Yen-Ting Wang

Master Thesis Student

@ SAP TI AI PS FrontRunner - Eduardo & Martin

@ Mannheim University - Prof. Han van der Aa

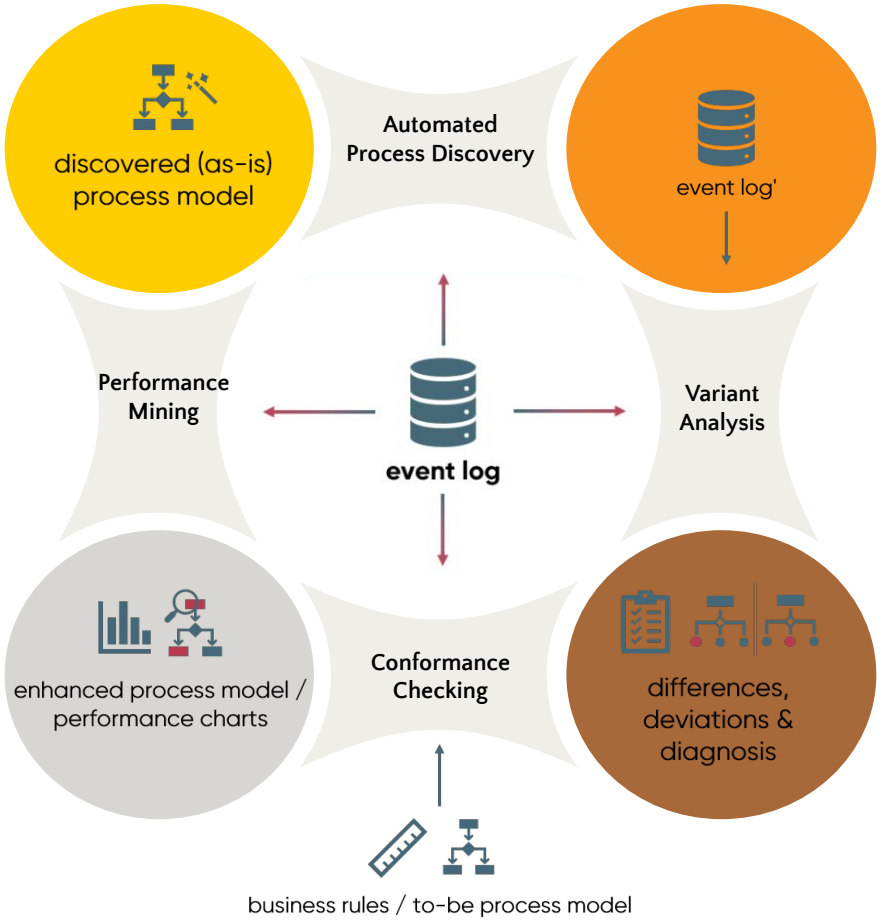
High-level Activity Label Generation

Event Log or Process Model Abstraction

Business Process Model Summarization Using Pre-trained Language Models

Process Mining
as the Bridge between
Process Science & Data Science

~Wil van der Aalst


discovered (as-is) process model
Automated Process Discovery
event log'
Performance Mining
event log
Variant Analysis
enhanced process model / performance charts
Conformance Checking
differences, deviations & diagnosis
business rules / to-be process model

Event Log or Process Model Abstraction

Event Log vs. Process Model

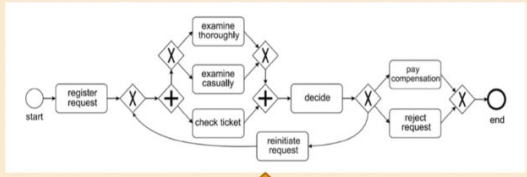
Raw Event Log

Case id	Event id	Timestamp	Event Description	Resource	Cost	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋯
12373	45632X1i	20193007T1102	reg_act_start	null	15	⋯
12373	45632X2i	20193007T1102	opsi_pp_open	112AA12	null	⋯
12373	45632X3i	20193007T1102	reg_act_end	null	35	⋯
12374	45633X1i	20193007T1132	reg_act_start	null	15	⋯
12374	45633X2i	20193007T1132	opsi_pp_open	1333A27	null	⋯
12374	45633X3i	20193007T1132	reg_act_end	null	35	⋯
12373	45634YZ1	20193007T1212	bs_op_stat_x	null	null	⋯
12373	45634YZ2	20193007T1212	ch_chk_tr	0093B74	100	⋯
12373	45634YZ3	20193007T1212	bs_op_stat_u	null	null	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋯

Expected Event Log

Case id	Timestamp	Activity	Resource	Transactional	Cost	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋯
12373	30-7-2019 11.02	Register request	Barbara	Start	50	⋯
12373	30-7-2019 11.12	Register request	Barbara	Complete	50	⋯
12374	30-7-2019 11.32	Register request	Jan	Start	50	⋯
12374	30-7-2019 11.44	Register request	Jan	Complete	50	⋯
12373	30-7-2019 12.12	Check ticket	Hajo	Start	100	⋯
12374	30-7-2019 14.16	Examine casually	Jorge	Start	400	⋯
12375	30-7-2019 14.32	Register request	Josep	Start	50	⋯
12374	30-7-2019 14.16	Examine casually	Jorge	Complete	400	⋯
12373	30-7-2019 14.42	Check ticket	Hajo	Complete	100	⋯
12375	30-7-2019 14.32	Register request	Josep	Complete	50	⋯
12375	30-7-2019 15.42	Examine thoroughly	Marlon	Start	600	⋯
12373	03-8-2019 11.18	Examine thoroughly	Barbara	Start	600	⋯
12375	03-8-2019 12.42	Examine thoroughly	Marlon	Complete	600	⋯
12373	03-8-2019 15.18	Examine thoroughly	Barbara	Complete	600	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋯

