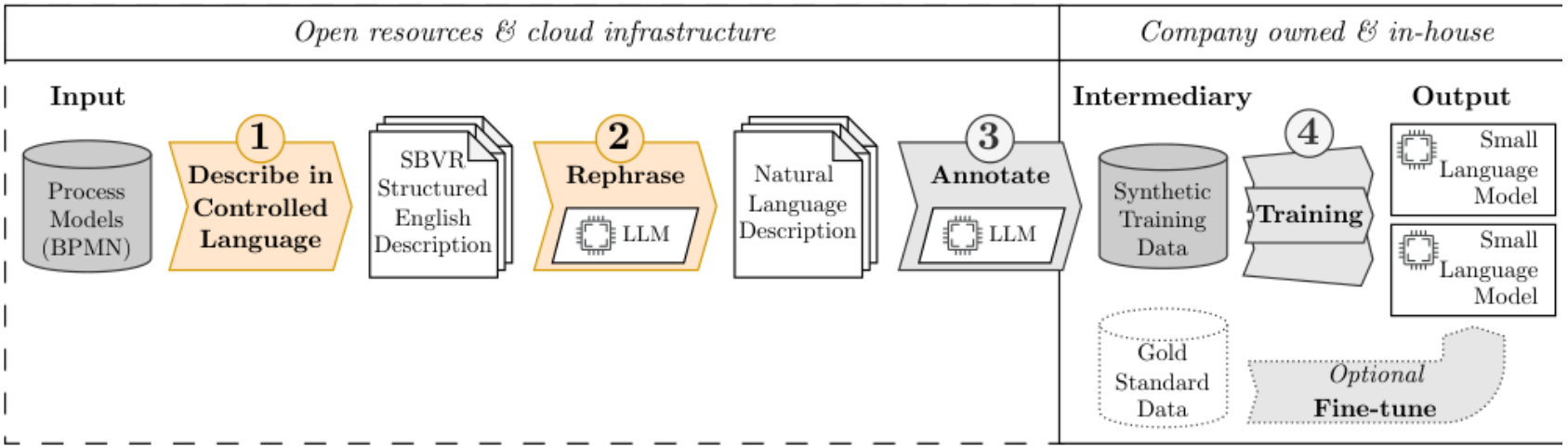
Problem: free-form descriptions leak modeling jargon into the text.

> *"[…] Next, there is an XOR-gateway with two paths […]"*

Such phrasing never appears in real process descriptions — and it confuses the downstream extractor.

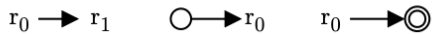
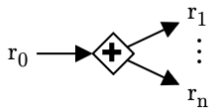
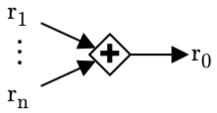
We need descriptions that are natural yet complete → a controlled description (next).

Approach Overview: From BPMN Models to Trained SLMs

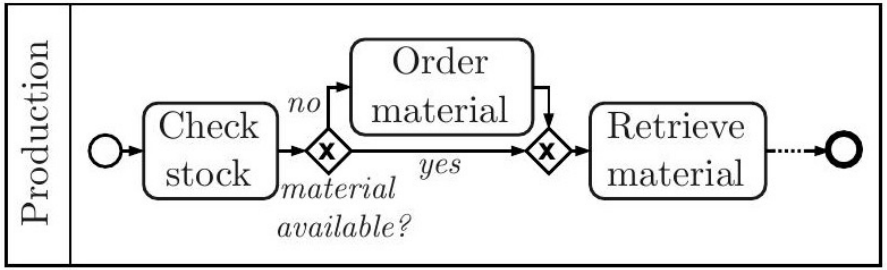


Stage 1: Describe Models in Controlled Language (SBVR)

- We first turn each BPMN model into a Controlled Natural Language (CNL) description using SBVR Structured-English templates.
- One template per control-flow pattern — 11 in total (sequence, exclusive choice, loops, …), located by graph matching.

