Ontology-based Information Extraction from Technical Documents

Syed Tahseen Raza Rizvi[1,2], Dominique Mercier[2], Stefan Agne[1], Steffen Erkel[3], Andreas Dengel[1] and Sheraz Ahmed[1]

[1]*German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany*

[2]*Kaiserslautern University of Technology, Kaiserslautern, Germany*

[3]*Bosch Thermo-technology, Lollar, Germany*

Keywords: Table Detection, Information Extraction, Ontology, PDF Document, Document Analysis, Table Extraction, Relevancy.

Abstract: This paper presents a novel system for extracting user relevant tabular information from documents. The presented system is generic and can be applied to any documents irrespective of their domain and the information they contain. In addition to the generic nature of the presented approach, it is robust and can deal with different document layouts followed while creating those documents. The presented system has two main modules; table detection and ontological information extraction. The table detection module extracts all tables from a given technical document while, the ontological information extraction module extracts only relevant tables from all of the detected tables. The generalization in this system is achieved by using ontologies, thus enabling the system to adapt itself, to a new set of documents from any other domain, according to any provided ontology. Furthermore, the presented system also provides a confidence score and explanation of the score for each of the extracted tables in terms of its relevancy. The system was evaluated on 80 real technical documents of hardware parts containing 2033 tables from 20 different brands of Industrial Boilers domain. The evaluation results show that the presented system extracted all of the relevant tables and achieves an overall precision, recall, and F-measure of 0.88, 1 and 0.93 respectively.

1 INTRODUCTION

Tabular data representation is one of the most common way of presenting a lot of information in compact form. Mostly, the tables are relatively simple but sometimes a piece of information is shared between multiple rows or columns in the form of merged rows or columns. Technical documents usually contain hundreds of pages with dozens or hundreds of tables. Most of the times, we are interested in only a few tables among all tables in a document.

A lot of solutions have been proposed so far for table detection and extraction but they were designed to work on a specific set of documents with a known layout. Furthermore, there are a bunch of complicated cases for merged rows and columns within a table. Sometimes data needs to be duplicated among merged rows or columns. While sometimes there could be possibility for an empty row, column or a cell. Existing systems can not handle complex table structures or empty cells, thus spoiling the final output. Also, previous systems were extracting all tables from a given document which is a very rare use case. But most of the time, we are interested only in a few tables of our concern from a document.

The objective of this work is to extract only relevant tables from given documents in a portable form which could be conveniently plugged into any system for direct usage.

2 RELATED WORK

This section provides an overview of different solutions available for information extraction from documents with table.

(Milosevic et al., 2016) proposed a rule based solution for extracting table data from tables in clinical documents in which the data is firstly decomposed into cell level structures depending on their complexity and then information is extracted from these cell structures. (Gatterbauer and Bohunsky, 2006) proposed a solution based on spatial reasoning in which a visual box is drawn around each of the HTML DOM

Rizvi, S., Mercier, D., Agne, S., Erkel, S., Dengel, A. and Ahmed, S.
Ontology-based Information Extraction from Technical Documents.
DOI: 10.5220/0006596604930500
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 2*, pages 493-500
ISBN: 978-989-758-275-2

element. Based on the alignment, certain visual boxes were merged together to form a hyper box. Eventually a table is segregated from other HTML DOM elements and information is extracted from this table.

(Ramakrishnan et al., 2012) presented 3 stage process for extracting text from layout aware PDF scientific articles. In which firstly, contiguous blocks of text are detected and then classifying them in different categories based on predefined rules. And finally stitching blocks together in correct order.

(Ruffolo and Oro, 2008) proposed an ontology based system, known as XONTO, for semantic information extraction from PDF documents. This system makes use of self-describing ontologies which help in identifying the relevant ontology object from the text corpus. (Chao and Fan, 2004) proposed a technique that extract layout and content information from a PDF document. Logical components of document, i.e. outline, style attributes and content, are identified and extracted in XML format.

(Rosenfeld et al., 2002) proposed a system which makes use of a learning algorithm known as structural extraction procedure. It extracts different entities from the text based on their visual characteristics and relative position in the document layout. (Liu et al., 2006) also proposed an approach which is used to extract meta-data, i.e. rown and column number, information from digital documents which could further be used to understand semantics of the textual content.

(Pinto et al., 2003) proposed the use of conditional random fields (CRFs) for the task of table extraction from plain-text government statistical reports. CRFs support the use of many rich and overlapping layout & language features. Later on tables were located and classified into 12 table related categories. This paper also discussed future extension of this work for segmentation of columns, finding cells and classifying them as data cells. (Tengli et al., 2004) proposed a technique that exploits format cues in semi-structured HTML tables. Then it learns lexical variants from training samples and matches labels using vector space. This approach was evaluated by applying it to 157 university websites.