Process Model



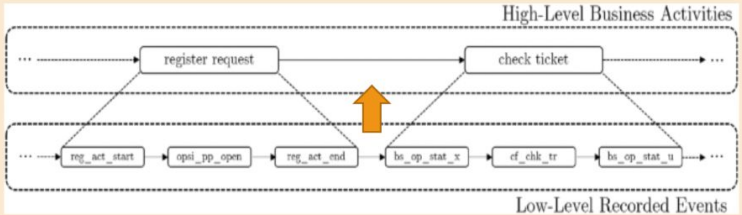
Assume 1:1 mapping

Why Abstraction > Motivation

Low-level Recorded Events

➡ **High-level** (Business) Activities

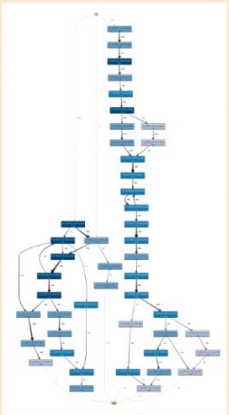
Log-lifting process



Spaghetti-like

➡ **Lasagna**-like Process Model

"Ocean" of observed process behavior



Huge behavioral variabilities

Pegasus: Pre-training with Extracted Gap-sentences for Abstractive Summarization

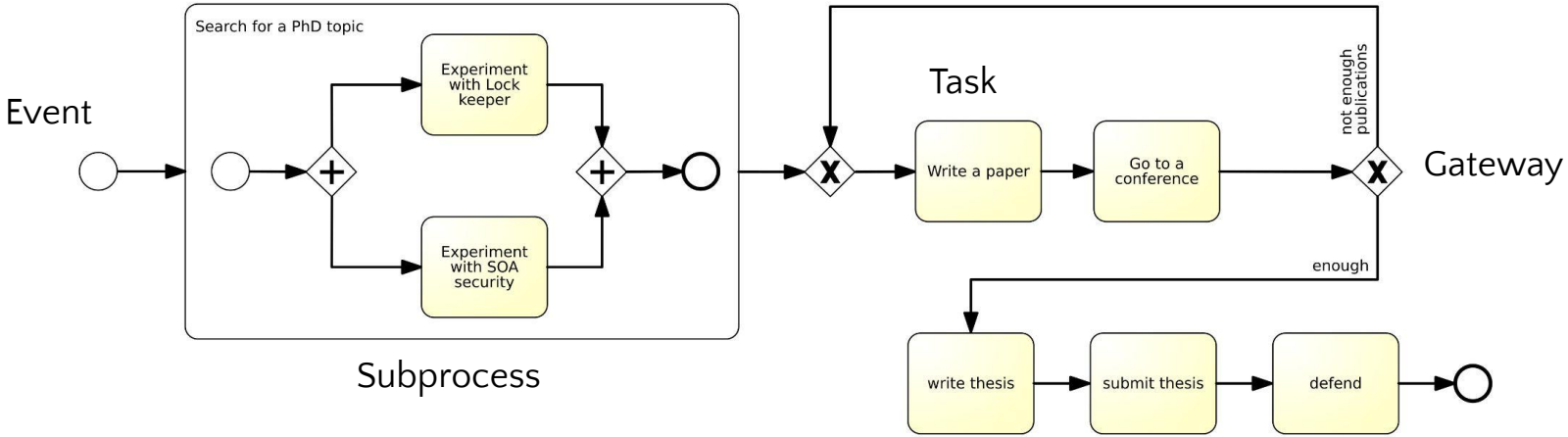
Labels that provides Abstraction

What makes a good label?

- reflects the **most important task** of the process fragment
- reflects the **main outcome** of the process fragment
- provides a **good overview** of the process fragment

Data Source

- Model Collection of the Business Process Management Academic Initiative (BPMAI)
  - BPMN 2.0 models with equal or more than 5 tasks (with descriptive labels) - 3284 examples
  - No data quality control
- Labeled data > subprocess - 100 very limited
- Event log vs. Process model

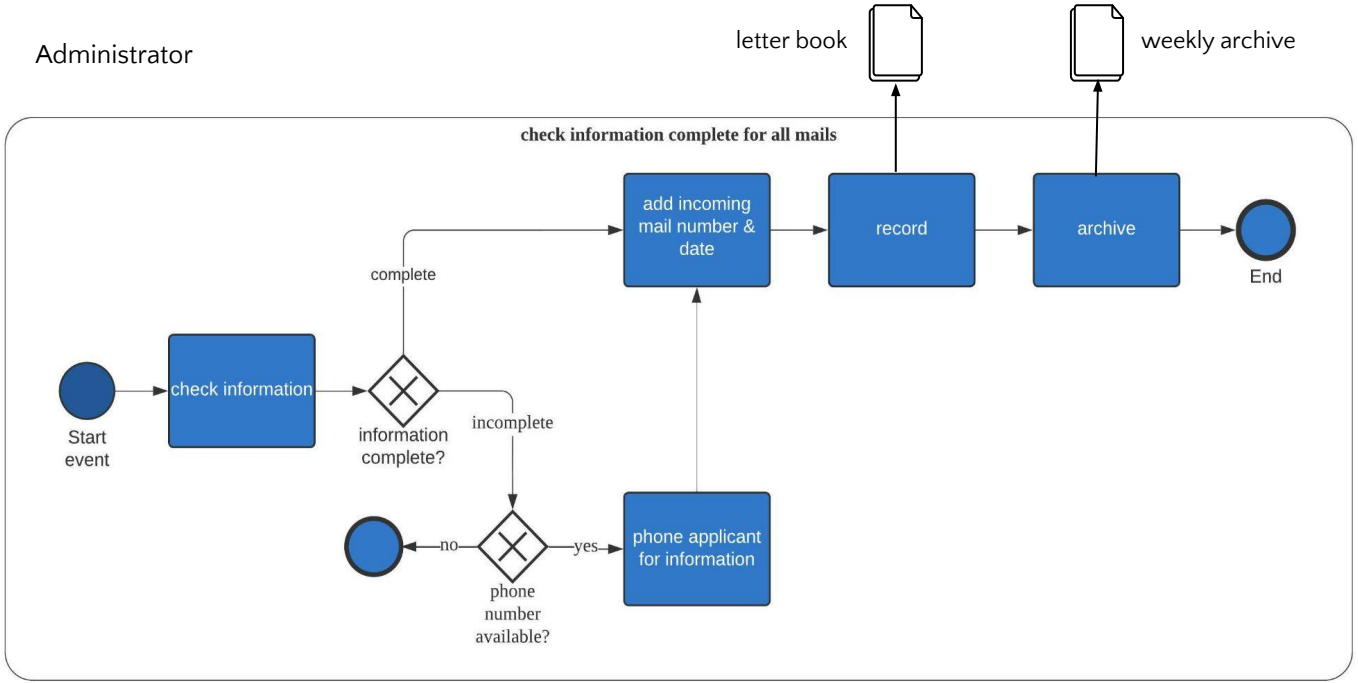


Text Summarization

Title generation

Extractive approach > unsupervised > BERT & GPT2 summarizer

Abstractive approach > self-supervised > Pegasus & T5




```
text = """
Administrator check information. Is information complete?
If complete, add incoming mail number & date, record to letter book, archive to weekly archive.
If incomplete, is phone number available? If yes, phone applicant for information.
"""
```

Extractive summarizers – BERT & GPT2

```
bert_summary = ''.join(bert_model(text, min_length=2))
print(bert_summary)
```

```
Administrator check information. If yes, phone applicant for information.
```

```
full = ''.join(GPT2_model(text, min_length=2))
print(full)
```

```
Administrator check information. If incomplete, is phone number available?
```

