Intelligent Document Processing

Final Presentation

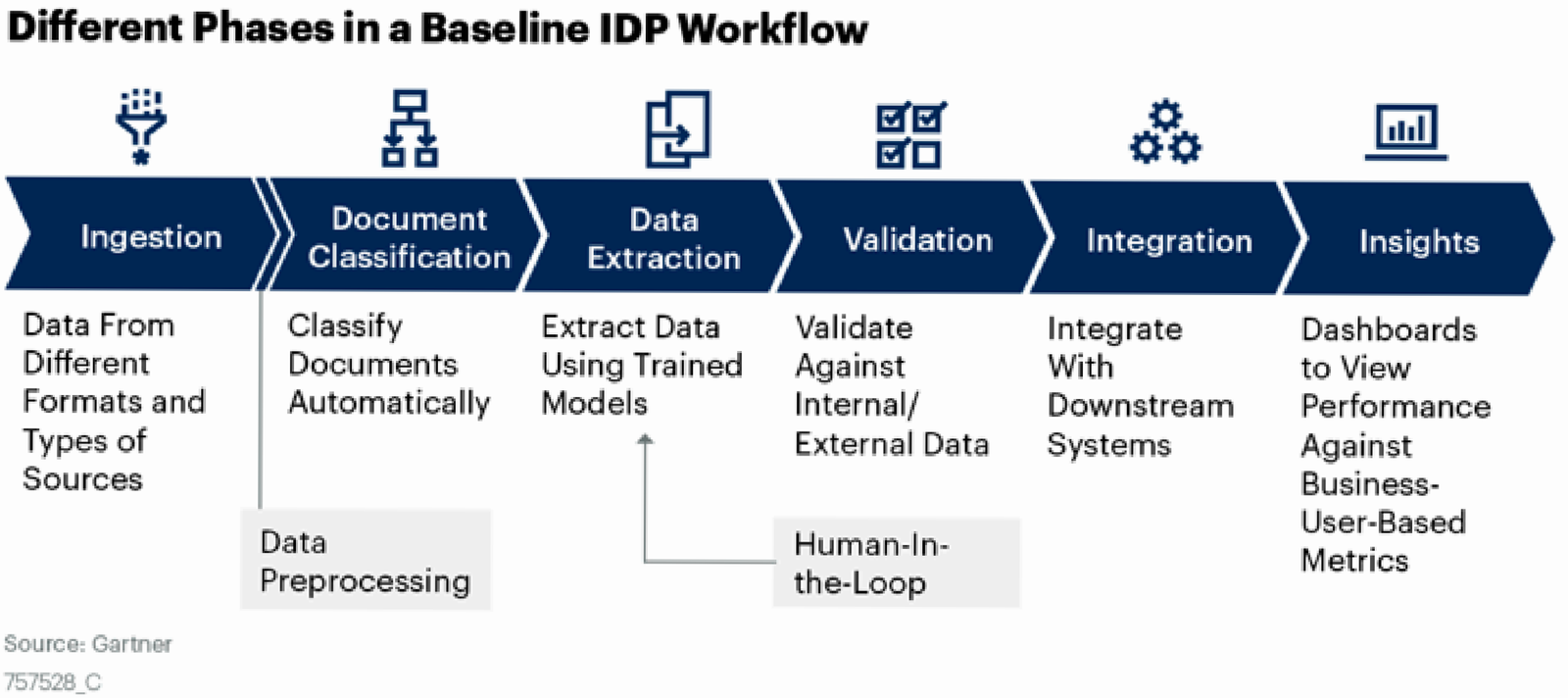
Presented by:

Ishleen Kaur	Krishnendu Biswas	Ali Haider Khan	Aditya Thite
Intern	Intern	Intern	Intern

Under the guidance of:

Dr. Chandan Biswas	Mr. Manpreet Singh	Mr. Akshay Kumar	Mr. Gopi Maguluri
Co-ordinator	Mentor	Mentor	Mentor

Introduction: IDP



Gartner

Project Overview

- Named Entity Recognition (NER): Leveraging fine-tuned transformer models such as LUKE and DistilBERT, NER automates the identification of key entities, enhancing accuracy and reducing manual effort in document processing.

