

# Gutenbrain: An Architecture for Equipment Technical Attributes Extraction from Piping & Instrumentation Diagrams

Marco Vicente<sup>1</sup><sup>a</sup>, João Guarda<sup>2</sup><sup>b</sup> and Fernando Batista<sup>3</sup><sup>c</sup>

<sup>1</sup>Leonard / Vinci Group, Paris, France

<sup>2</sup>Axians Digital Consulting, Lisbon, Portugal

<sup>3</sup>INESC-ID, Human Language Technologies Department, Lisbon, Portugal


**Keywords:** Information Retrieval, Question-answering, Piping & Instrumentation Diagrams.


**Abstract:** Piping and Instrumentation Diagrams (P&ID) are detailed representations of engineering schematics with piping, instrumentation and other related equipment and their physical process flow. They are critical in engineering projects to convey the physical sequence of systems, allowing engineers to understand the process flow, safety and regulatory requirements, and operational details. P&IDs may be provided in several formats, including scanned paper, CAD files, PDF, images, but these documents are frequently searched manually to identify all the equipment and their inter-connectivity. Furthermore, engineers must search the related technical specifications in separate technical documents, as P&ID usually don't include technical specifications. This paper presents Gutenbrain, an architecture to extract equipment technical attributes from piping & instrumentation diagrams and technical documentation, which relies in textual information only. It first extracts equipment from P&IDs, using meta-data to understand the equipment type, and text coordinates to detect the equipment even when it is represented in multiple lines of text. After detecting the equipment and storing it in a database, it allows retrieving and inferring technical attributes from the related technical documentation using two question answering models based on BERT-like contextual embeddings, depending on the equipment type meta-data. One question answering model works with free questions of continuous text, while the other uses tabular data. This ensemble approach allows us to extract technical attributes from documents where information is unstructured and scattered. The performance results for the equipment extraction stage achieve about 97,2% precision and 71,2% recall. The stored information can be later accessed using Elasticsearch, allowing engineers to save thousands of hours in maintenance engineering tasks.


## 1 INTRODUCTION

In the Oil & Gas Upstream industry, the life cycle of any new asset (offshore or onshore) starts with the Engineering, Procurement & Construction phase (EPC). The first activity of a Maintenance & Inspection Engineering Contract (also called a MIEC) is to create the asset register, which is the hierarchy of all the equipment. This is the very first mandatory milestone of a MIEC to be able to move on to other activities such as criticality studies (to define the criticality of each equipment), or the definition of the spare parts to be procured and put in stock for later use. And finally, definition of maintenance plans, proce-

dures, and manuals, and link them to the asset register. The asset register is created based in the information provided in Piping and Instrumentation Diagrams (P&ID). P&IDs are detailed representations of engineering schematics with piping, instrumentation and other related equipment and their physical process flow. P&IDs are critical in engineering projects to convey the physical sequence of systems, allowing engineers to understand the process flow, safety and regulatory requirements, and operational details. P&IDs are provided in several formats: scanned paper, CAD files, PDF, images. Usually, these documents are searched manually to identify all the equipment and their inter connectivity. Furthermore, engineers must search technical specifications in separate technical documents, as P&ID usually don't include technical specifications. The process consists in gathering all documentation coming from EPC con-

<sup>a</sup> <https://orcid.org/0000-0003-1123-9917>

<sup>b</sup> <https://orcid.org/0000-0002-5733-9553>

<sup>c</sup> <https://orcid.org/0000-0002-1075-0177>

tractor, manufacturers, and suppliers. Each of these thousands of documents must be searched manually to identify all the tags of the equipment, group them by system and sort them as per their parent/child relationship. When a new revision of such a document appears, it must be processed again to make sure there is no impact on the asset register. So, this documentation searching activity is heavy, time consuming and with low value for the project. The information is hard to find, and the manual processing and the redundancy of the activity sometimes leads to human error.