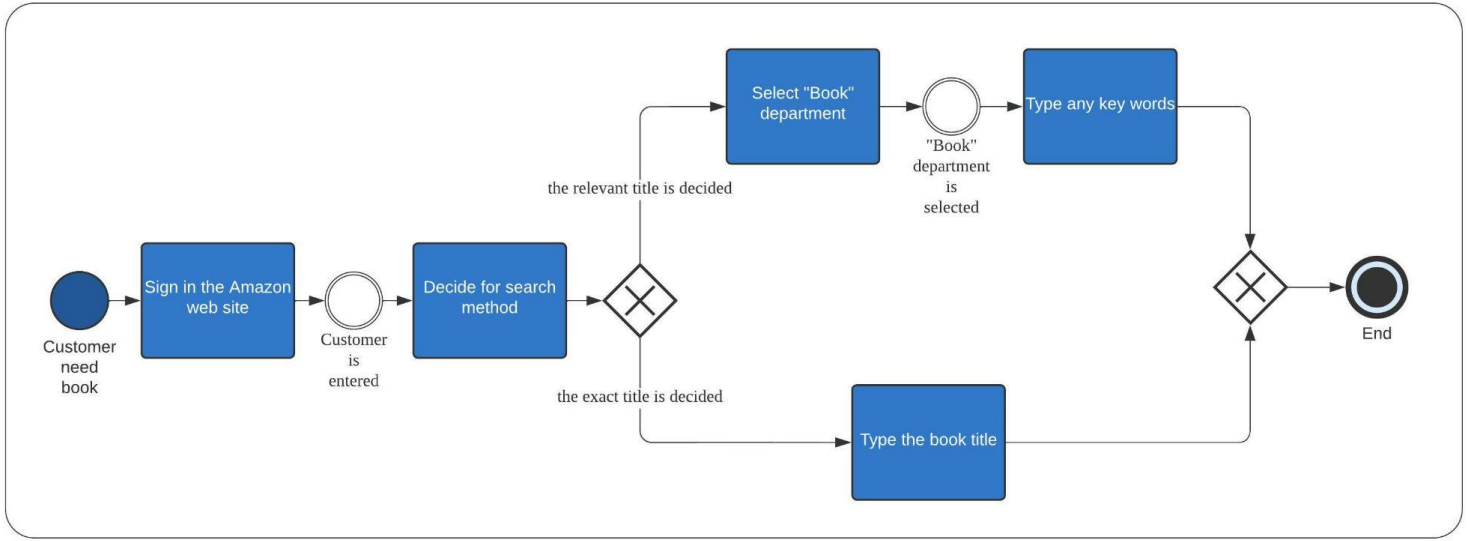
Abstractive summarizers – Pegasus & T5

```
# decode summary
tokenizer.decode(summary[0])
```

```
'All information is copyrighted.'
```

```
Summarized text:
 administrator check information. is information complete?
```

Extractive method – GPT2 seems to perform the best




```
text2 = """
Customer need book. Sign in the Amazon web site. Customer is entered. Decide for search method.
Either the relevant titles is decided, select "Book"department, "Book" department is selected, type any key words,
or the exact title is decided, Type the book title.
"""
```

Extractive summarizers – BERT & GPT2

```
bert_summary = ''.join(bert_model(text2, min_length=2))
print(bert_summary)
```

```
Customer need book.
```

```
full = ''.join(GPT2_model(text2, min_length=2))
print(full)
```

```
Customer need book. Decide for search method.
```

Abstractive summarizers – Pegasus & T5

```
# decode summary
tokenizer.decode(summary[0])
```

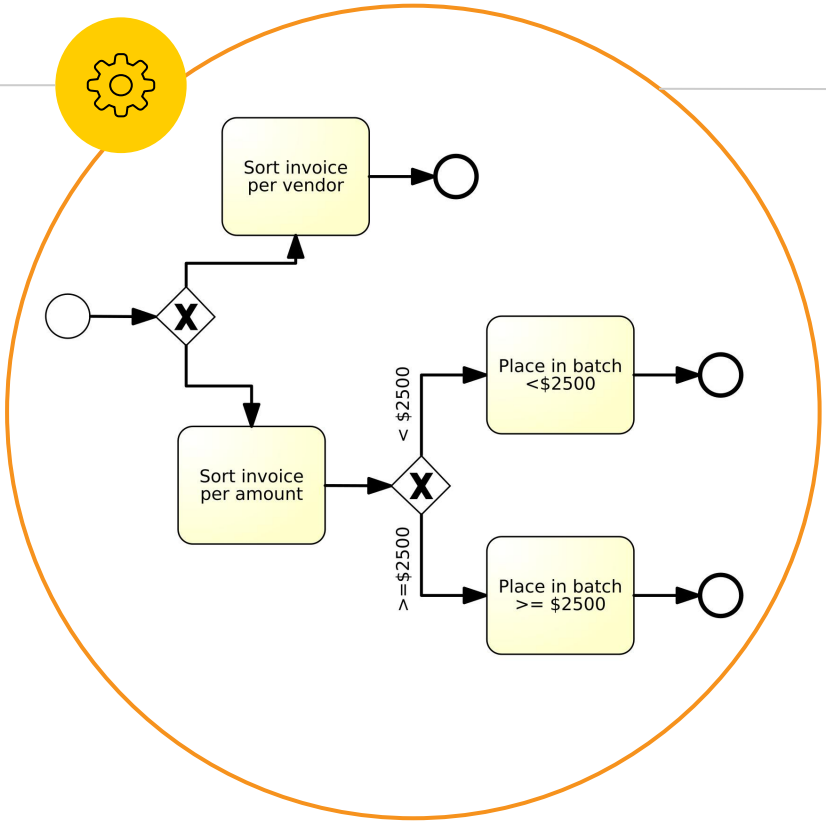
```
'How to find a book on Amazon?'
```

```
Summarized text:
 customer need book. Sign in the amazon web
```

Extractive method – GPT2 seems to perform the best

What are we looking for?

Extractive approach seems to perform better when directly using **out-of-the-box** method. However, a **concise domain-targeted labeling** is what we are looking for.



Extracted labels

sort invoice per vendor, sort invoice per amount, place in batch <$2500, place in batch >= $2500

Different from the usual syntactically structured text...

- Target both event log & process model abstraction

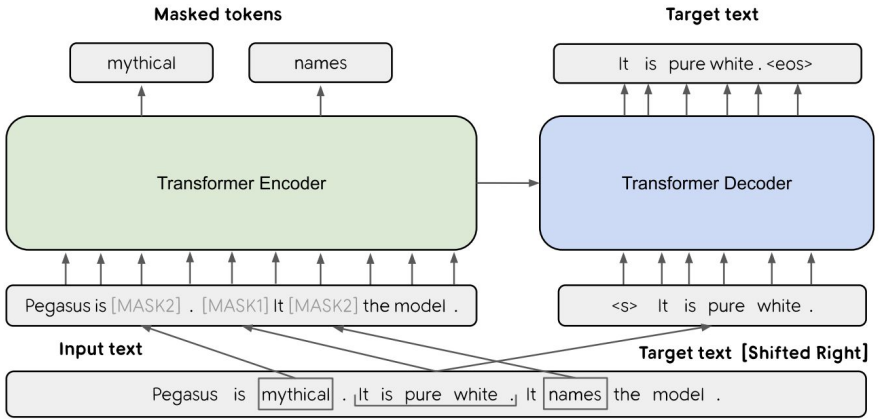
Labeling Optimization

In Process Domain

Fine-tuning language model - abstractive summarizer - to achieve representative domain-specific abstraction

Pegasus Extracted Gap-Sentences Training

- **Seq2Seq** summarisation model
- **Gap-sentences generation** (GSG) self-supervised learning
  - Important sentences are masked from input document
  - Similar to an extractive summarization
- **Decoder** acts as an **autoregressive model**
  - Reproduce important sentences at output
- **GSG** works well as a pre-training objective for **downstream summarization tasks**
- Model performs well on **low-resource** summarization (-1000 examples)