(Peng and McCallum, 2006) proposed an approach, based on CRFs for constraint co-reference information. In this approach, several local features, external lexicon features and global layout features. (Chang et al., 2006) performed a survey of approaches for information extraction from web pages. The comparison between different systems was performed based on three factors. Firstly, the extent to which a system failed to handle any web page. Secondly, the quality of technique used. Thirdly, degree of automation

(Freitag, 1998) observed the task of information extraction from the perspective of machine learning. The proposed approach suggested the implementation of a relational learner for information extraction task. Where extensible token oriented feature set, consisting of structural and other information, is provided as input to the system. Based on the input, system learns extraction rules for given specific domain. (Rahman et al., 2001) proposed a solution for automatically summarizing content from web pages. In this approach, structural analysis of the document is performed followed by decomposition of the document based on extracted structure. Then document is further divided into sub-documents based on contextual analysis. Finally the labeling of a each sub-document is performed.

(Wei et al., 2006) proposed an approach to extract answers from the tables in a document. In which a cell document is created where each table cell has its title or header as metadata. A model was designed for retrieval which ranks the cells using a given language model. This approach was applied to Government statistical websites and news articles. (Adelfio and Samet, 2013) proposed an approach which makes use of CRFs in combination with logarithmic binning specially designed for table extraction task. This approach was proposed for the extraction of a table along with its structural information in the form of schema. This solution could work on web tables as well as tables in spreadsheets. At the end schema also included its characteristics information like row grouping etc.

3 PROPOSED APPROACH

Figure 1 shows the workflow overview of proposed system. The presented system has three major phases i.e. Preprocessing, Ontological information extraction and Reliability assessment. Preprocessing phase involves converting PDF document into HTML document. Ontological information extraction involves table extraction, relevancy assessment, preparing extracted data in memory and exporting into CSV format. The system is generic and can be applied to documents from any domain.

An ontology consists of entities, relationships and instances. Figure 2 shows an example ontology, where there are different entities i.e. Document, Relevant Information, Irrelevant Information, Relevant Terms, Warning Terms, Region and Trade. It can be observed that there are some child entities and they have a ”is a relationship” from child to parent entity .i.e. Region is a Relevant Term. While the entities at

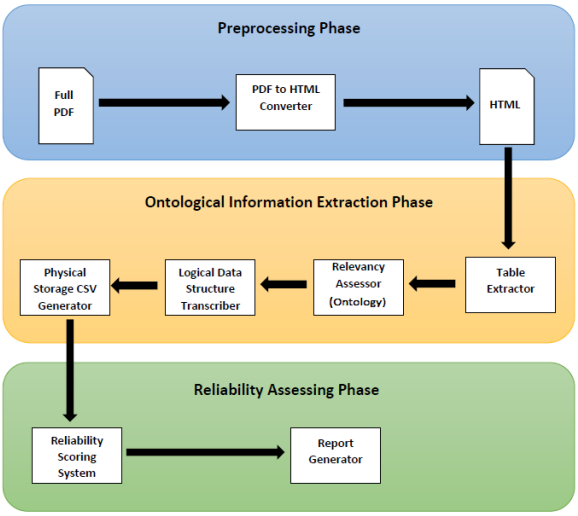


Figure 1: Overview of the workflow of the proposed system.

the bottom have instances like Asia, Europe, Import, Export, Cost, Production etc.

In the given example, the main entity is Document, which includes two different entities Relevant Information and Irrelevant Information. Irrelevant Information consists of an instance "Production". It means that within a given set of documents, this term will always give us a hint that the part of the document under consideration is irrelevant for us. On the Other hand, Relevant Information further consists of Relevant Terms and Warning Terms. Warning terms have an instance "Cost". Which represents that in some context this term may be relevant while in some context it might be not. While Relevant Terms have further two entities Region and Trade. Region has two instances Asia and Europe. While Trade has two instances Import and Export. Which represents that these terms definitely represent the information of our interest.

After understanding the basic components and their relationships of the example ontology, now one needs to understand that what does the ontology in Figure 2 represents. The given example ontology is designed to target statistics in a document related to trade in different regions of the world. There could be some additional rules based on the use case. i.e. Coexistence of multiple entities or exclusive presence of entities define the relevancy of a piece of information in complex use cases. For our system, all the rules provided along with the ontology and ontology itself were used to define heuristics based on which we inspected the relevancy of the information under consideration.

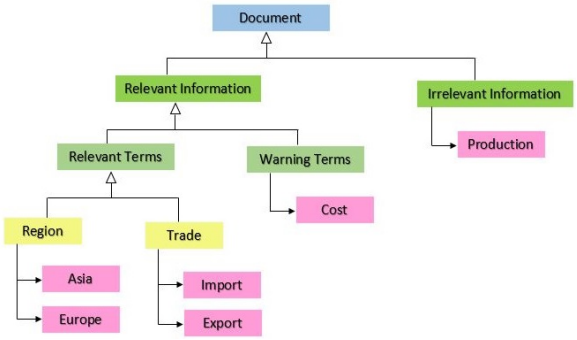


Figure 2: Illustrated example of an ontology.

3.1 Preprocessing Phase

In order to extract information stored in a layout, document needs to be converted into some other intermediary format which can sustain not only text but also the layout in which the text is stored. Layout plays a vital role in building sense about the text stored in the layout. Information stored in a layout connects different bits of information together to form a context.

Conversion of PDF to an intermediary format consists of two crucial steps, Selection of suitable intermediary file format and Conversion from PDF to selected file format.

Selection of suitable intermediary file format is a quite challenging task. There is a wide range of potential formats which can keep text along with layout information attached to it.