This paper presents Gutenbrain, an architecture to extract equipment technical attributes from piping & instrumentation diagrams and technical documentation, which relies in textual information only. It first extracts equipment from P&IDs, using meta-data to understand the equipment type, and text coordinates to detect the equipment even when it is represented in multiple lines of text. After detecting the equipment and storing it in a database, it allows retrieving and inferring technical attributes from the related technical documentation using two question answering models based on BERT-like contextual embeddings, depending on the equipment type meta-data. One question answering model works with free questions of continuous text, while the other uses tabular data. This ensemble approach allows us to extract technical attributes from documents where information is unstructured and scattered. The stored information can be later accessed using Elasticsearch, allowing engineers to save thousands of hours in maintenance engineering tasks.

This document is organised as follows: Section 2 presents an overview of the related literature. Section 3 describes the proposed architecture. Section 4 reports the achieved evaluation results. Finally, Section 5 presents the main conclusions, and pinpoints future working directions.

## 2 RELATED WORK

This section presents an overview of the existing literature. It starts by focusing on the literature concerning information extraction from diagrams, and then focuses on the retrieval and inference of technical attributes from additional data.

### 2.1 Information Extraction from P&IDs

Literature on data extraction for piping & instrumentation diagrams of for engineering drawings is scarce. Most of the current techniques are based on Computer Vision algorithms and machine learning models.

Nonetheless, there is a gap between this specific domain and state-of-the-art techniques and algorithms. (Moreno-García et al., 2019) presents a comprehensive study on the techniques used in the piping & instrumentation diagrams domain, referring that most work that has been done in this field focus on using computer vision models and algorithms to extract attributes based on the shapes present in the documents.

The early work reported by (Yu et al., 1997) uses a set of rules applied to the lines of a symbol to classify it on generic engineering drawings, and not only P&IDs, and each symbol has a set of rules. (Wenyin et al., 2007) proposes a similar method by creating a database of symbols which were described by four geometric constrains extracted by an algorithm. Although generic and good performance, this approach requires a pre-processing of all symbols to be detected. More recently, (Fu and Kara, 2011) proposes the use of a multi-scale sliding window and Connected Component Analysis to locate the symbols and a Convolutional Neural Network (CNN) to classify them. This model requires labelled data to learn, and therefore a large sample of data has to be labelled in order to apply this method.

In the last few years, researchers applied several Computer Vision techniques in the P&ID domain. (Elyan et al., 2018) proposed a heuristic to locate symbols and random forest combined with clustering for the classification. Research also focused on extracting the relationships between the symbols. (Kang et al., 2019) proposed not only a method to extract symbols (with contour algorithms) but also extracting the text with OCR and establishing relationships between symbols by extracting connection lines. A more robust and modern methodology was proposed by (Rahul et al., 2019). They use a Fully Convolutional Network (FCN) and can do all the segmentation (detection and classification) with a single model. It also uses rules to detect connections and relationships. Finally, (Gao et al., 2020) uses the ResNet-50 (He et al., 2015), a Faster Regional Convolutional Neural Network (Faster RCNN), backbone with data augmentation techniques to detect and classify symbols. Once again, rules are applied to infer connections and relationships.

### 2.2 Retrieval of Technical Attributes

Pre-trained transformer models are the state-of-the-art for question-answering (Q&A) tasks. One of them is DistilBERT (Sanh et al., 2019), a distilled version of BERT (Devlin et al., 2018) that leverages the complex architecture that BERT was trained on but is faster, smaller and lighter. To further improve the perfor-

mance of the model in a question-answering task, usually, it is fine-tuned using the (Rajpurkar et al., 2016) dataset. It has more than 100,000 question-answers pairs from Wikipedia.

The models described before don't work well with tabular data. Therefore, (Herzig et al., 2020) introduces TAPAS, an extension to BERT to encode tables as input, to do question-answering over tables without using logical forms as previous literature suggested. Recently, (Chen et al., 2020) proposed a method to apply Q&A to both textual and tabular data. "Early fusion" is used to fuse tabular and textual units into a block and a cross-block reader to capture the dependency between multiple evidence.