Pegasus: Pre-training with Extracted Gap-sentences for Abstractive Summarization

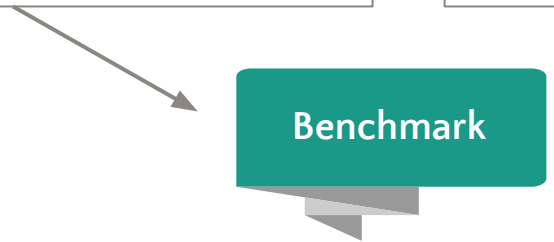
Pre-train & Fine-tune Pegasus

Step 1 > further pre-training

- Gap-sentences pre-training on **process text** with much less data
- Important task or event label masking
  - Tasks in front of gateways
  - Start event or first task
  - End event or end task
- Self-supervised learning - **1000** data samples

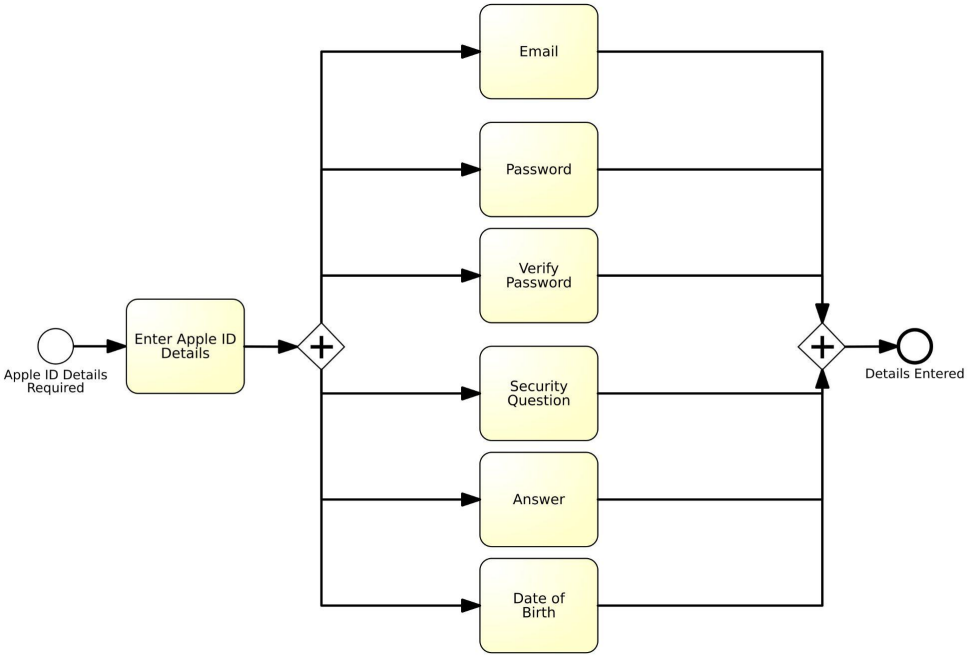
Step 2 > fine-tuning on step-1 model

- Fine-tuning for summarization downstream task on **subprocesses** with even less data
- During training, model is given a group of low-level tasks and events as source and human label as target
- Model output **abstractive summary**
- Supervised learning - **100** data samples

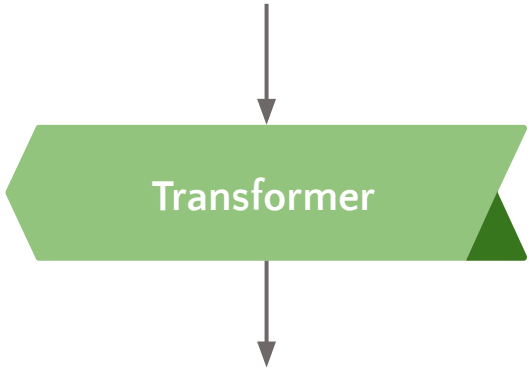


Benchmark Results

Step-1 > Mask task In front of gateway



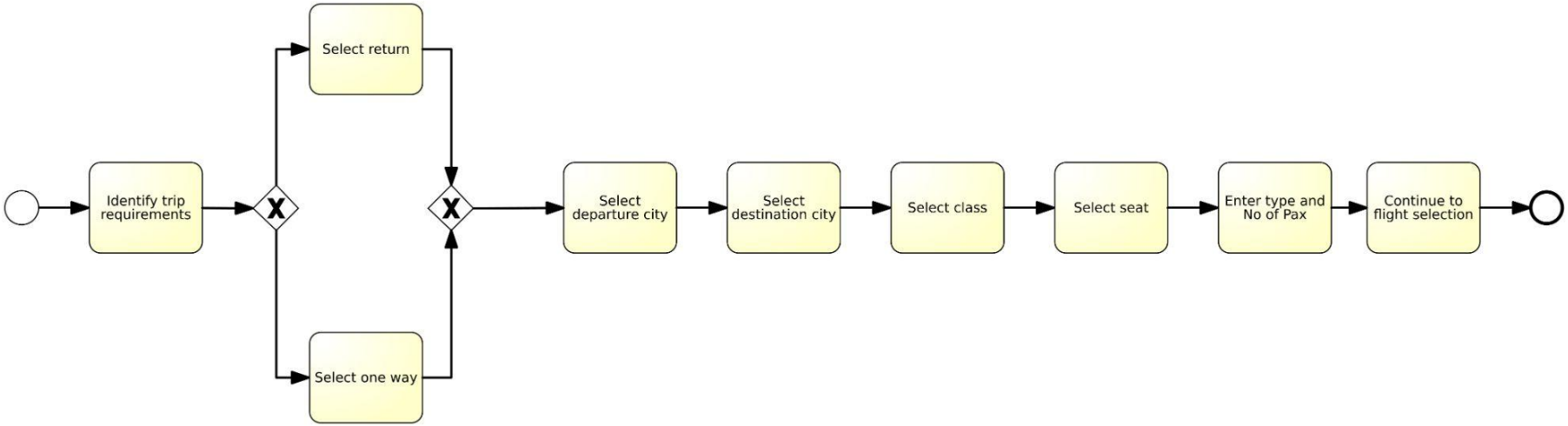
Apple ID Details Required, [mask_1], Date of Birth, Email, Password, Verify Password, Security Question, Answer, Details Entered



Model result **Enter Required Information**

Label **Enter Apple ID Details**

Step-1 > Mask first task (also in front of gateway)