- Text Summarization: Utilizing Pegasus and BART-CNN, text summarization condenses extensive texts into concise summaries, saving time and mitigating information overload, thereby streamlining the handling of large volumes of text.

- Table Structure Recognition (TSR): Employing YOLOv8 and PaddleOCR, TSR accurately extracts and preserves relationships in tabular data, improving efficiency, scalability, and integration with enterprise systems, making IDP a powerful solution for modern data management.

Introduction: NER

Named Entity Recognition (NER) is a technique used to identify and categorize named entities within unstructured text into predefined categories such as persons, organizations, locations, dates and quantities.

How NER Identifies and Categorizes Entities:

- Machine Learning:NER systems utilize supervised machine learning models
- Rule-Based Approaches: NER systems use predefined rules and patterns

Problem Statement

The problem statement of Named Entity Recognition (NER) in Intelligent Document Processing (IDP) typically revolves around the challenges and goals associated with accurately extracting and categorizing specific information from unstructured documents.

Unstructured Data

Ambiguity and Context

Accuracy and Reliability

Model

```
Star B-MISC
Wars I-MISC
is O
a O
film O
written O
and O
directed O
by O
George B-PER
Lucas I-PER
```

LUKE (Language Understanding with Knowledge-based Embeddings)

LUKE is a large-scale pre-trained language model that extends BERT (Bidirectional Encoder Representations from Transformers) with knowledge embeddings.

Key Features

- Entity-aware Embeddings
- Large-scale Knowledge Integration

How LUKE Works

- Tokenization and Input Encoding
- Entity Span Classification

Using LUKE for Named Entity Recognition

- **Model Selection and Loading**: Utilize the LUKE model pre-trained on entity span classification tasks, such as LukeForEntitySpanClassification.
- **Data Preparation and Preprocessing:** Load and preprocess documents or datasets, ensuring they are formatted for input to the LUKE model.
- **Model Inference and Evaluation**: Perform batched inference on prepared examples using LUKE.
- **Integration with IDP Systems:** Integration of LUKE-based NER into IDP workflows for automated document processing.

Model


```
'input': [['Jacob',
  'Collier',
  'is',
  'a',
  'Grammy',
  'awarded',
  'English',
  'artist',
  'from',
  'London']],
'entity_prediction': [[{'type': 'person',
   'entity': ['Jacob', 'Collier'],
   'position': [0, 1],
   'probability': [0.9962937235832214, 0.998
  {'type': 'location',
   'entity': ['London'],
   'position': [9],
   'probability': [0.987984836101532]}]]}
```


distilBERT-base-cased

distilBERT-base-cased is a smaller, faster, and cheaper version of BERT, designed by distilling the knowledge from a larger BERT model.

Key Features

- Lightweight
- Fast
- Versatile

Fine-tuning distilbert-base-cased for Named Entity Recognition with T-NER

- **Define Searcher for Hyperparameter Optimization:**
- Create a GridSearcher instance, which will manage the hyperparameter optimization process.
- Set the checkpoint directory where the model states will be saved during training

- **Train the Model:**
- Initiate the training process by calling searcher.train(), which starts fine-tuning the model.
- The GridSearcher will try different hyperparameter configurations to find the best-performing model setup

- **Evaluate the Model:**
- Load the best model from the checkpoint directory using TransformersNER.
- Verify the model's predictions on a sample text to check the NER performance.

Text Summarization

Text summarization is the process of condensing a long document into a shorter version, capturing its main ideas and key information.