Workflow P.	Search Pattern	Templates
Sequence		It is obligatory that r_1 after r_0
Parallel Split		It is obligatory that **all of** r_1 ... **and** r_n after r_0
Synchron-ization		It is obligatory that r_0 after **all of** r_1 ... **and** r_n

Stage 1 Example: BPMN Pattern → SBVR Rule



⇓ Sequence Pattern

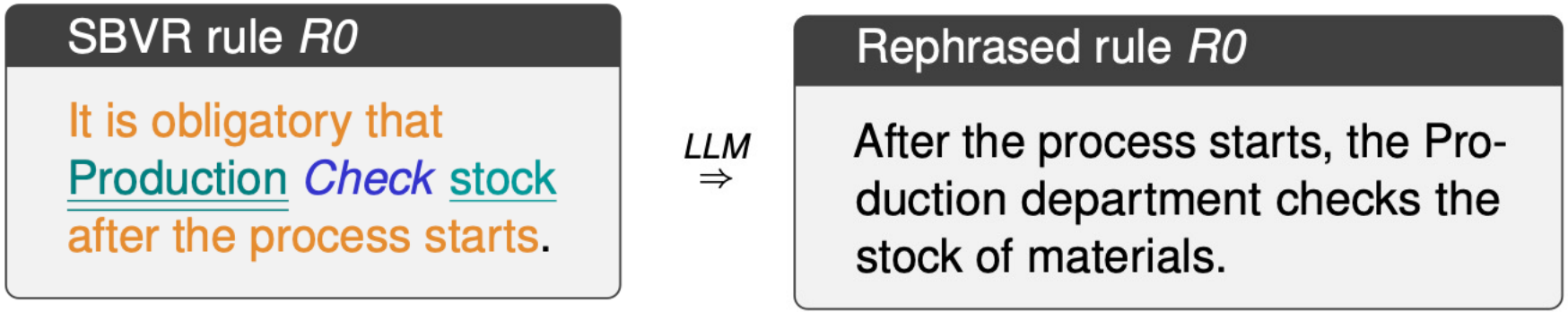
SBVR rule *R0*

It is obligatory that Production *Check* stock after the process starts.

A Sequence pattern becomes SBVR rule R0.

Stage 2: Rephrase into Natural Language

- CNL is complete but repetitive and stilted — real descriptions are fluent and ambiguous.
- We prompt an LLM to rephrase each rule as a style-transfer task, using different domain styles (handbook, legal, medical).



Stage 3: Automatically Annotate Entities & Relations

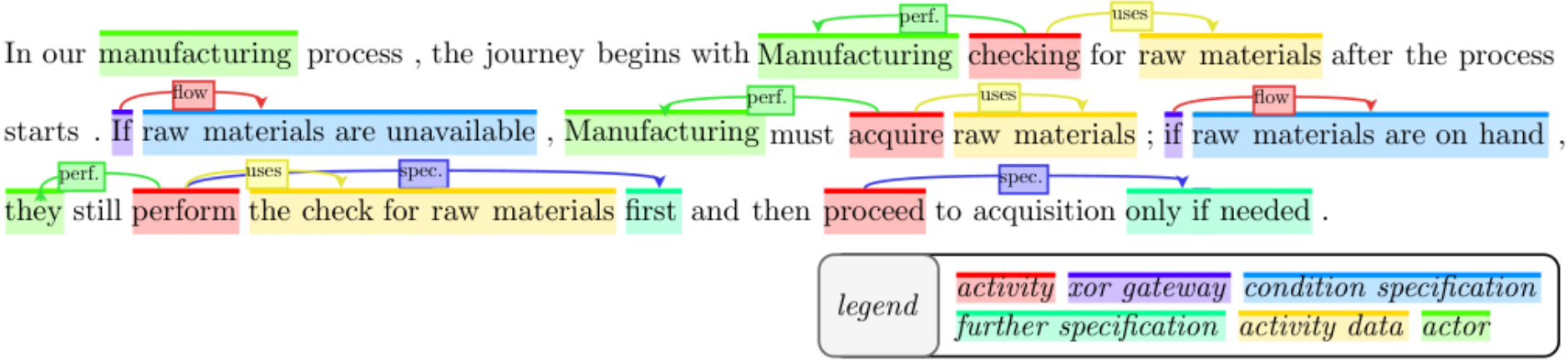
- A LLM labels the rephrased text with entities (activities, actors, data) and their relations (performer, uses, flow) in a single zero-shot prompt.
- Following distant supervision, we add annotation hints: the SBVR description with XML-like tags marking the expected activities and actors.
- **This guides the annotator and yields synthetic, automatically-labelled training data.**

Annotation hint

It is obligatory that <actor> Production </actor> <activity> Check stock </activity> after the process starts.