These techniques have been applied in the instruction manuals domain to extract technical attributes. (Abinaya Govindan and Verma, ) proposed a pipeline for dealing with image, text and tabular data using pre-trained Q&A models. (Nandy et al., 2021) also proposed a pipeline, for text only, using a model built on RoBERTa (Liu et al., 2019).

### 3 PROPOSED ARCHITECTURE

This section describes the proposed two-steps methodology for extracting technical attributes from P&ID and technical documentation in detail.

#### 3.1 P&ID Information Extraction

To be able to detect equipment in P&IDs, we first ingest all documentation into a database. Then, we use this information and meta-data to detect equipment. Figure 1 shows the pipeline for this initial process.

##### 3.1.1 Extract Text and Coordinates from P&IDs

The first step consists in ingesting all P&IDs of a project to extract the embedded data. We start by extracting all the text and text coordinates into a database. Documents can contain text, raster images, vectors and text without Unicode mapping (usually it happens in documents exported from CAD to PDF without the font used). We start by extracting native text, and bounding boxes. The bounding boxes are useful information, containing the coordinates of the text. When we detect characters without Unicode mapping, we send the bounding box of this piece of undetectable text to Optical Character Recognition (OCR), using Google OCR API. After extracting all the native text, we remove all text from the PDF and if the pages still have information, we send the pages stripped of native text to OCR and store the received

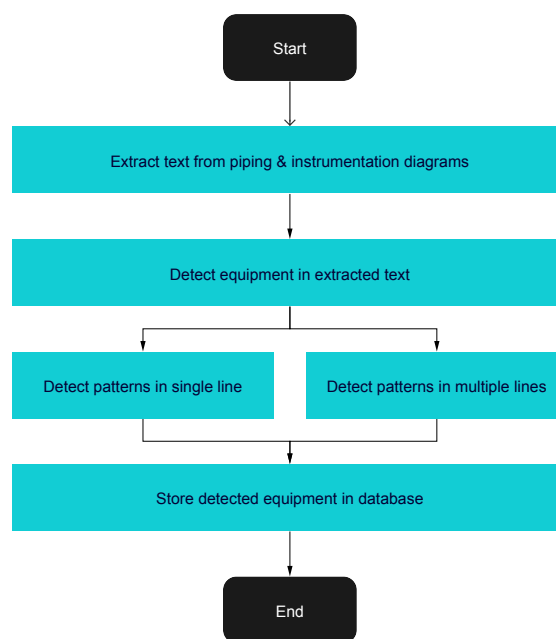


Figure 1: P&IDs information extraction pipeline.

text and coordinates into our database. An example of the extracted text and bounding boxes can be seen in Figure 2. All information extracted is stored in the database.

##### 3.1.2 Detect Equipment in Extracted Text

In the industry, equipment and piping numbering are structured according to rules with meaningful information. The numbering of a piping or a equipment will provide, at least, its equipment type, system, subsystem and sequential number. Using regular expressions, we detect the equipment present in each P&ID. Having the equipment tag, we can infer its equipment type (e.g.: water pump), system (e.g.: Water Processing) and subsystem (e.g.: Filtration). The challenge with this simple approach is that sometimes equipment tags are multi-line, as we can see in Figure 3. We developed a function to detect the nearest text boxes of a determined text, using the coordinates of the bounding boxes. If a text is composed of the beginning of a pattern, we validate if the surrounding bounding boxes are the missing parts. After extracting all the equipment, equipment type, system and subsystem, we store all this information in the database.

#### 3.2 Retrieving and Inferring the Technical Attributes

To make informed decisions, engineers still need technical information regarding each equipment. For