Most common file formats are XML, Docx, HTML etc. XML keeps the information stored in a structured and convenient way. But it can not keep layout information. Docx is another potential file format which can keep both textual and layout information. There are a bunch of libraries around for Docx parsing but none of them is reliable libraries to parse Docx file properly. Specially when it comes to complicated tables, those libraries are not so robust and reliable. Lastly, HTML is the file format which not only sustains layout and textual information but is also relatively simple to generate and parse. Additional advantage of selecting HTML format is that, the problems during file format conversion can be quickly identified by visual inspection of HTML in a web browser. For this use case using HTML, due to having most advantages, looks like the most dominant choice for intermediary file format.

On the other hand, quality of generated HTML depends on the tool used for conversion of PDF to HTML. Every tool has its own formatting of resultant HTML as they put the extracted content from PDF into their own customized structures and layouts.

Using a different tool for PDF to HTML conversion, refers to different HTML parser to be used for extraction of text from HTML. The tool used for PDF to HTML conversion in this use case is Adobe Acrobat. Preliminary experimentation proved that Adobe Acrobat is the most reliable choice for format conversion task, as Adobe has almost 23 years of experience in document analysis domain. Also it is very mature product from Adobe, which evolved over years of experience and development. Unlike other tools or open source libraries, Adobe can successfully convert most of the PDF documents to HTML with almost the same look and feel as in original PDF document. On the other hand, other tools and open source libraries either unable to convert some PDF documents due to encoding incompatibilities or are unable to convert PDF to HTML in the correct layout i.e. placing table data out of table layout in resultant HTML file or unexpectedly merging cell data from two different cells of the table into one.

It is to be noted that the final output of the system relies a lot on quality of conversion of PDF to HTML. If there are any errors or mistakes occurred during this conversion phase, then it will also be depicted in the final extracted output. Since the system is designed to extract data out of the document even if there are unexpected column merges or missing table data during conversion process. It will not effect the extraction process but will badly effect quality of extracted data.

3.2 Ontological Information Extraction Phase

The HTML file obtained from preprocessing serves as input to the system. It is to be noted that complete HTML file is fed to the system instead of feeding selective part of HTML file or a subset of the file. The objective of the system is to keep users interruption and effort as less as possible, so that the system is automatically able to find out relevant content by itself.

3.2.1 Table Extractor

HTML file provided as input is then processed to filter out all the tables in document along with their textual contents. In order to extract tables, HTML file is carefully parsed and filtered all tables from the file. HTML tags play an important role in identifying tables in an HTML file. It is to be noted that the tables extracted at this stage are in a raw form. i.e. the data from merged rows or columns only exists just once for all rows or columns sharing that data. The filtered tables are then pruned to keep only those which are relevant to users needs.

3.2.2 Relevancy Assessor

Defining relevancy is sometimes a too subjective task and can vary from one person to another. Thus in order to find out a relevant table, we need to recognize each column title as an entity which is in accordance with the provided ontology. Relevance is decided based on rules and relationships defined, between different entities, in the ontology. In this stage, ontology is used to define heuristics upon which the table filtering is performed. The tables which adhere to the provided ontology are kept while leaving the others.

```
<Report>
 − <File name="111602-0106(43).html_data_table1.csv">
     <Status> Warning </Status>
     <Confidence_Score> 90.0 </Confidence_Score>
   − <Reason>
       Table Titles contain a major relevant term and a warning term as well.
     </Reason>
   </File>
 − <File name="111602-0106(43).html_data_table2.csv">
     <Status> Success </Status>
     <Confidence_Score> 100.0 </Confidence_Score>
     <Reason> Table titles and content look very relevant. </Reason>
   </File>
</Report>
```

Figure 3: Sample output report of the system.

3.2.3 Logical Data Structure Transcriber

Pruned tables based on defined relevancy are then stored in logical structures. It is not as simple as it seems, as tables can have a bunch of cases for merged rows and/or columns. In tables, merged rows or columns means that the piece of data is shared among those merged rows or columns respectively. And sometimes multiple cases can occur simultaneously i.e. A table cell can have merged rows and columns at the same time. In order to overcome all such problems, data from the shared rows or columns needs to be duplicated very carefully among the merged rows or columns respectively.

3.2.4 Physical Storage CSV Extractor

Finally, data from logical structure is stored in some physical storage i.e. Comma separated values (CSV file). The data stored in CSV file is stored in a way that it can be used anywhere, by any text file reading system, without any issue. CSV is quite flexible file format which can be customized to any system requirements.

3.3 Reliability Assessing Phase

Once a system generates output, one is curious to find out that how well the system performed to achieve the given task. The only way to find out is to validate the quality of the output by comparing it to the desired result for a specific input. Depending upon the subjectivity of task and system, there are different measures which can evaluate output of the system: Confidence scoring, Precision, Recall, F-Measure & Accuracy.

3.3.1 Reliability Scoring System

The system generates a separate output CSV file for each table. Thus every output file will be assessed separately and each will be given a separate score computed using defined rules.