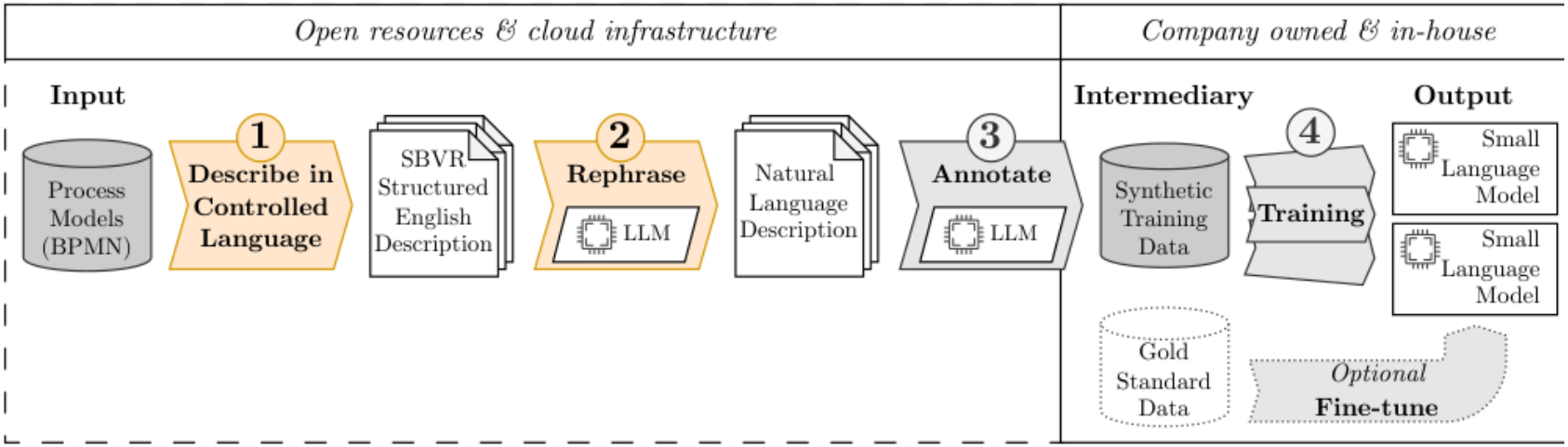
Quality of the Synthetic Data

Automatically generated data contains errors — mislabels, missing flows, over-specification — tolerated because volume compensates.



Concrete errors above: “manufacturing” mislabeled; a missing flow between checking and the XOR gateway; over-specification (“first”, “only if needed”).

Stage 4: Train and Optionally Fine-Tune the SLM



- We pre-train the SLM on the synthetic DS data — it learns basic process concepts despite the noise.
- If gold-standard data is available, an optional fine-tuning step corrects residual errors.
- The same synthetic dataset trains multiple SLMs — new models can be adopted at no extra data-generation cost.

Evaluation Setup

Evaluation goals:

- **External validity:** can SLMs trained only on our synthetic data match LLMs on real, human-annotated text (PET)?
- **Efficiency:** how do SLMs compare to LLMs in speed, memory, and energy?
- **Internal validity:** which parts of the approach actually matter? (ablation studies)

Experimental setup:

- **Source models:** SAP-SAM (1,021,471 BPMN models) → 4,505 selected via 7 general + 4 structural quality requirements.
- **Data generation (offline, once):** GPT-5-nano rephrases; GPT-5-mini annotates.
- **SLMs:** PIQN & ACE (entity recognition); UniRel & PL-Marker (relation extraction).
- **Evaluation:** PET gold standard (45 texts), micro-F1, fine-tune on 80% / test on 20%, averaged over 5 seeds.
- **Baselines:** local Llama-3 (1B–70B) and GPT-4o; we also measure time, GPU memory and energy.

Results: SLMs Match LLM Accuracy on Entity Recognition

Production department checks

	Extraction Model	Avg. F_1 Score and std. dev.		Time per document	Size on GPU	Energy per document
SLMs	ACE	74.2%	± 4.7%	**4.9s**	10.7GB	**0.12Wh**
	PIQN	**80.7%**	± 3.0%	5.8s	**2.2GB**	0.13Wh
LLMs	GPT-4o	80.1%	± 9.2%	10.0s	*N/A*	1.22Wh†
	Llama3.3 70B	70.8%	± 11.4%	96.1s	43.5GB	7.70Wh
	Llama3.1 70B	72.4%	± 13.5%	50.1s	41.1GB	3.88Wh
	Llama3.2 3B	47.3%	± 12.4%	6.4s	4.1GB	0.46Wh
	Llama3.2 1B	25.2%	± 17.1%	12.1s	3.0GB	0.79Wh

Take-away: small in-house models (PIQN 80.7%, ACE 74.2%) match cloud GPT-4o (80.1%) — at a fraction of the energy and with no data leaving the company.