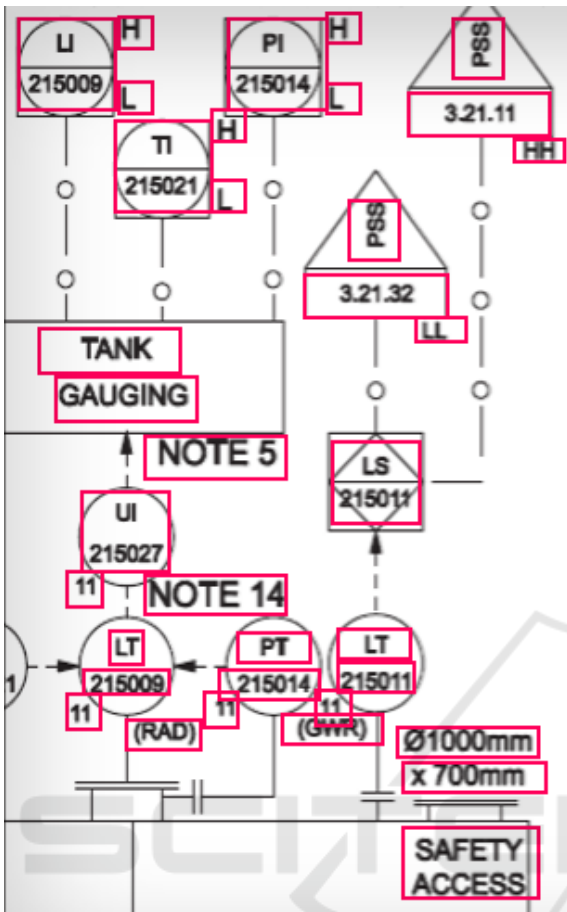


Figure 2: Extracted text.

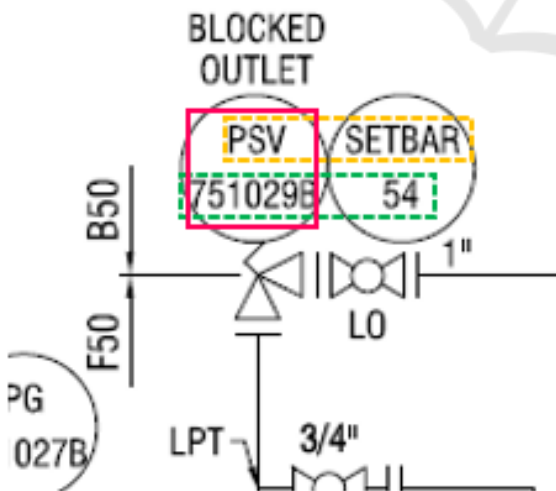


Figure 3: Multi-line equipment tags.

this, we developed a pipeline to extract relevant information regarding each equipment.

### 3.2.1 Preprocess Technical Documentation

We ingest all technical documentation of a project to extract the embedded data. We start by processing data from thousands of unstructured files into a database. These files are comprised by textual, tabular and technical drawing data, sometimes stored in legacy data formats. For each document we store the type of equipment they refer to, and we store the extracted text in two formats: continuous text and tabular data. This distinction will be useful to later steps, when deciding what model to use to extract technical attributes. The ingestion of technical documentation is fully automatized. Figure 4 shows the preprocessing of technical documentation.

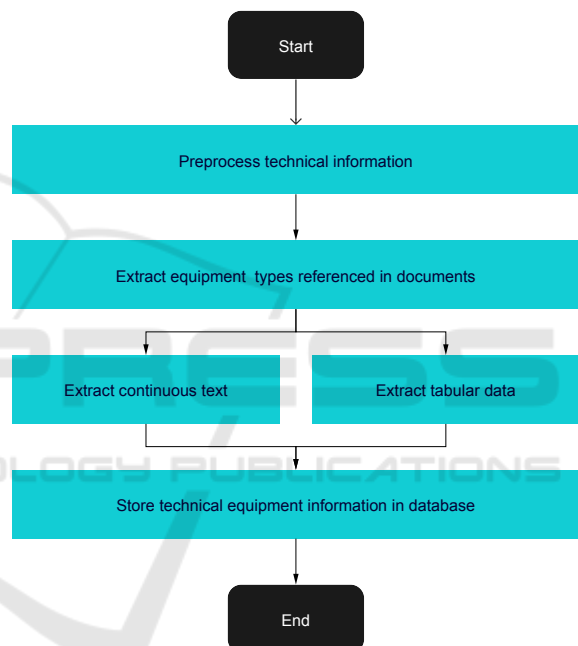


Figure 4: Technical documentation preprocessing pipeline.