The quality of the output plays a key role in defining the rules for confidence scoring of the output. For that purpose, we defined three lists of terms based on ontology 1) Relevant Terms 2) Irrelevant Terms 3) Warning Terms. Relevant terms are those which are related to our topic of interest. Irrelevant terms, as their names suggest, are those which are not related in any manner to our topic of interest. Lastly, Warning terms are those which might be relevant in some context while irrelevant in any other context. Every output table starts with an initial confidence score of 100 at the time of extraction. Later on, the compliance of those tables is checked by the heuristics defined on provided ontology. The confidence score decreases if the titles of the table are not in accordance with relevant terms in our ontology, then the confidence score decreases. Also, if a warning term is spotted, the confidence score decreases to 0.5. Each table is assessed by using these rules and remaining final score at the end represents the extent to which system find that specific table to be relevant.

3.3.2 Report Generator

After computing confidence score for each table, the system reports these statistics to the user. In addition to individual confidence score for each table, system also reports the reason why the score for a particular table is less than 100. Report file consists of 4 data columns i.e. Status, Filename, Confidence Score and Reason. An example of a sample report file is shown in Figure 3.

In the above example, ”warning term” refers to such column titles which have different meanings based on context. Thus it makes the relevancy a bit doubtful. The reasoning along with confidence scoring is self explanatory for the user to understand the reason for that specific score. If the confidence score is reported as 100.0, then the user can directly use that specific file without any doubt. In case of low confidence score for a certain file, user will have to look explicitly into the area of the output file reported in the reason section of the report.

4 EVALUATION

This section discusses dataset details and the results obtained from different experiments performed on the data set. Evaluation of results provide an insight into the strength and robustness of the system.

4.1 Dataset

The dataset consists of 76 documents from 20 different manufactures of industrial boilers. All documents were full text PDF documents. All the documents were randomly divided into Train and Test sets.

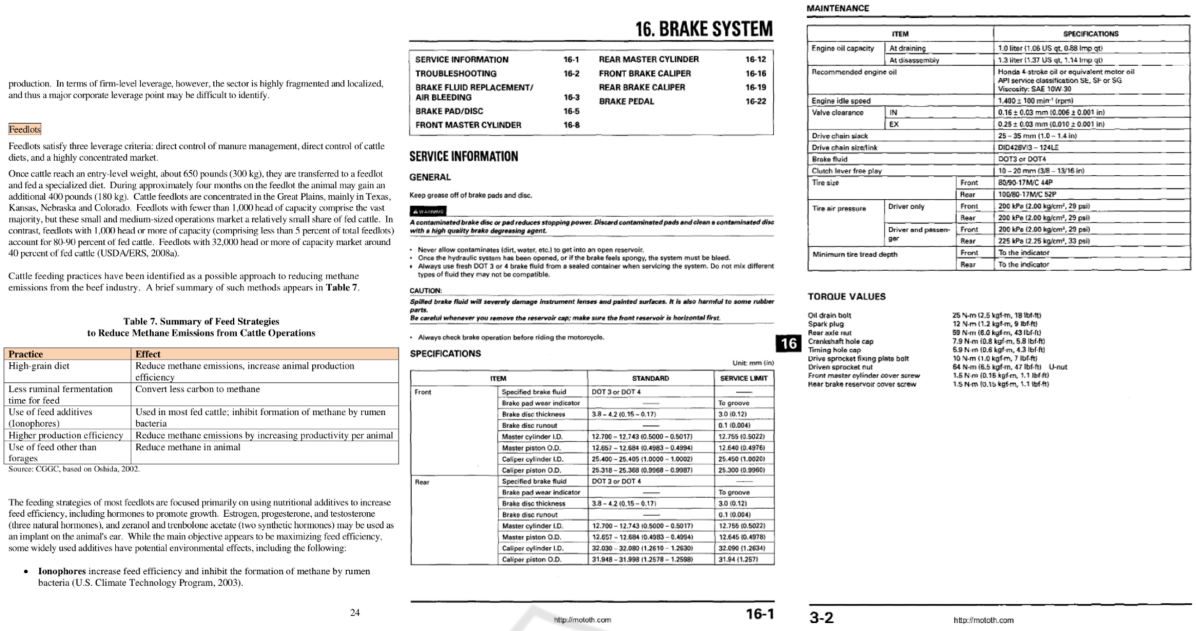
Complexity Levels

Due to huge variations in the document layout and table complexity, all documents were divided into three different difficulty levels based on complexity of their table layouts.

Complexity level 1 is the simplest of all levels as it contains all simple tables, where there is no merged row or column and they have very clear structure. An example of document containing such table is shown in Figure 4a. In training set, 4 documents were designated as level 1 documents. While in test set, 8 documents were allocated to Complexity level 1.

Complexity level 2 is a bit more complex level as compared to Complexity level 1. As it contains cases for merged rows or columns. More specifically, documents in level 2 have either one merged row or column at a time. An example of document containing merged rows and merged columns is shown in the Figure 4b. In training set, 3 documents were designated as level 2 documents. While in test set, 13 documents were allocated to Complexity level 2.

Complexity level 3 is the most complicated level, as it contains more complex cases of merged rows and columns. The documents in this level has either both merged rows and column cases at a time or multiple cases of merged rows or columns, which makes it more complicated and tricky as compared to previous levels. An example of document containing such table is shown in Figure 4c. In training set, 3 documents were designated as level 3 documents. While in test set, 7 documents were alloted to Complexity level 3.



production. In terms of firm-level leverage, however, the sector is highly fragmented and localized, and thus a major corporate leverage point may be difficult to identify.