Steps included:

- Input Text
- Tokenization
- Pre-training
- Fine-tuning
- Summary Generation
- Output

Selected Models and Their Rationale

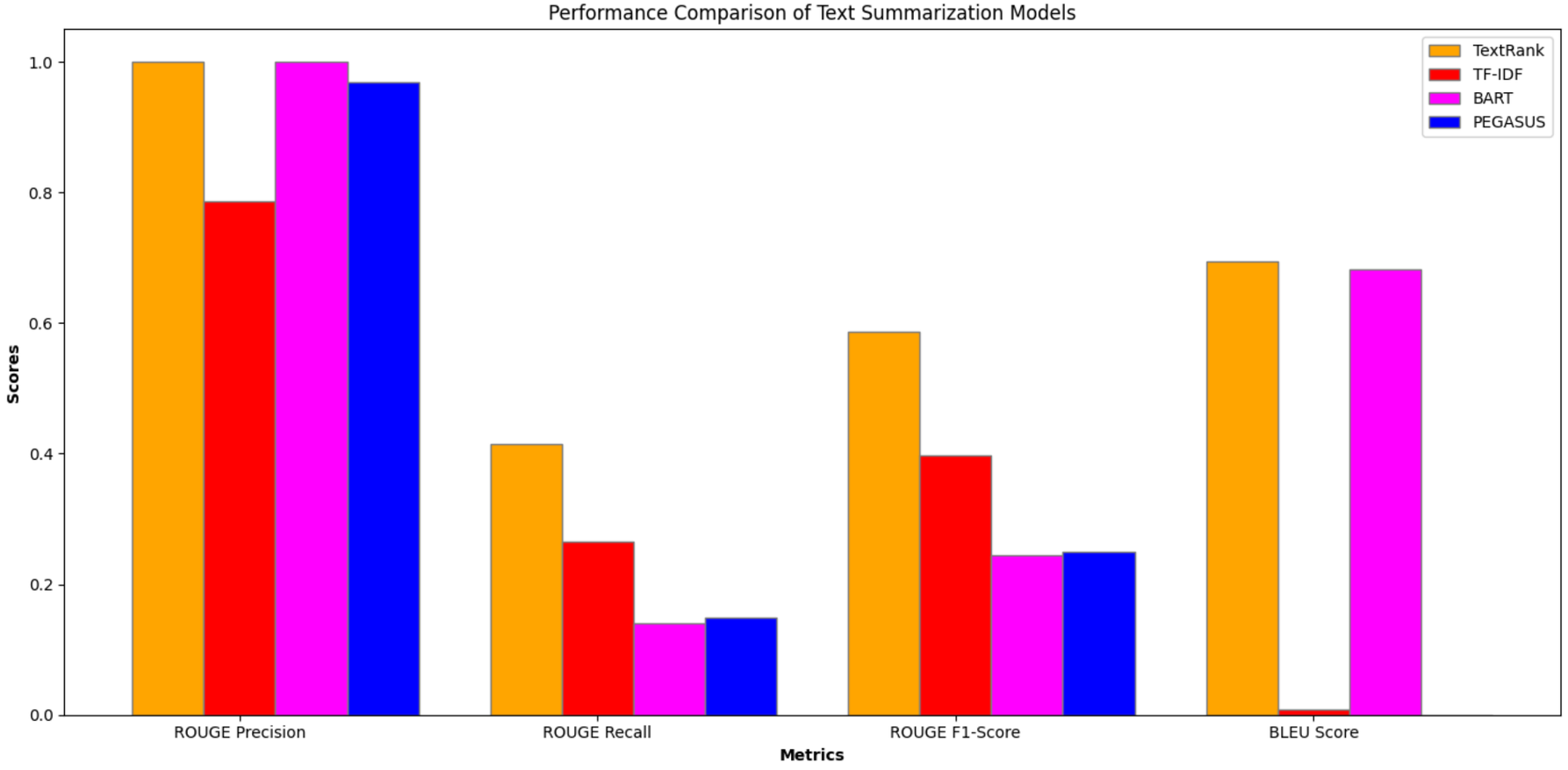
- **TextRank**
  - An unsupervised graph-based ranking algorithm.
  - Reason for Selection: Known for its efficiency and effectiveness in extractive summarization.
- **TF-IDF (Term Frequency-Inverse Document Frequency)**
  - A statistical method used to evaluate the importance of a word in a document relative to a collection of documents.
  - Reason for Selection: Its simplicity and effectiveness as a baseline model for text summarization.
- **BART (Bidirectional and Auto-Regressive Transformers)**
  - A denoising autoencoder for pretraining sequence-to-sequence models.
  - Reason for Selection: Its robust performance in both extractive and abstractive summarization tasks.
- **PEGASUS**
  - Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence
  - Reason for Selection: Its flexibility and strong performance across various NLP tasks, including summarization.