### 3.2.2 Retrieval Technical Attributes

To be able to extract technical information for each equipment type, we have created a database of 693 types of equipment, and for each equipment we filled their attributes. Each equipment can have from 1 to 20 technical attributes, as illustrated in the examples presented in Figure 1.

Table 1: Examples of equipment and its corresponding attributes.

Equipment	Attribute 1	Attribute 2
Amplifier	Input	Output
Air Conditioner	Type	Power
Centrifugal Pump	Model	Flow Rate

Having this information loaded, we run a search for each equipment. First, we read its equipment type. Second, we search in the technical documentation for documents classified as having information regarding that type of equipment. Third, we divide the information of these documents in two separate contexts, namely: continuous text information, and tabular information. Forth, we generate questions for each technical attribute. For the question construction, we send a full text question. e.g.: what is the power of the air conditioner. Finally, we send the questions and the context information to two separate question answering models. The continuous text information is used as context to the question answering model DistilBERT (Sanh et al., 2019). The model used is a fine-tune checkpoint of DistilBERT-base-uncased, fine-tuned using knowledge distillation on SQuAD v1.1. The tabular information is sent to the question answering model TAPAS (Herzig et al., 2020). TAPAS is a BERT-like transformers model, pretrained on raw tables and associated texts, allowing to query tabular information. Finally, we compare the best result of each model and save in the database the one with higher score. Figure 5 shows the extraction of technical attributes.

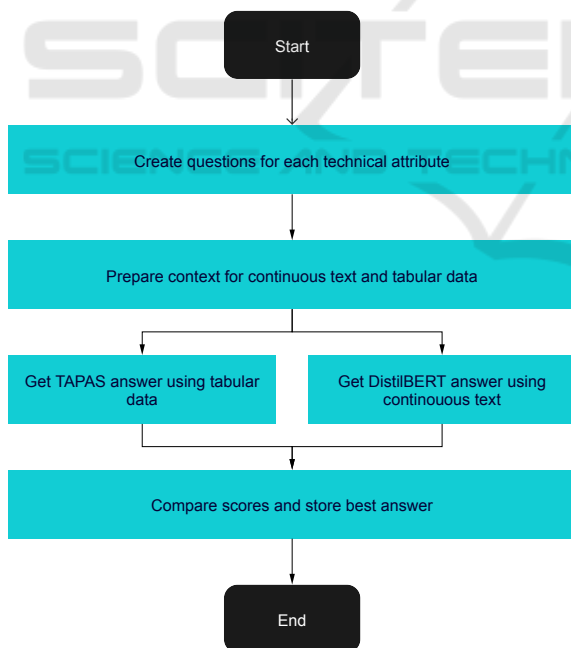


Figure 5: Retrieval of technical attributes.

### 3.3 Supporting Architecture

To provide Gutenbrain functionalities to users, we rely on a cloud-based architecture. The user interface is built in React, a JavaScript library. It allows users to

visualise and edit extracted information from P&IDs. All the back-end functionalities are available through REST APIs developed with fast-API and deployed in Uvicorn, an ASGI server. Information extracted is stored in MongoDB. MongoDB allows queries with regular expressions, critical to equipment detection in the P&IDs. To allow users to query semantically in the information of all equipment, we use Elasticsearch, a search engine based on the Lucene library. Documents are stored in a persistent storage account, enabling the usage of containers for all other building blocks of the architecture. Finally, Google OCR is used to extract text from images. An overview can be seen in Figure 6.

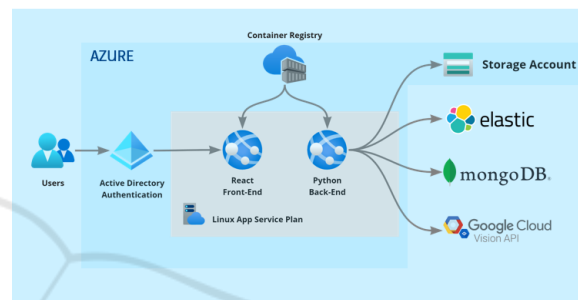


Figure 6: Reference Architecture.