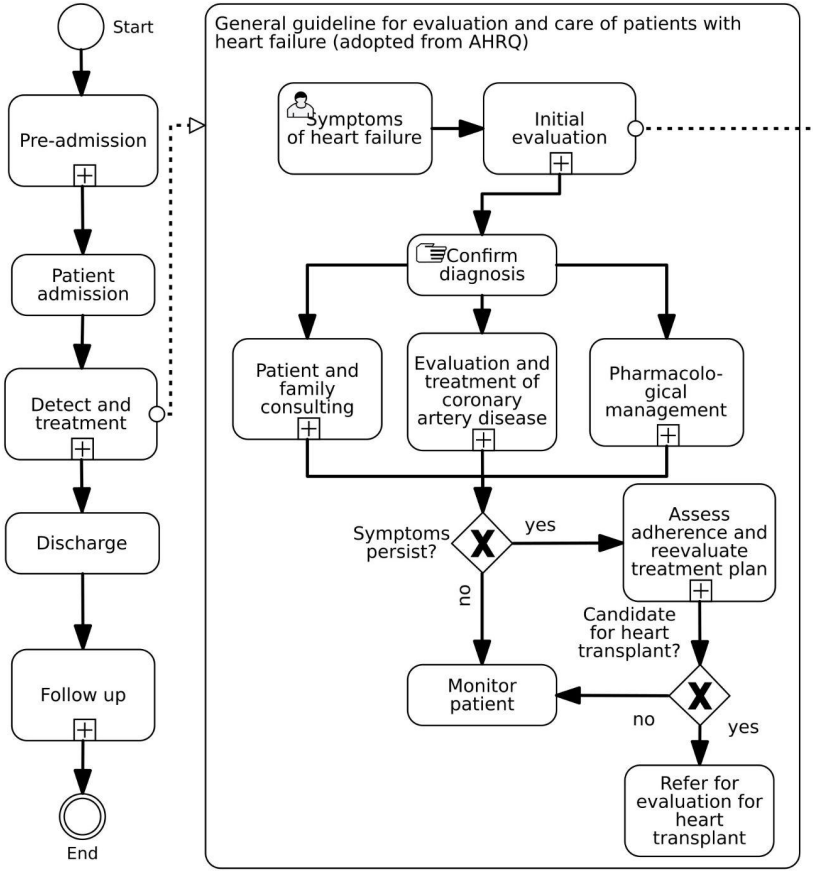


[mask_1], Select one way , Select return,
Select departure city, Select destination city,
Select class, Select seat, Enter type and No of Pax,
Continue to flight selection

Transformer

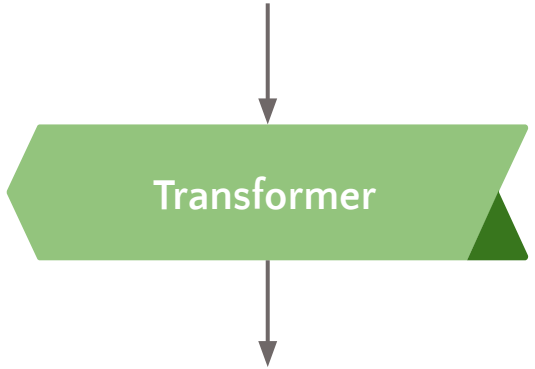
Model result Select flight

Label Identify trip requirements



Step-2 > Provide good overview of the process

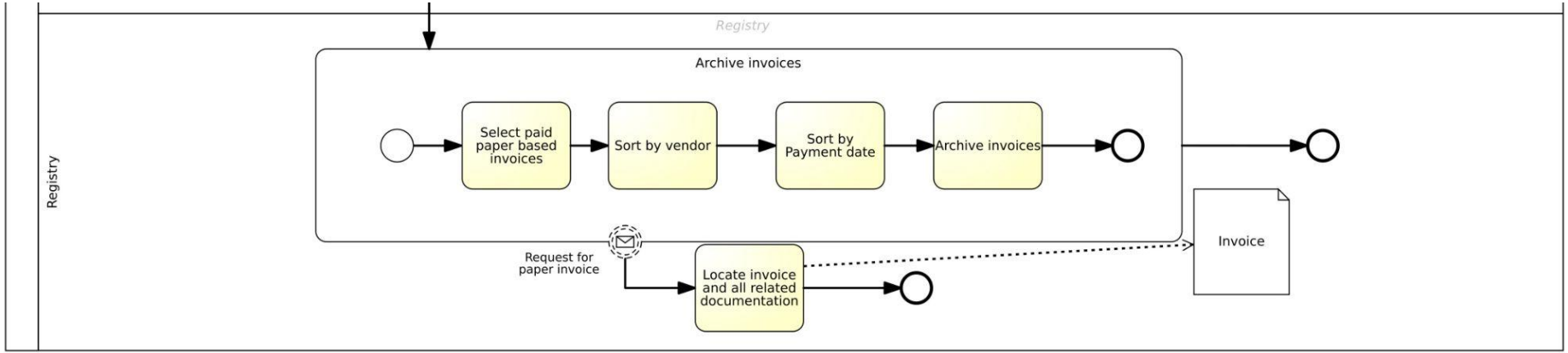
Symptoms of heart failure, Confirm diagnosis, Monitor patient, Refer for evaluation for heart transplant



Model result Assess patient

Label General guideline for evaluation and care of patients with heart failure (adopted from AHRQ)

Step-2 > Main outcome of the process



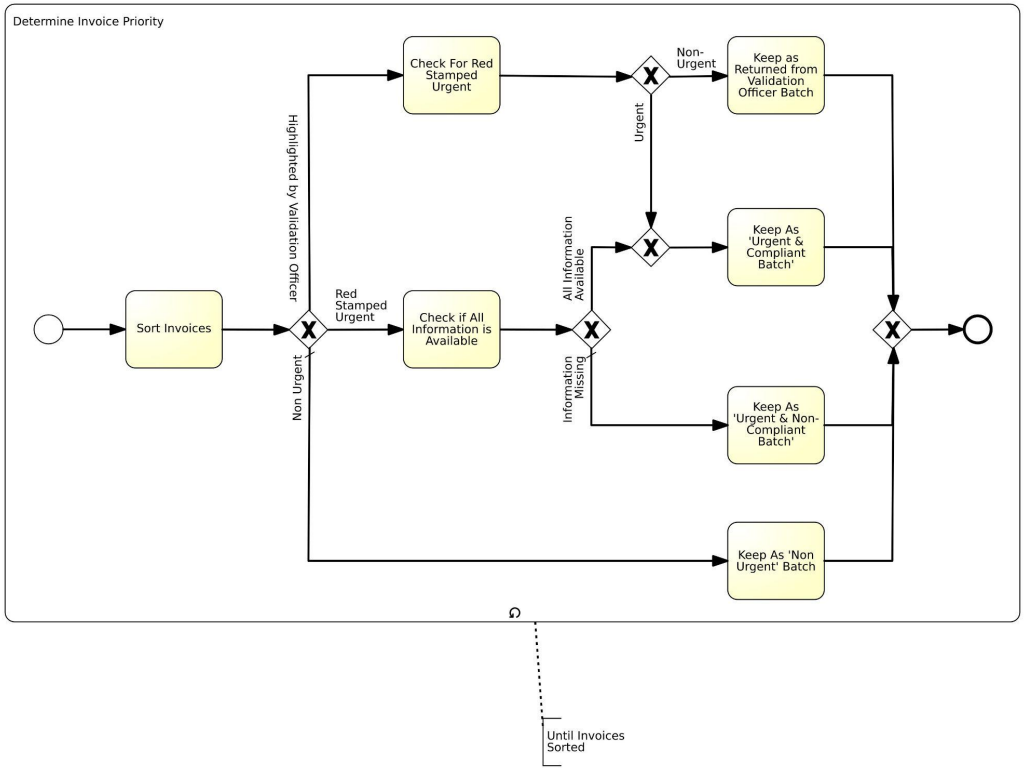
Select paid paper based invoices, Sort by vendor, Sort by Payment date, Archive invoices