Model Comparison

METRICS	TEXTRANK	TF-IDF	BART	PEGASUS
ROUGE Precision	1.000	0.787	1.00	0.969
ROUGE Recall	0.414	0.266	0.140	0.149
Rouge F1	0.586	0.398	0.245	0.250

These models were selected to provide a comprehensive evaluation of both traditiona advanced techniques in text summarization.

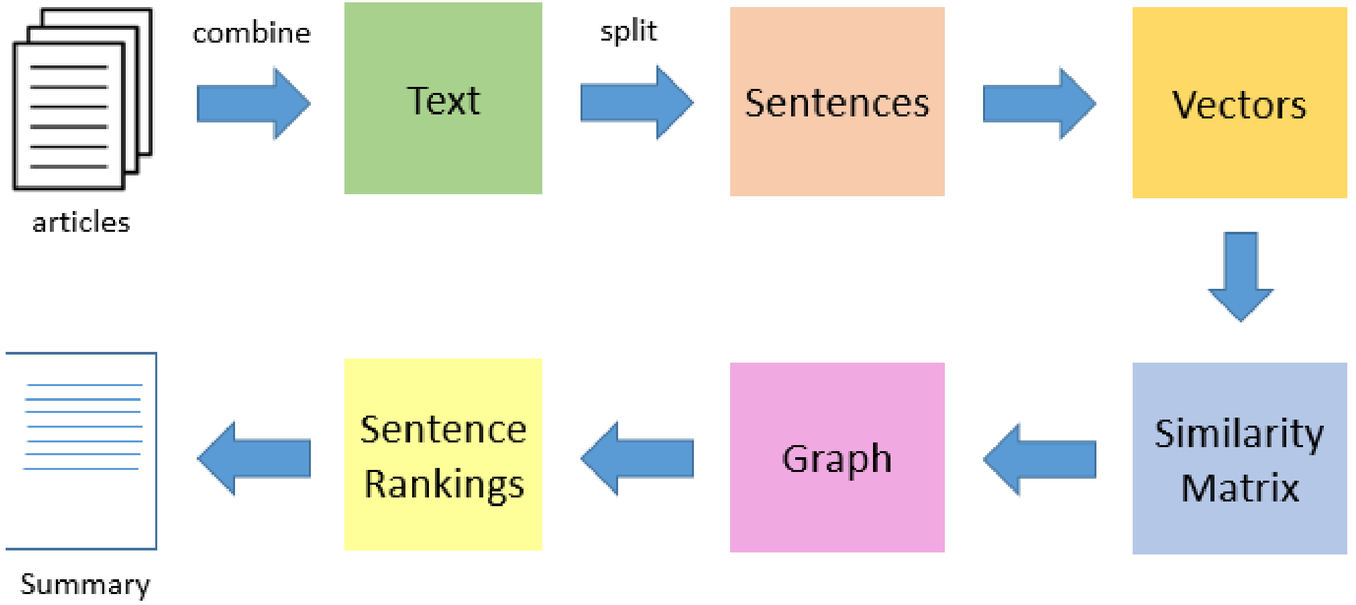
- **TextRank** emerged as the best-performing model, excelling in ROUGE Precision, ROUGE F1-Score, and BLEU Score.
- **PEGASUS** showed competitive performance with high ROUGE Precision.
- **BART** demonstrated high ROUGE Precision but lower performance in other metrics.
- **TF-IDF** served as a baseline and showed lower performance across all metrics.



What Do You Need To Know is Why These Models Were Selected?