Feedlots

Feedlots satisfy three leverage criteria: direct control of manure management, direct control of cattle diets, and a highly concentrated market.

Once cattle reach an entry-level weight, about 650 pounds (300 kg), they are transferred to a feedlot and fed a specialized diet. During approximately four months on the feedlot the animal may gain an additional 400 pounds (180 kg). Cattle feedlots are concentrated in the Great Plains, mainly in Texas, Kansas, Nebraska and Colorado. Feedlots with fewer than 1,000 head of capacity comprise the vast majority, but these small and medium-sized operations market a relatively small share of fed cattle. In contrast, feedlots with 1,000 head or more of capacity (comprising less than 5 percent of total feedlots) account for 80-90 percent of fed cattle. Feedlots with 32,000 head or more of capacity market around 40 percent of fed cattle (USDA/ERS, 2008a).

Cattle feeding practices have been identified as a possible approach to reducing methane emissions from the beef industry. A brief summary of such methods appears in **Table 7**.

Table 7. Summary of Feed Strategies to Reduce Methane Emissions from Cattle Operations

Practice	Effect
High-grain diet	Reduce methane emissions, increase animal production efficiency
Less ruminal fermentation time for feed	Convert less carbon to methane
Use of feed additives (Ionophores)	Used in most fed cattle; inhibit formation of methane by rumen bacteria
Higher production efficiency	Reduce methane emissions by increasing productivity per animal
Use of feed other than forages	Reduce methane in animal

Source: CGGC, based on Ohhida, 2002.

The feeding strategies of most feedlots are focused primarily on using nutritional additives to increase feed efficiency, including hormones to promote growth. Estrogen, progesterone, and testosterone (three natural hormones), and zeranol and trenbolone acetate (two synthetic hormones) may be used as an implant on the animal's ear. While the main objective appears to be maximizing feed efficiency, some widely used additives have potential environmental effects, including the following:

- **Ionophores** increase feed efficiency and inhibit the formation of methane by rumen bacteria (U.S. Climate Technology Program, 2003).

24

(a) Table Complexity Level 1

16. BRAKE SYSTEM

SERVICE INFORMATION	16-1	REAR MASTER CYLINDER	16-12
TROUBLESHOOTING	16-2	FRONT BRAKE CALIPER	16-16
BRAKE FLUID REPLACEMENT/ AIR BLEEDING	16-3	REAR BRAKE CALIPER	16-19
BRAKE PAD/DISC	16-5	BRAKE PEDAL	16-22
FRONT MASTER CYLINDER	16-8		

SERVICE INFORMATION

GENERAL

Keep grease off of brake pads and disc.

A contaminated brake disc or pad reduces stopping power. Discard contaminated pads and clean a contaminated disc with a high quality brake degreasing agent.

- Never allow contaminates (dirt, water, etc.) to get into an open reservoir.
- Once the hydraulic system has been opened, or if the brake feels spongy, the system must be bleed.
- Always use fresh DOT 3 or 4 brake fluid from a sealed container when servicing the system. Do not mix different types of fluid they may not be compatible.

CAUTION:

Spilled brake fluid will severely damage instrument lenses and painted surfaces. It is also harmful to some rubber parts.
Be careful whenever you remove the reservoir cap; make sure the front reservoir is horizontal first.

- Always check brake operation before riding the motorcycle.

SPECIFICATIONS

Unit: mm (in)

ITEM		STANDARD	SERVICE LIMIT
Front	Specified brake fluid	DOT 3 or DOT 4	—
	Brake pad wear indicator	—	To groove
	Brake disc thickness	3.8 – 4.2 (0.15 – 0.17)	3.0 (0.12)
	Brake disc runout	—	0.1 (0.004)
	Master cylinder I.D.	12.700 – 12.743 (0.5000 – 0.5017)	12.755 (0.5022)
	Master piston O.D.	12.657 – 12.684 (0.4983 – 0.4994)	12.640 (0.4976)
	Caliper cylinder I.D.	25.400 – 25.405 (1.0000 – 1.0002)	25.450 (1.0020)
	Caliper piston O.D.	25.318 – 25.368 (0.9968 – 0.9987)	25.300 (0.9960)
Rear	Specified brake fluid	DOT 3 or DOT 4	—
	Brake pad wear indicator	—	To groove
	Brake disc thickness	3.8 – 4.2 (0.15 – 0.17)	3.0 (0.12)
	Brake disc runout	—	0.1 (0.004)
	Master cylinder I.D.	12.700 – 12.743 (0.5000 – 0.5017)	12.755 (0.5022)
	Master piston O.D.	12.657 – 12.684 (0.4983 – 0.4994)	12.645 (0.4978)
	Caliper cylinder I.D.	32.030 – 32.080 (1.2610 – 1.2630)	32.090 (1.2634)
	Caliper piston O.D.	31.948 – 31.998 (1.2578 – 1.2598)	31.94 (1.257)