Transformer

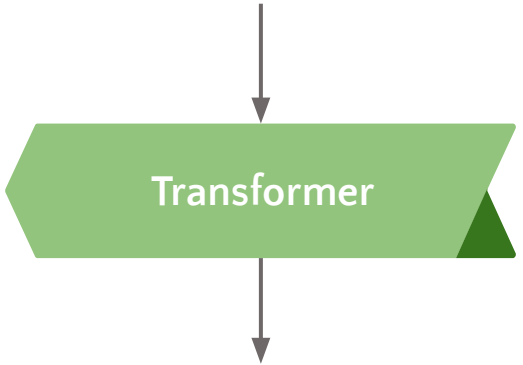
Model result **Invoices validation**

Label **Archive invoices**

Step-2 > Most important task in the process



Sort Invoices, Check if All Information is Available, Keep As 'Urgent & Compliant Batch', Keep As 'Urgent & Non-Compliant Batch', Keep As 'Non Urgent' Batch, Check For Red Stamped Urgent, Keep as Returned from Validation Officer Batch



Model result **Batch Validation**

Label **Determine Invoice Priority**

Problems we've seen now in the benchmark approach

Structural information missing

- **Directly-follows** relation vs. **Parallel** relation
- Next sentence prediction (BERT)
  - Entailment relation
  - Downstream task such as,
    Natural Language Inference or Question Pairs
- Incorporate (graph) structural information
  - Do it without having special tokens [CLS] & [SEP]

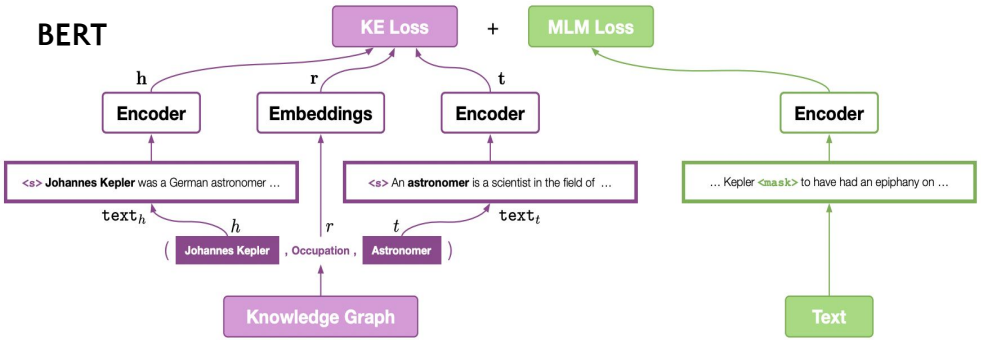
Limited data & poor data quality

- **Manual labeling** is needed
  - Labeling strategy
- **Data augmentation**
  - Suitable for tiny dataset

Advanced Approaches

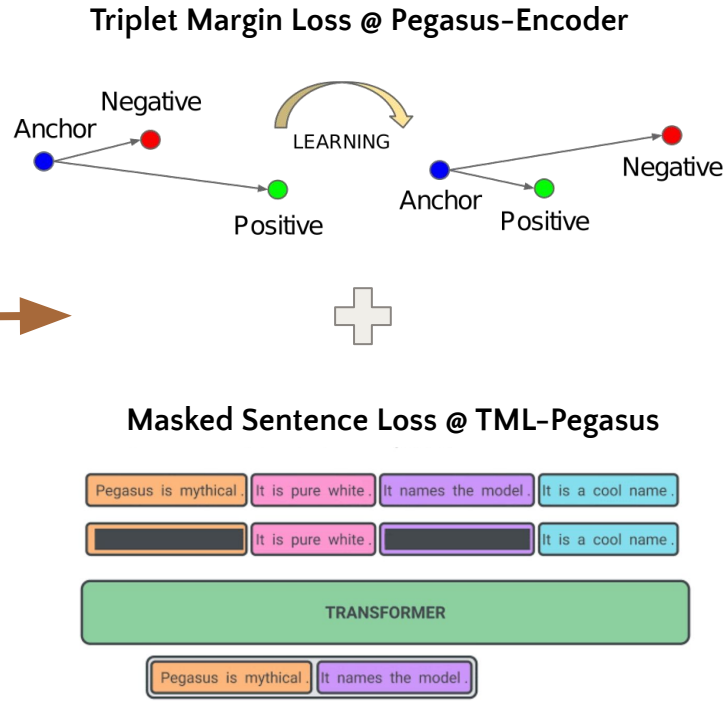
Pre-training

KEPLER

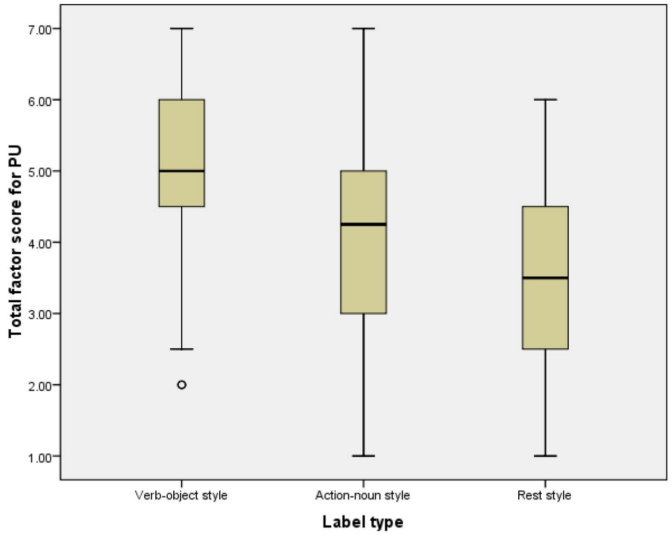


KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation

TML-Pegasus



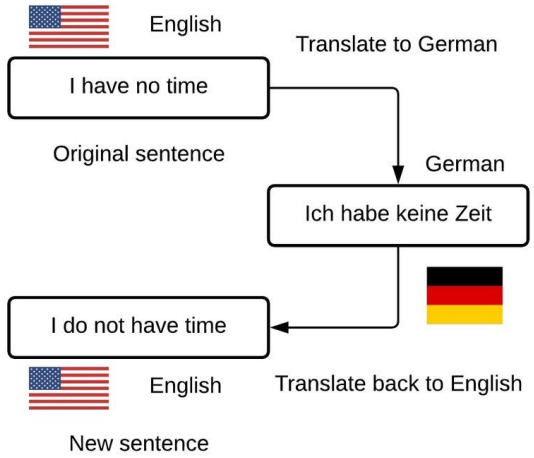
Labeling Strategy



- Verb-object **Fulfill order**
- Action-noun **Order fulfillment**

Figure > Activity labeling in process modeling: Empirical insights and recommendations

Data Augmentation via Round-trip Translation



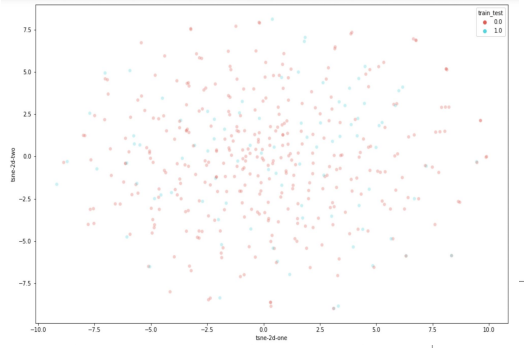
- **Top 10** beam hypothesis
- With and without **Summary Augmentation**

Figure > Data Expansion using Back Translation and Paraphrasing for Hate Speech Detection