TextRank: Graph-Based Text Summarization

- The first step would be to concatenate all the text contained in the articles
- Then split the text into individual sentences
- In the next step, we will find vector representation(word embeddings) for each and every sentence



- The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation

- Finally, a certain number of top-ranked sentences form the final summary

PEGASUS

It is a state-of-the-art text summarization model developed by Google Research. It is designed to generate high-quality summaries of long documents by leveraging a novel pre-training strategy.

```
model_name = "google/pegasus-large"
tokenizer = PegasusTokenizer.from_pretrained(model_name)
model = PegasusForConditionalGeneration.from_pretrained(model_name)
```

```
inputs = tokenizer([text], max_length=text_len, return_tensors='pt', truncation=True)
summary_ids = model.generate(inputs.input_ids, num_beams=6, length_penalty=0.8,
                             max_length = int(text_len * 0.15), min_length = int(text_len * 0.1),
                             early_stopping=True)
summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
```

BART

It is a sequence-to-sequence model introduced by Facebook AI Research in 2019. It's designed to excel in text generation and comprehension tasks such as text summarization, machine translation, and question answering. BART combines the benefits of bidirectional context and autoregressive generation, making it a versatile and powerful model for various NLP applications.

```
model_name = "facebook/bart-large-cnn"
tokenizer = BartTokenizerFast.from_pretrained(model_name)
model = BartForConditionalGeneration.from_pretrained(model_name)
```

```
inputs = tokenizer([text], max_length = text_len, return_tensors='pt', truncation=True)
summary_ids = model.generate(inputs.input_ids, num_beams=6, length_penalty=0.6,
                             max_length = int(text_len * 0.15), min_length = int(text_len * 0.1),
                             early_stopping=True)
summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
```

Future Scope for Text Summarization

- **Enhanced Summarization Accuracy:**

1. TextRank: Improve coherence by refining graph construction.
2. TF-IDF: Integrate with semantic models for better summaries.

- **Hybrid Models:**

1. Combine extractive (TextRank, TF-IDF) and abstractive (BART, PEGASUS) methods for comprehensive summaries.

- **Domain-Specific Summarization:**

1. BART: Fine-tune for specific industries like legal, medical, and financial.
2. PEGASUS: Adapt pre-training for domain-specific datasets.

Table Structure Recognition

Introduction :

Table Structure Recognition (TSR) refers to the process of automatically identifying and extracting the structural layout of tables within documents or images. This includes recognizing the rows, columns, headers, and cells that define the organization of tabular data.

Importance

- **Efficiency**: Automates data extraction from documents, saving time and effort.
- **Accuracy:** Reduces errors compared to manual data entry.
- **Data Analysis**: Facilitates quick analysis and visualization of structured data.
- **Document Digitization:** Converts physical documents into searchable digital formats

Problem Statement

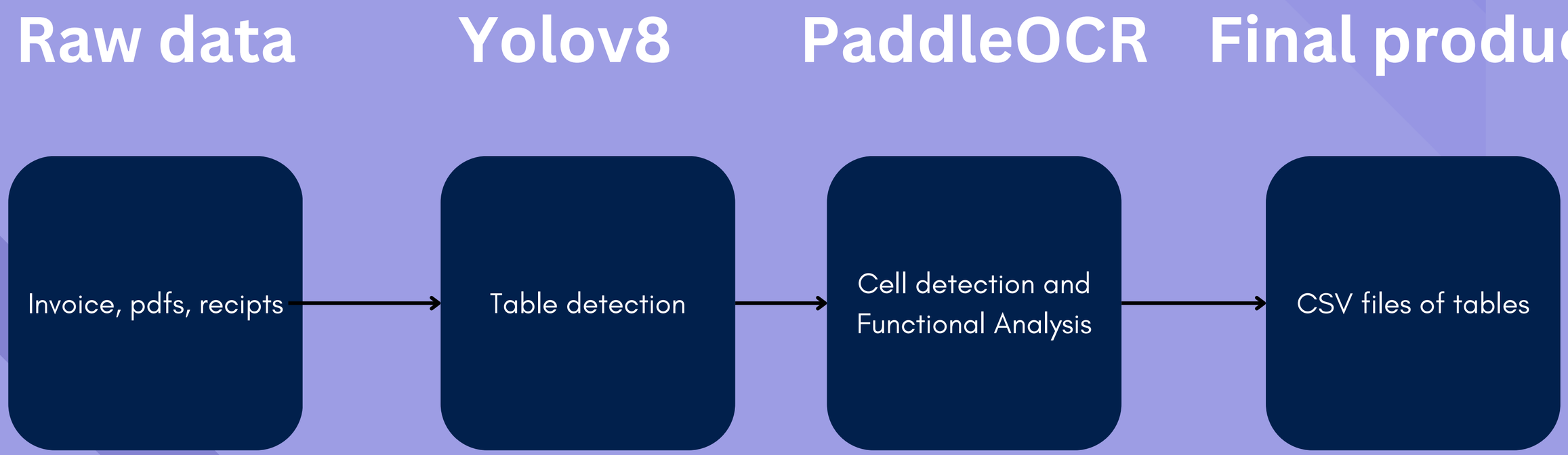
The problem statements are proper table detection and table schema generation in CSV or XLSX format for creating properly structured data.