16

http://mototh.com 16-1

(b) Table Complexity Level 2

MAINTENANCE

ITEM			SPECIFICATIONS
Engine oil capacity	At draining		1.0 liter (1.06 US qt, 0.88 Imp qt)
	At disassembly		1.3 liter (1.37 US qt, 1.14 Imp qt)
Recommended engine oil			Honda 4-stroke oil or equivalent motor oil API service classification SE, SF or SG Viscosity: SAE 10W-30
Engine idle speed			1,400 ± 100 min⁻¹ (rpm)
Valve clearance	IN		0.16 ± 0.03 mm (0.006 ± 0.001 in)
	EX		0.25 ± 0.03 mm (0.010 ± 0.001 in)
Drive chain slack			25 – 35 mm (1.0 – 1.4 in)
Drive chain size/link			DID428V3 – 124LE
Brake fluid			DOT3 or DOT4
Clutch lever free play			10 – 20 mm (3/8 – 13/16 in)
Tire size		Front	80/90-17M/C 44P
		Rear	100/80-17M/C 52P
Tire air pressure	Driver only	Front	200 kPa (2.00 kg/cm², 29 psi)
		Rear	200 kPa (2.00 kg/cm², 29 psi)
	Driver and passenger	Front	200 kPa (2.00 kg/cm², 29 psi)
		Rear	225 kPa (2.25 kg/cm², 33 psi)
Minimum tire tread depth		Front	To the indicator
		Rear	To the indicator

TORQUE VALUES

Oil drain bolt	25 N·m (2.5 kgf·m, 18 lbf·ft)
Spark plug	12 N·m (1.2 kgf·m, 9 lbf·ft)
Rear axle nut	59 N·m (6.0 kgf·m, 43 lbf·ft)
Crankshaft hole cap	7.9 N·m (0.8 kgf·m, 5.8 lbf·ft)
Timing hole cap	5.9 N·m (0.6 kgf·m, 4.3 lbf·ft)
Drive sprocket fixing plate bolt	10 N·m (1.0 kgf·m, 7 lbf·ft)
Driven sprocket nut	64 N·m (6.5 kgf·m, 47 lbf·ft) U-nut
Front master cylinder cover screw	1.5 N·m (0.15 kgf·m, 1.1 lbf·ft)
Rear brake reservoir cover screw	1.5 N·m (0.15 kgf·m, 1.1 lbf·ft)

3-2 http://mototh.com

(c) Table Complexity Level 3

Figure 4: Documents with different complexity level tables.

Störcode	Beschreibung	Mögliche Ursache	Kontrolle/ Behebung
e0	Fehler Vorlauf- oder Rücklaufsensor.	Kurzschluss des Vorlauf- oder Rücklaufsensors	Visuell die Verdrahtung und Anschluss der Sensoren kon- trollieren. Sitzen die Stecker richtig? Mit Multimeter Widerstand von Verdrahtung und Anschluss messen.
e0	Fehler Vorlauf- oder Rücklaufsensor.	Defekter oder nicht ord- nungsgemäß angeschlosse- ner Vorlauf- oder Rücklauf- temperatursensor	Funktion der Sensoren kontrollieren. Sensoren heraus- nehmen, mit Multimeter den Widerstand bei Raumtem- peratur (20 - 25 °C) messen; der Sensor ist in Ordnung, wenn Widerstand zwischen 12 - 15 k liegt , siehe Widerstandsgrafik.
e1	Vorlauftemperatur ist höher als die einge- stellte Höchsttempe- ratur.	Zu wenig Wasser	Mindestwasserdruck am Manometer kontrollieren.
e1	Vorlauftemperatur ist höher als die einge- stellte Höchsttempe- ratur.	Kein Wasserumlauf	Funktion der Pumpe kontrollieren, mit Schraubendreher Welle ggf. gangbar machen; falls diese gangig ist, die Pumpe jedoch noch nicht reagiert, Spannungsversorgung der Pumpe kontrollieren; sollte diese in Ordnung sein, so ist die Pumpe defekt.
e1	Vorlauftemperatur ist höher als die einge- stellte Höchsttempe- ratur.	Luft in der Anlage	Anlage entlüften.
e1	Vorlauftemperatur ist höher als die einge- stellte Höchsttempe- ratur.	Abweichung von Vorlauf- oder Rücklaufsensor	Funktion der Sensoren kontrollieren; Sensoren heraus- nehmen, mit Multimeter den Widerstand bei Raumtem- peratur (20 - 25 ° C) messen; der Sensor ist in Ordnung, wenn Widerstand zwischen 12 - 15 k liegt, siehe Wider- standsgrafik.