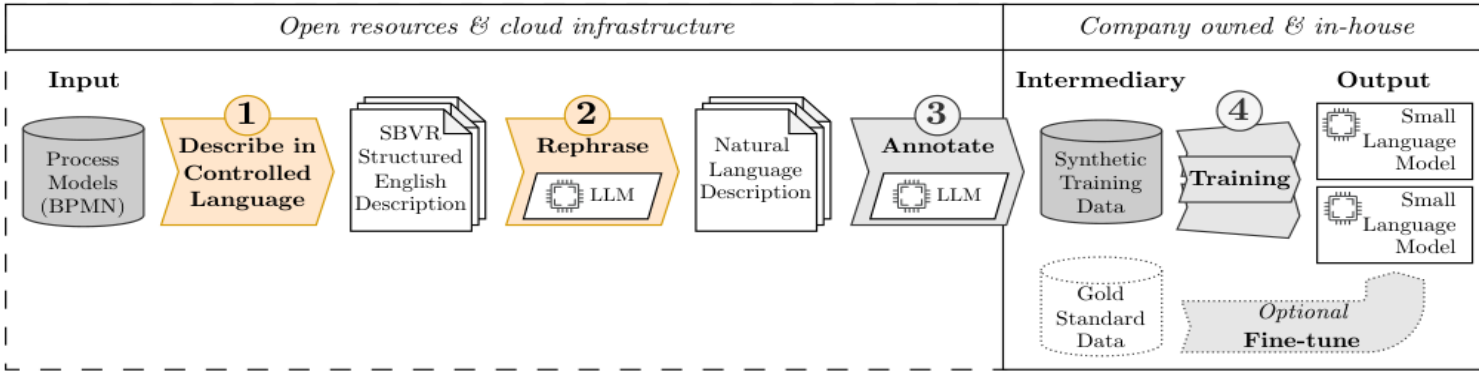
Results: SLMs Stay Competitive on Relation Extraction

Extraction Model	*Avg. F_1 Score and std. dev.*	*Time per document*	*Size on GPU*	*Energy per document*
PL-Marker	83.3% ± 4.7%	1.4s	5.1GB	0.06Wh
UniRel*	44.6% ± 6.8%	**0.6s**	8.5GB	**0.01Wh**
GPT-4o	**91.0%** ± 7.7%	3.85s	*N/A*	1.22Wh†
GPT-4o*	69.8% ± 16.6%	15.6s	*N/A*	2.43Wh†
Llama3.3 70B	84.1% ± 10.0%	34.8s	43.5GB	2.18Wh
Llama3.1 70B	82.3% ± 12.9%	26.2s	41.1GB	2.13Wh
Llama3.2 3B	28.9% ± 11.6%	2.78s	4.1GB	0.18Wh
Llama3.2 1B	13.3% ± 9.5%	2.31s	**3.0GB**	0.12Wh

Take-away: PL-Marker (83.3%) approaches 70B Llama models while running far faster (1.4s vs. ~30s/doc) at ~0.06 Wh per document.

universität wien

Ablations: What Actually Matters



- **Fine-tuning matters:** without it, entity F1 drops ~10% (PIQN −10.4%, ACE −10.5%) and PL-Marker falls −13.2%.
- **Annotation hints (to the LLM) matter:** removing them costs UniRel 15% F1.

Threats to Validity

- Only four SLMs, all encoder-only transformers, selected from non-BPM benchmarks — may bias model selection.
- Default hyper-parameters, no tuning — reported scores are a lower bound.
- Energy measured via nvidia-smi (~5% deviation, 10 Hz) — precise readings need hardware instrumentation.

Concluding Remarks

Threats to Validity:

- Only four SLMs, all encoder-only transformers
- Default hyper-parameters, no tuning — reported scores are a lower bound.
- Energy measured via nvidia-smi (~5% deviation, 10 Hz) — precise readings need hardware instrumentation.
- There is also limited evaluation data..

Take-aways:

- Small, locally-trained models can approach the performance of LLMs for text-to-process extraction — without the cost, privacy risk, or energy footprint.
- Our approach uses LLMs once, offline, to synthesize training data; reserve LLMs for where you truly need them.