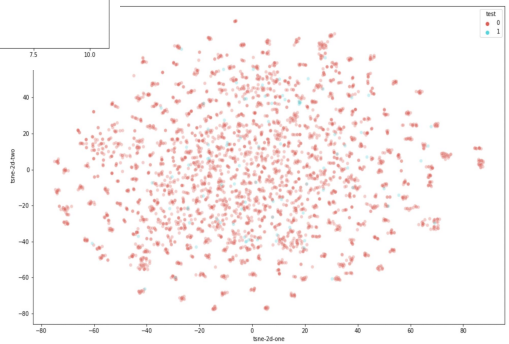
Process Content Diversity

via Doc2Vec – TSNE for visualization

No Data Augmentation



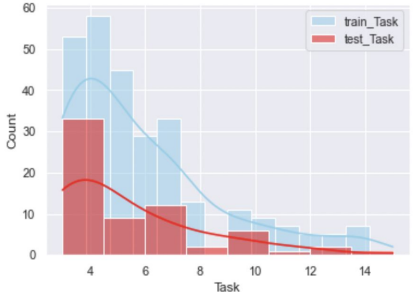
Data Augmentation

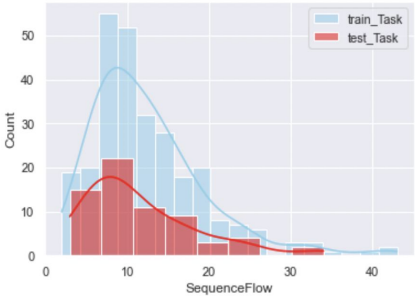


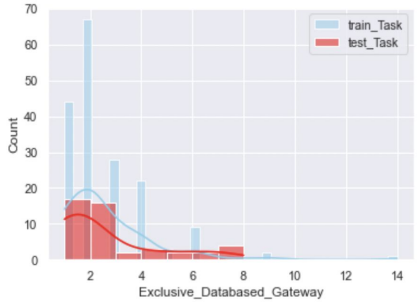
Process Model Complexity

via number of Tasks,

Sequence flows, Gateways

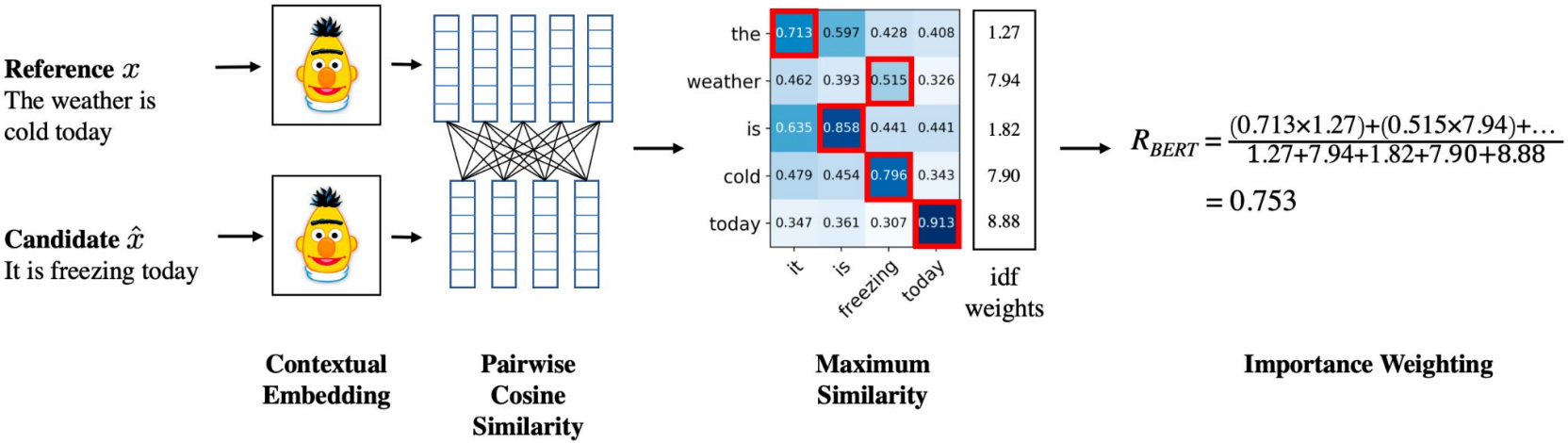




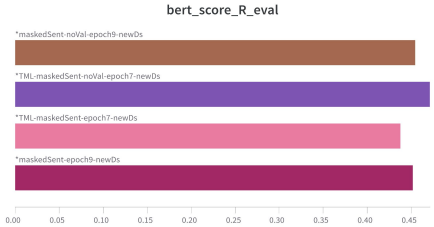


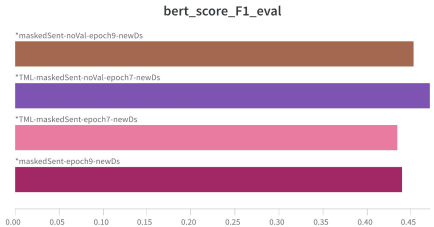
Evaluation Metrics

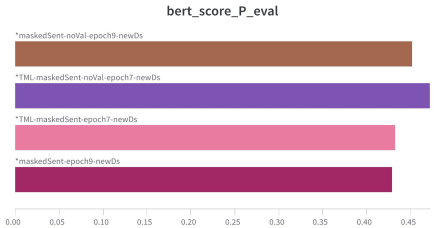
BertScore BERTScore Github



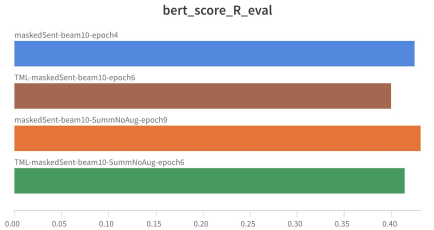
No Data Augmentation

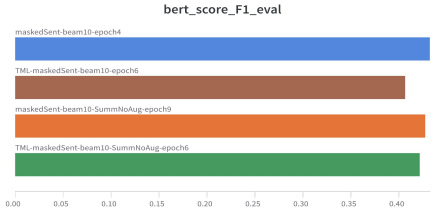


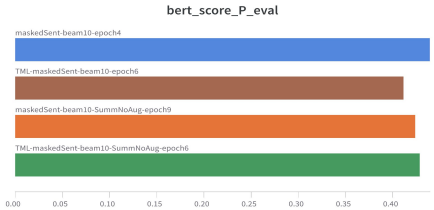




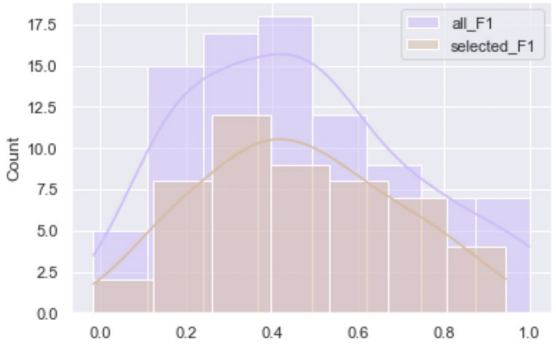
Data Augmentation







Difficult to evaluate pure quantitatively
That's why we need human evaluation



Human evaluation samples
› **50 out of 90**
selected based on the averaged BertScore F1 distribution of all test examples

Human Evaluation & Case Analysis

Labeling task is subjective

Conduct User Study to evaluate the usefulness of the result

When looking with human ...