(a) Output from our system

Störcode	Beschreibung	Mögliche Ursache	Kontrolle/Behebung
e0	Fehler Vorlauf- oder Rücklaufsensor	Kurzschluss des Vorlauf- oder Rücklaufsensors	Visuell die Verdrahtung und Anschluss der Sensoren kon- trollieren. Sitzen die Stecker richtig? Mit Multimeter Widerstand von Verdrahtung und Anschluss messen. Funktion der Sensoren kontrollieren. Sensoren heraus- nehmen, mit Multimeter den Widerstand bei Raumtem- peratur (20 - 25 °C) messen; der Sensor ist in Ordnung, wenn Widerstand zwischen 12 - 15 k liegt , siehe Widerstandsgrafik.
e1	Vorlauftemperatur ist höher als die einge- stellte Höchsttempe- ratur.	Zu wenig Wasser	Mindestwasserdruck am Manometer kontrollieren.
		Kein Wasserumlauf	Funktion der Pumpe kontrollieren; mit Schraubendreher Welle ggf. gangbar machen; falls diese gangig ist, die Pumpe jedoch noch nicht reagiert, Spannungsversorgung der Pumpe kontrollieren; sollte diese in Ordnung sein, so ist die Pumpe defekt.
		Luft in der Anlage	Anlage entlüften.
		Abweichung von Vorlauf- oder Rücklaufsensor	Funktion der Sensoren kontrollieren; Sensoren heraus- nehmen, mit Multimeter den Widerstand bei Raumtem- peratur (20 - 25 °C) messen; der Sensor ist in Ordnung, wenn Widerstand zwischen 12 - 15 k liegt, siehe Wider- standsgrafik.

(b) Output from Adobe Acrobat Pro

Störcode	Beschreibung	Mögliche Ursache	Kontrolle/Behebung
		• Kurzschluss des Vorlauf-	• Visuell die Verdrahtung und Anschluss der Sensoren kon-
		oder Rücklaufsensors	trollieren. Sitzen die Stecker richtig?
			• Mit Multimeter Widerstand von Verdrahtung und
			Anschluss messen.
	Fehler Vorlauf- oder		
e0	Rücklaufsensor.	• Defekter oder nicht ord-	• Funktion der Sensoren kontrollieren. Sensoren heraus-
		nungsgemäß angeschlosse-	nehmen, mit Multimeter den Widerstand bei Raumtem-
		ner Vorlauf- oder Rücklauf-	peratur (20 - 25 °C) messen; der Sensor ist in Ordnung,
		temperatursensor	wenn Widerstand zwischen 12 - 15 kΩ liegt , siehe
			Widerstandsgrafik.
		• Zu wenig Wasser	• Mindestwasserdruck am Manometer kontrollieren.
		• Kein Wasserumlauf	• Funktion der Pumpe kontrollieren; mit Schraubendreher
			Welle ggf. gangbar machen; falls diese gangig ist, die
	Vorlauftemperatur ist		Pumpe jedoch noch nicht reagiert, Spannungsversorgung
	höher als die einge-		der Pumpe kontrollieren; sollte diese in Ordnung sein, so
e1	stellte Höchsttempe-	• Luft in der Anlage	ist die Pumpe defekt.
	ratur.	• Abweichung von Vorlauf-	• Anlage entlüften.
		oder Rücklaufsensor	• Funktion der Sensoren kontrollieren. Sensoren heraus-
			nehmen, mit Multimeter den Widerstand bei Raumtem-
			peratur (20 - 25 °C) messen; der Sensor ist in Ordnung,
			wenn Widerstand zwischen 12 - 15 kΩ liegt, siehe Wider-
			standsgrafik.

(c) Output from Tabula

Figure 5: Comparison with outputs from different tools.

4.1.1 Training Set

Training set consisted of total 10 documents distributed into 3 complexity levels. Training set along with ontology was used to define heuristics that represent the relevance. Table 1 shows training set distribution statistics.

Table 1: Training set Distribution.

Levels	Total no. of Tables	Relevant Tables
Level 1	195	19
Level 2	164	14
Level 3	97	25
Overall	456	58

4.1.2 Validation Set

Validation set consisted of total 38 documents distributed into 3 complexity levels. Validation set was used to evaluate the significance of earlier defined heuristics. Table 2 shows validation set distribution statistics.

Table 2: Validation Set Distribution.

Levels	Total no. of Tables	Relevant Tables
Level 1	301	12
Level 2	364	43
Level 3	42	4
Overall	707	59

4.1.3 Test Set

Testing set consisted of total 28 documents which were also divided into 3 levels based on their layout complexity level. Table 3 shows test set distribution statistics.

Table 3: Test Set Distribution.

Levels	Total no. of Tables	Relevant Tables
Level 1	310	28
Level 2	444	47
Level 3	116	18
Overall	870	93

4.2 Results

This section not only discusses results from the developed system but also from a couple of renowned tools around for solution of the problem stated in our use case.

In this section we will discuss results from evaluation of our developed system. Table 4 shows the results when test set was fed into our system. In Table 4, it can be observed that there exists no case where relevant tables are missed by the developed system. Such measure is represented by False -ve in the given tables. It depicts robustness of the developed system against the variation in terminologies used by different manufacturers.

It is quite evident from the statistics that as soon as layout complexity increases from one document level to the other, number of issues also increases. It is to be noted that the all results mentioned in this section are based on documents from 20 different manufacturers, with a lot of variation and no generalized layout format or set terminology followed in any of these documents.

Comparison with Renowned Tools

It is to be noted that existing tools for table extraction are not directly comparable with the proposed approach. This is because, they do not provide a feature of extracting relevant tables. Therefore, in the paper we provide a comparison with these tools, only on table extraction level.

For results comparison, we selected one tool with top performance in both open source and premium categories. Output from each system is compared with output of proposed system while providing same set of documents to each system.