## 4 EXPERIMENTS AND RESULTS

This section describes the results obtained for the proposed Gutenbrain architecture, also reporting on the faced challenges. It starts by focusing on the stage of extracting information from P&ID, and then on the stage of retrieving technical attributes from the stored data.

### 4.1 P&ID Information Extraction

#### 4.1.1 OCR

Most of the P&IDs are scanned, have images, or are exported from CAD files into vectors. The use of OCR is essential to extract that text information. To be able to extract equipment from diagrams using OCR, three main challenges arise.

First, in the context of diagrams, text is sometimes overlapped with symbols, as shown in Figure 7. This causes two types of error: 1) only part of the text is extracted, ignoring the characters overlapped with symbols, or 2) overlapped shape and characters are merged and the retrieved text is incorrect. In both cases, the equipment tag is unrecoverable, as we have partial or incorrect information.

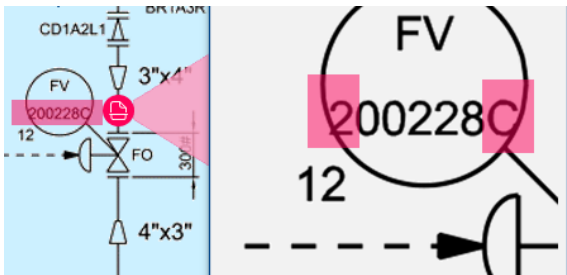


Figure 7: Text overlapped with symbol.

Second, multiple lines are hard to detect when there is little space between, as we see in Figure 8. This means that relevant information is lost, either by removing full equipment or part of multi-line equipment.

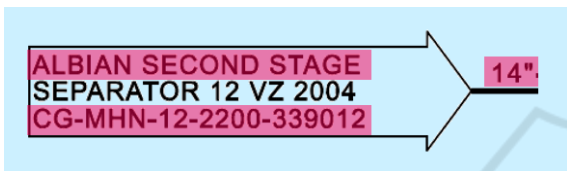


Figure 8: Multi-line text.

And finally, sometimes OCR adds spaces where there are none. With these nonexistent spaces, two things can occur: 1) the equipment pattern is not detected, as it contains spaces, or 2) the OCR detects the text as separate words or separate text blocs. An example can be seen in Figure 9.

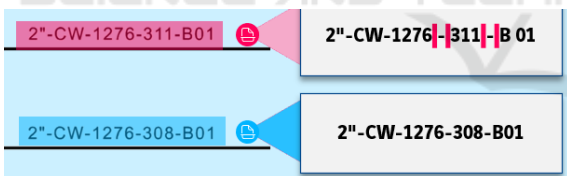


Figure 9: Nonexistent spaces added by OCR capture.

To tackle these challenges, we performed an OCR benchmark between Tesseract, Azure and Google OCR. Tesseract is convenient, as it works on-premises, but its performance is sub-par when comparing with both Azure and Google. When comparing Azure and Google OCR regarding how they tackle the three above mentioned challenges, Google outperformed Azure in all of them. Figure 10 shows how Google was able to detect text even with overlapped symbols.

We are currently using Google OCR to extract text from all diagrams and technical documentation.

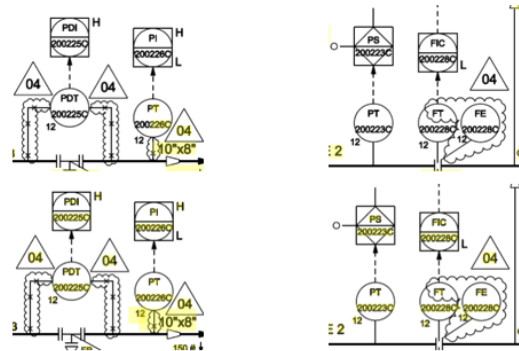


Figure 10: Azure results above. Google results below.

#### 4.1.2 Equipment Sanitisation

When extracting equipment, their codes can have different separation of components, as they are usually not normalised. Sometimes they have hyphen separating attributes (equipment type, system, subsystem), other times they have space, and others they are in multiple lines, having only the break of line to separate them, as shown in Figure 11. To ensure the same equipment is considered as such if it appears in several documents, we sanitise the equipment. We add hyphen where a space or line break is found when storing in the database.