Meta-Data Lost

Ambiguity and Context

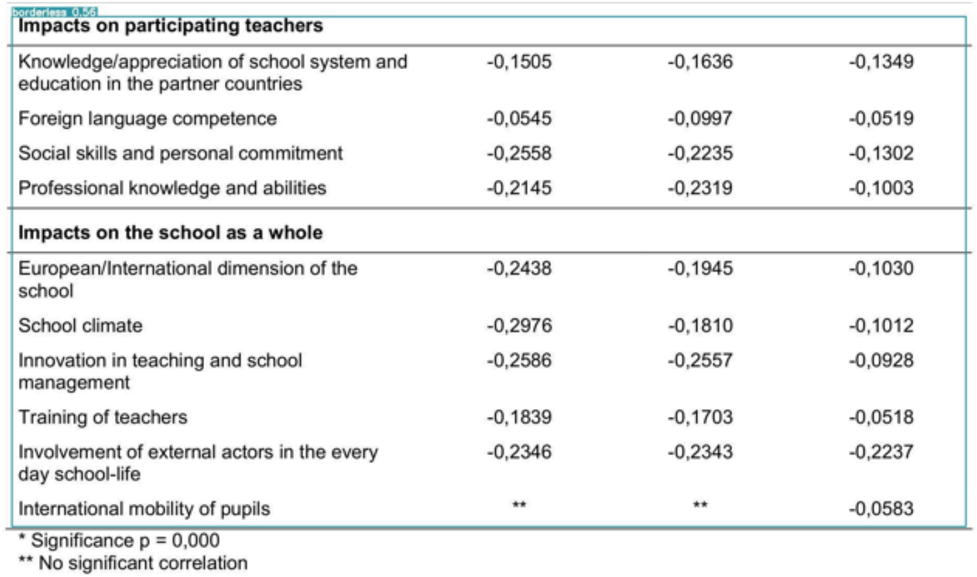
Correct table schema generation

Pipeline: TSR



Yolov8-table Extractor

A CNN-backbone-based deep learning model that is trained on a custom dataset on COCO format



Impacts on participating teachers			
Knowledge/appreciation of school system and education in the partner countries	-0,1505	-0,1636	-0,1349
Foreign language competence	-0,0545	-0,0997	-0,0519
Social skills and personal commitment	-0,2558	-0,2235	-0,1302
Professional knowledge and abilities	-0,2145	-0,2319	-0,1003
Impacts on the school as a whole			
European/International dimension of the school	-0,2438	-0,1945	-0,1030
School climate	-0,2976	-0,1810	-0,1012
Innovation in teaching and school management	-0,2586	-0,2557	-0,0928
Training of teachers	-0,1839	-0,1703	-0,0518
Involvement of external actors in the every day school-life	-0,2346	-0,2343	-0,2237
International mobility of pupils	**	**	-0,0583