Tabula is an open source tool freely available online for all types of usage. It specializes in extracting tables out of PDF documents. It provides two ways of extracting tables. One by automatic detection and other by manual selection.

Acrobat Pro is very famous product of Adobe family. There are several ways which Adobe Acrobat Pro provide for extracting data from PDF document. Acrobat extracts tables by exporting complete document in the form of an excel sheet. In this way all the content and tabular data will be exported to an excel sheet.

Comparison with other Tools

This section discusses the comparison of the system output with different state-of-the-art tools to witness the effectiveness of the output generated by our system.

Figure 5 shows the sample output from each of the systems, provided that a sample document containing a table with merged rows was fed to each of the systems respectively. Figure 5a shows the output of our system. It can be seen that all row and column data is extracted with absolute precision where there are crisp boundaries between all rows and columns. Additionally, the data in merged rows is duplicated carefully to the respective row cells. Figure 5b shows the output from Adobe Acrobat Pro. It can be seen that merged rows were not been detected correctly. But also the merged rows were considered as separate rows thus leaving the cells empty for the later row and resulting the gaps in the tabular data. Figure 5c shows the output from Tabula. It can be seen that neither the merged rows were detected correctly, nor the data in each row cell was considered as a single block. Each line was considered as a separate row thus leaving a lot of table cells empty because of misinterpretation of rows, columns and their respective cells.

From such performance of state-of-the-art tools, it can be inferred that it is indeed not so simple task to extract information from complex merged rows and columns. The proposed system overcame this problem and made it possible to extract quality wise reliable data from the tables.

5 CONCLUSION

This paper presents ontology based method for information extraction from technical documents. It serves as a tool for relevant table extraction from a PDF document. Relevancy is defined in the form of an ontology in the system. When this ontology is incor-

Table 4: Test Set Evaluation Results.

Levels	True +ve	False +ve	True -ve	False -ve	Precision	Recall	F-Measure
Level 1	28	0	282	0	1	1	1
Level 2	53	6	385	0	0.89	1	0.94
Level 3	26	8	82	0	0.76	1	0.86
Overall	107	14	749	0	0.88	1	0.93

porated with the system, it enables the system to be generic enough to use it for documents from any other domain. The presented system is totally autonomous and can process the documents without any human feedback. The presented system is able to produce output efficiently irrespective of the size of document. It is also very robust as it can process documents from a bunch of different brands with no standardization of terminologies or layouts. Reliability of output is represented in the report generated along the output files, where each table has separate confidence score with reasoning.

The presented system is implemented in such a way that it does not adhere to any specific use case, but can also work for any other domain documents with relevant data tables extraction problem. The presented system could be tested on any other domain documents by simply replacing the current ontology with the desired domain ontology.

REFERENCES

Adelfio, M. D. and Samet, H. (2013). Schema extraction for tabular data on the web. *Proc. VLDB Endow.*, 6(6):421–432.

Chang, C.-H., Kayed, M., Girgis, M. R., and Shaalan, K. F. (2006). A survey of web information extraction systems. *IEEE Trans. on Knowl. and Data Eng.*, 18(10):1411–1428.

Chao, H. and Fan, J. (2004). *Layout and Content Extraction for PDF Documents*, pages 213–224. Springer Berlin Heidelberg, Berlin, Heidelberg.

Freitag, D. (1998). Information Extraction from HTML: Application of a General Machine Learning Approach. In *AAAI/IAAI*, pages 517–523.

Gatterbauer, W. and Bohunsky, P. (2006). Table extraction using spatial reasoning on the css2 visual box model. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1313–1318. AAAI Press.

Liu, Y., Mitra, P., Giles, C. L., and Bai, K. (2006). Automatic extraction of table metadata from digital documents. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '06, pages 339–340, New York, NY, USA. ACM.

Milosevic, N., Gregson, C., Hernandez, R., and Nenadic, G. (2016). Extracting patient data from tables in clinical literature - case study on extraction of bmi, weight and number of patients. In *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2016)*, pages 223–228.

Peng, F. and McCallum, A. (2006). Information extraction from research papers using conditional random fields. *Inf. Process. Manage.*, 42(4):963–979.

Pinto, D., McCallum, A., Wei, X., and Croft, W. B. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 235–242, New York, NY, USA. ACM.

Rahman, A. F. R., Alam, H., and Hartono, R. (2001). Content extraction from html documents. In *Int. Workshop on Web Document Analysis (WDA)*, pages 7–10.

Ramakrishnan, C., Patnia, A., Hovy, E., and Burns, G. A. (2012). Layout-aware text extraction from full-text pdf of scientific articles. *Source Code for Biology and Medicine*, 7(1):7.

Rosenfeld, B., Feldman, R., and Aumann, Y. (2002). Structural extraction from visual layout of documents. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, pages 203–210, New York, NY, USA. ACM.

Ruffolo, M. and Oro, E. (2008). Xonto: An ontology-based system for semantic information extraction from pdf documents. *2008 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 01:118–125.

Tengli, A., Yang, Y., and Ma, N. L. (2004). Learning table extraction from examples. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wei, X., Croft, B., and Mccallum, A. (2006). Table extraction for answer retrieval. *Inf. Retr.*, 9(5):589–611.