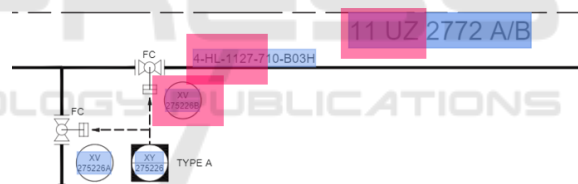


Figure 11: Several equipment patterns.

#### 4.1.3 Equipment Extraction Results

To be able to validate the equipment extraction process, we have annotated manually a dataset of P&IDs containing 607 equipment records. These documents are very big, but unlike the computer vision approaches where documents are split and shrunk, we enlarged them in order to improve the OCR results. The dataset is composed of a mix of documents with native text, images, vectors and some text without Unicode mapping.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

The performance of the system was evaluated using *Precision* and *Recall*, two metrics commonly used

in the literature and defined in equations 1 and 2, where  $TP$  represents true positives,  $FP$  represents false positives, and  $FN$  represents false negatives. We also present the  $F1$ -measure, a metric that combines  $precision$  and  $recall$ . Table 2 shows the achieved results, revealing that our system is able to achieve an impressive performance, specially in terms of Precision. We can see that the recall is highly affected when the document must be processed with OCR in order to retrieve the text, since it introduces errors.

Table 2: Performance results for the equipment extraction stage.

Results	Precision	Recall	F1-measure
Text	0,984	0,824	0,897
OCR	0,960	0,600	0,738
Average	0,972	0,712	0,822

Current studies of P&ID information extraction use image-based techniques, recognising symbols through template matching. When compared with the approach of previous works, our architecture achieves better results. Comparison is shown in Table 3.

Table 3: Performance results comparison.

Method	Authors	Precision
Symbol detection	(Rahul et al., 2019)	0,799
Symbol detection	(Kang et al., 2019)	0,90
Text detection	Gutenbrain	0,972

## 4.2 Retrieval and Inference of Technical Attributes

### 4.2.1 Preprocessing of Technical Documentation

To be able to use technical documents' information in the question answering models, data was extracted to the database in two different stacks: 1) continuous text, and 2) tabular data. This will be important when extracting technical attributes, as it is different to search in continuous text or in tabular data. Besides extracting this information, each document was also classified with the equipment type it contained information.

### 4.2.2 Retrieval of Technical Attributes

The goal of retrieving technical attributes is to find all the characteristics for each equipment, according to its type. e.g.: for a centrifugal pump, we want to know its model, use, flow rate, output, pressure, intake temperature, viscosity and input connection.

The first step is to find the documents that have

information regarding the identified equipment type. After finding the subset of relevant documents, we create a set of questions to be used in both question answering models. We use questions in full sentences: "what is the <attribute> of the <equipment type>". e.g.: what is the flow rate of the centrifugal pump? The next step is to send the question and all continuous text as context to our DistilBert model. The model gives us the best answer and their score. Afterwards, we send the same question to our TAPAS base model, but using as context all tabular information found in the selected documentation. Again, it will give us the best answer and their score. To extract answers from tabular data, our approach is similar to the one proposed by (Chen et al., 2020). We compare the score of each model and store the best result in our database. Our suggestion to the technical attribute will be the result with best confidence among both models. As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Masked Language Model (MLM) objective enables the representation to use both the left and the right context, which allows to pre-train a deep bidirectional Transformer. Since maintenance engineering has special jargon, BERT's sub-word representations and word-piece tokenization are useful for the out-of-vocabulary words that often appear in the corpus. This only mitigates the special jargon issue to fully solve it the models should be fine-tuned to our specific domain. BERT's architecture works well for this task-specific finetuning, since it was trained on a large corpus

Table 4 shows some examples of questions, and the corresponding answer with the associated score.

Table 4: Some examples of questions.