- **Local perspective** > TML-Pegasus with new labeled data

  Better at catching **important task** or **main outcome** or **the combination**

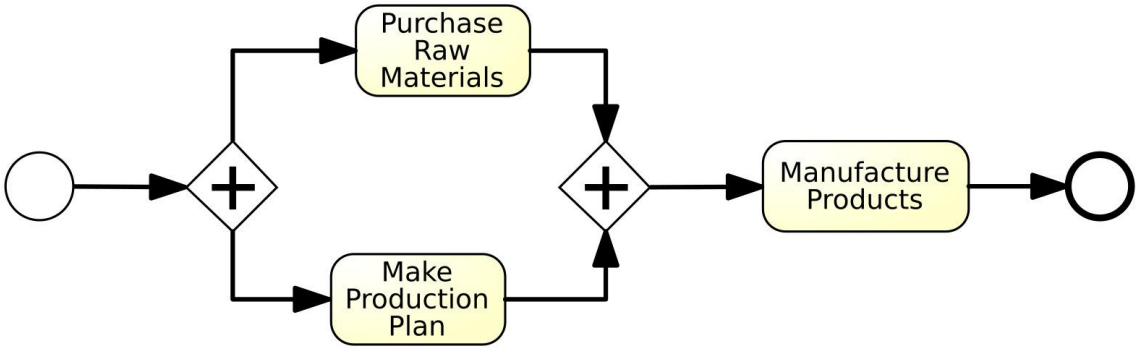
- **Global perspective** > Benchmark Pegasus with new labeled data

  Better at giving a **good overview**

- **Generalization** > Both models + Data Augmentation with new labeled data

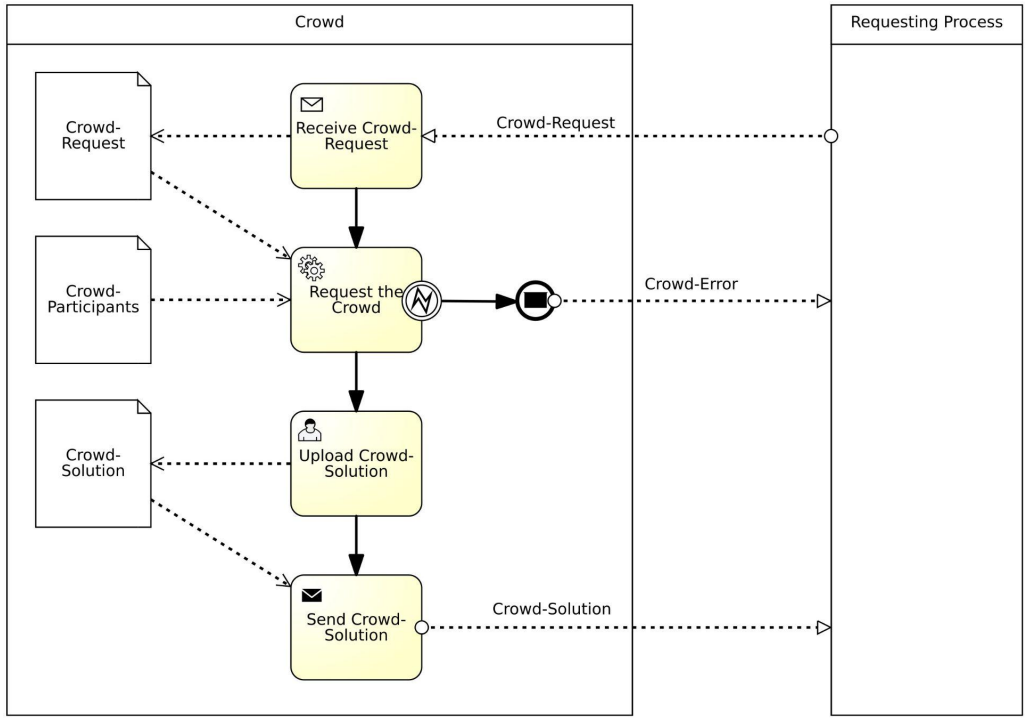
  Learned to also offer a more **general point of view**

Case 1



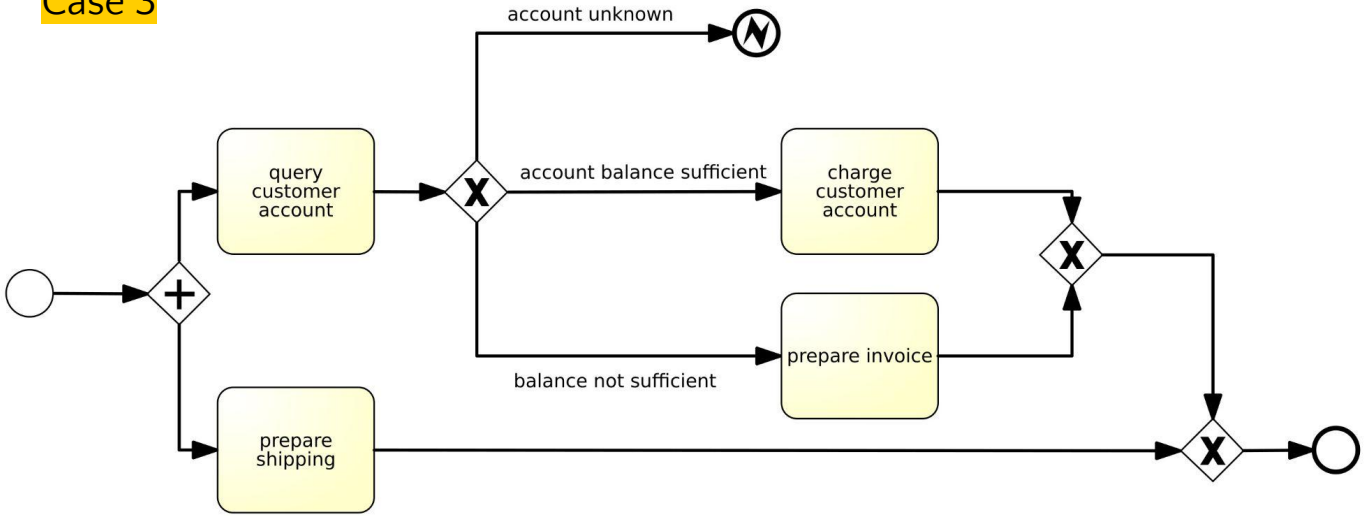
- manufacture products
- plan production
- manufacture of customized products
- manage order production

Case 2



- request crowd-based solutions
- upload crowd-request
- handle crowd request
- operate problem solving process

Case 3



- prepare shipping invoice
- prepare customer order
- manage customer account configuration
- process shipping order request

Interested in finding out more?

> *Help with Evaluation!*

The main takeaways

- Labeling automatically for processes is hard
- Utilizing pre-trained language model achieves good results with very limited data
- What we achieve now is being able to offer valid suggestions to users
  - View different perspectives of suggestions
  - Select or combine the suggestions
- Perhaps offer ranking for the future

Q & A

I am curious about your questions or discussion

You can find me at

- yen-ting.wang@sap.com

Before we go, let me introduce the evaluation survey briefly