* Significance p = 0,000
** No significant correlation

- Fine-tuned on a custom Roboflow dataset for better results

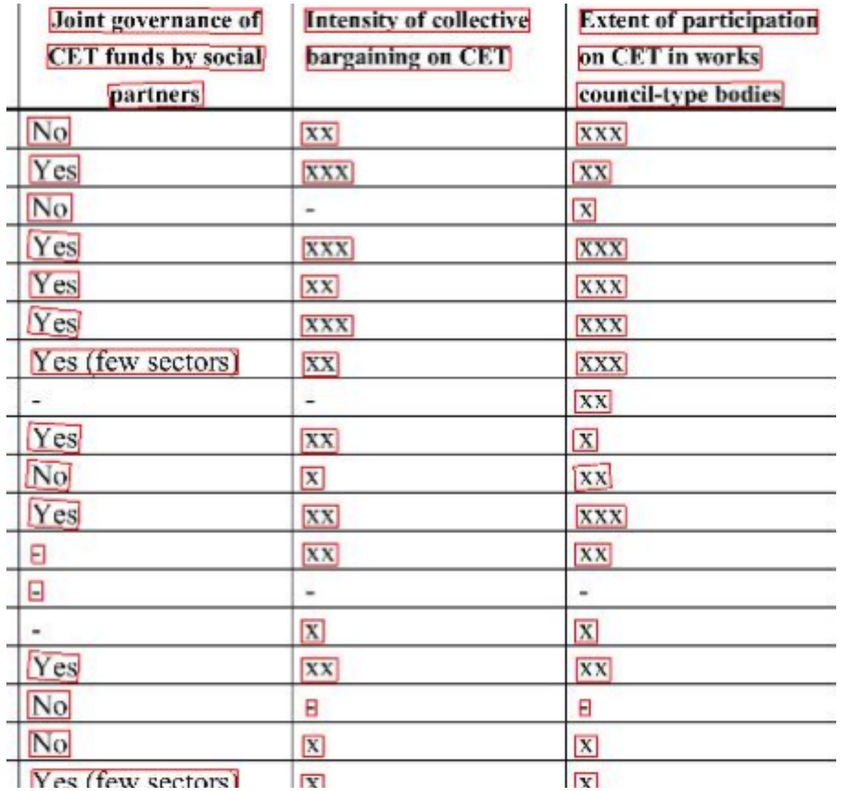
Why is it chosen?

- **Better Performance on Small Objects**: Enhanced detection of smaller items.
- **Enhanced Feature Extraction**: Better detection under various conditions.
- **Flexibility and Scalability**: Suitable for both edge devices and powerful servers.
- **Robustness and Generalization**: Better generalization across different datasets.

Recall	Precision	F1
93.83	91.51	92.66

Paddle-OCR

PaddleOCR is an open-source optical character recognition (OCR) toolkit developed by Baidu's PaddlePaddle deep learning platform



Joint governance of CET funds by social partners	Intensity of collective bargaining on CET	Extent of participation on CET in works council-type bodies
No	xx	xxx
Yes	xxx	xx
No	-	x
Yes	xxx	xxx
Yes	xx	xxx
Yes	xxx	xxx
Yes (few sectors)	xx	xxx
-	-	xx
Yes	xx	x
No	x	xx
Yes	xx	xxx
-	xx	xx
-	-	-
-	x	x
Yes	xx	xx
No	-	-
No	x	x
Yes (few sectors)	x	x

Why is it chosen?

- PaddleOCR employs advanced text detection algorithms to locate and outline text regions within images.
- It generates bounding boxes around detected text regions, which can correspond to individual cells in a table.
- By analyzing spatial relationships between detected text regions, PaddleOCR infers the structure of the table, including rows and columns.
- PaddleOCR facilitates the extraction of text content from each cell once positions are determined.
- It integrates OCR to convert text within each cell into machine-readable formats, enabling further processing or analysis.