Question	Answer	Score
What is the wattage of the lamp?	26W	79,30%
What are the dimensions of the Converter?	40 x 119 x 115 mm	54,10%
What is the manufacturer of the light?	SCHNEIDER ELECTRIC	82,89%

We were able to find a great part of the technical attributes from equipment, and, when the confidence was low, we were still able to show engineers the right documents for them to extract manually. As we are using extractive question and answering models, we can show the documents in the right pages for engineers to validate the extracted attributes.

## 5 CONCLUSIONS AND FUTURE WORK

We have proposed an approach to extract equipment and technical attributes from P&IDs and retrieve technical documentation from technical sheets, and described an architecture to support this approach. We have performed experiments on a manually labelled dataset of P&IDs, containing 607 equipment, and the performance results for the equipment extraction stage achieve about 97,2% precision and 71,2% recall.

In the Oil & Gas Upstream industry, EPC projects take in average 90k hours of technical engineers to create the asset register, to do criticality studies (to define the criticality of each equipment), the definition of the spare parts to be procured and put in stock for later use, and the definition of maintenance plans, procedures, and manuals. The proposed architecture, Gutenbrain, allows the saving of 16k per project, either by extraction automatically the equipment information or by allowing to search in the technical information using semantic search in content otherwise unsearchable. The experimental validation presents an average reduction of approximately the 60% of engineers' effort in cumbersome tasks of extracting equipment information allowing the saved hours to be spent by engineers on tasks with higher value. This approach still requires users to validate the extracted information and extract the undetected information, but we provide a user interface for engineers to have the autonomy to do so.

In the future we would like to fine-tune the question-answering models with closed-context data from past projects to improve the results. We would also like to complement the equipment extraction with the computer vision approach of detecting symbols within diagrams. The hypothesis being that an ensemble method using text and symbols might outperform our current approach.

## REFERENCES

- Abinaya Govindan, G. R. and Verma, A. Intelligent question answering module for product manuals.
- Chen, W., Chang, M.-W., Schlinger, E., Wang, W., and Cohen, W. W. (2020). Open question answering over tables and text. *arXiv preprint arXiv:2010.10439*.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Elyan, E., Garcia, C. M., and Jayne, C. (2018). Symbols classification in engineering drawings. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Fu, L. and Kara, L. B. (2011). From engineering diagrams to engineering models: Visual recognition and applications. *Computer-Aided Design*, 43(3):278–292.
- Gao, W., Zhao, Y., and Smidts, C. (2020). Component detection in piping and instrumentation diagrams of nuclear power plants based on neural networks. *Progress in Nuclear Energy*, 128:103491.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.
- Herzig, J., Nowak, P. K., Müller, T., Piccinno, F., and Eisen-schlos, J. M. (2020). Tapas: Weakly supervised table parsing via pre-training.
- Kang, S.-O., Lee, E.-B., and Baek, H.-K. (2019). A digitization and conversion tool for imaged drawings to intelligent piping and instrumentation diagrams (p&i;d). *Energies*, 12(13):2593.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pre-training approach. *arXiv preprint arXiv:1907.11692*.
- Moreno-García, C. F., Elyan, E., and Jayne, C. (2019). New trends on digitisation of complex engineering drawings. *Neural computing and applications*, 31(6):1695–1712.
- Nandy, A., Sharma, S., Maddhashiya, S., Sachdeva, K., Goyal, P., and Ganguly, N. (2021). Question answering over electronic devices: A new benchmark dataset and a multi-task learning based qa framework. *arXiv preprint arXiv:2109.05897*.
- Rahul, R., Paliwal, S., Sharma, M., and Vig, L. (2019). Automatic information extraction from piping and instrumentation diagrams. *arXiv preprint arXiv:1901.11383*.
- Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.
- Wenyin, L., Zhang, W., and Yan, L. (2007). An interactive example-driven approach to graphics recognition in engineering drawings. *International Journal of Document Analysis and Recognition (IJ DAR)*, 9(1):13–29.
- Yu, Y., Samal, A., and Seth, S. C. (1997). A system for recognizing a large class of engineering drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):868–890.