Thank You — Questions Welcome

- Code & data: github.com/JulianNeuberger/slm-for-process-extraction
- Contact: han.van.der.aa@univie.ac.at

References

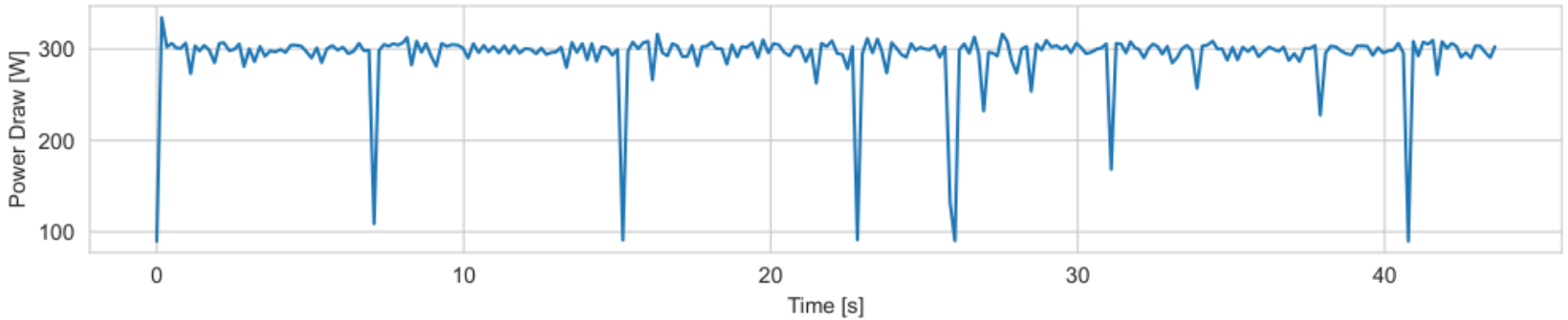
- Dumas, M., Rosa, M. L., Mendling, J., & Reijers, H. A. (2013). Fundamentals of business process management. Springer.
- Bellan, P., Ghidini, C., Dragoni, M., Ponzetto, S. P., & van der Aa, H. (2022). Process extraction from natural language text: The PET dataset and annotation guidelines. In NL4AI.
- Bellan, P., van der Aa, H., Dragoni, M., Ghidini, C., & Ponzetto, S. P. (2022, September). PET: an annotated dataset for process extraction from natural language text tasks. In International Conference on Business Process Management (pp. 315-321). Cham: Springer International Publishing.

Paper pre-print

Backup

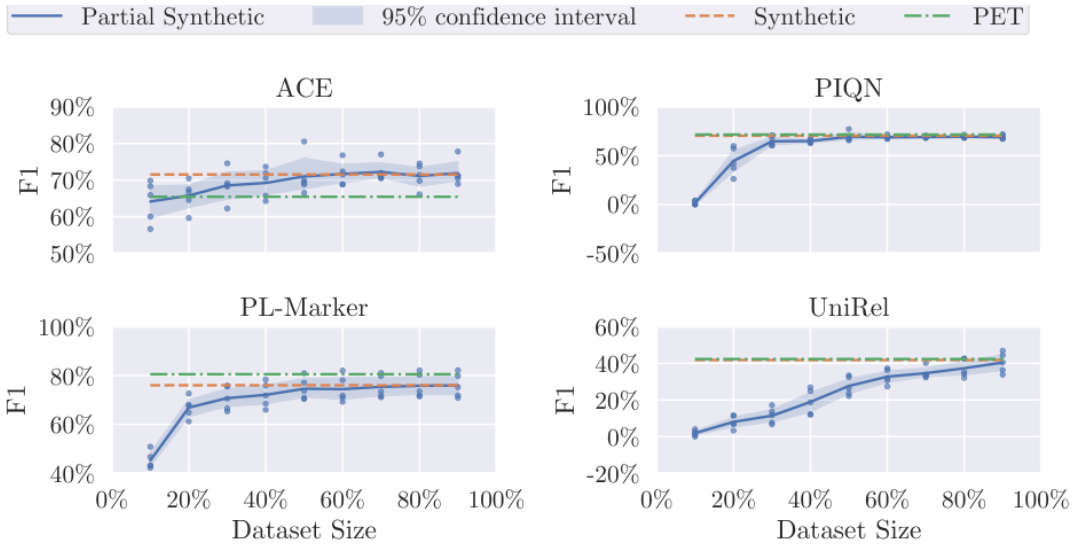
universität wien

Backup: Power Draw Over a Run



Measured GPU power draw over time for a representative extraction run.

Backup: Effect of Training-Data Size



F1 vs. amount of (partial-)synthetic training data for ACE, PIQN, PL-Marker and UniRel.