Paddle-OCR

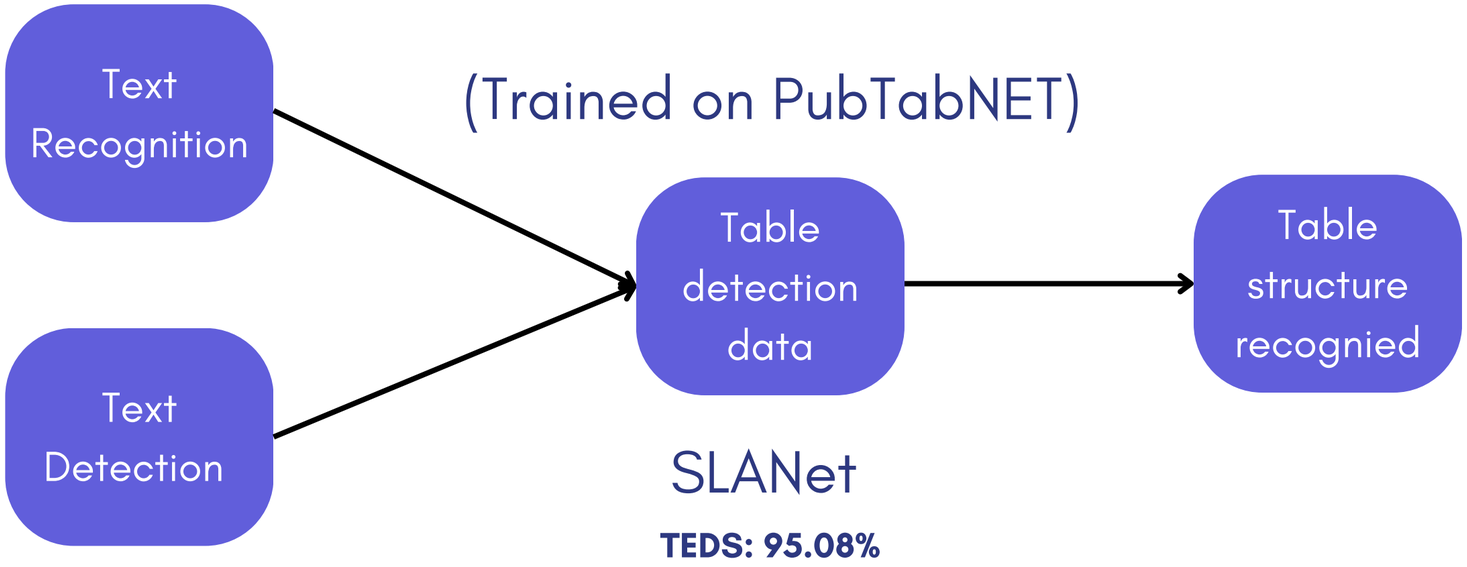
PaddleOCR is an open-source optical character recognition (OCR) toolkit developed by Baidu's PaddlePaddle deep learning platform

B
力指标可以从宏观上反映一个区域水 资源的丰裕/稀缺程度和开发利用程
评价方法 总体供需平衡 比例 管理性缺水 水质性缺水
工程性缺水 资源性缺水 混合性缺水 异常情况下的水资源风险
水量保障程度 水质保障程度 水价承受能力
水分配社会公平 水量保障程度 水质保障程度 经济承受能力
水生态压力 水生态状态 水生态响应 的分析和诊断，必须尽可能全面地

Multi-Modal TSR

Generation of Structured CSV

- After extraction of table features is done
- A structured CSV is made which retains the spatial and logical structure of the table.



Future Scope

- **Current Limitation:**
- The existing TSR models primarily support text recognition in Chinese and English, which limits their applicability to documents in other languages.
- **Future Direction:**
- **Multilingual Text Recognition:** Future work should focus on training TSR models using different text recognition algorithms that support a broader range of languages. This involves integrating advanced OCR (Optical Character Recognition) systems capable of recognizing text in various scripts and fonts.
- **Dataset Expansion:** Curating diverse datasets containing tables in multiple languages will be crucial for training and evaluating these models. This can include multilingual documents from scientific literature, business reports, and other sources.

Conclusion

- **Named Entity Recognition (NER)**
  - Tools: Fine-tuned transformer models such as LUKE and DistilBERT
  - Benefits: Automates key entity identification
  - Impact: Enhances accuracy and reduces manual effort

- **Text Summarization**
  - Tools: Text-Rank, Pegasus and BART
  - Benefits: Condenses extensive texts into concise summaries
  - Impact: Saves time and mitigates information overload

- **Table Structure Recognition (TSR)**
  - Tools: YOLOv8 and PaddleOCR
  - Benefits: Accurately extracts and preserves relationships in tabular data
  - Impact: Improves efficiency, scalability, and integration with enterprise systems

- **Overall Impact** - These advanced AI-driven solutions collectively streamline document processing, enhance data management, and integrate seamlessly with enterprise systems, positioning Intelligent Document Processing (IDP) as a pivotal technology.

Thank You