TRANSPARENT DATA
DATA SOFTWARE HOUSE

Case Study Data Services: Parsing table data from PDF files

Introduction

A few words about the project

Our client, like many companies that collect business information from various, often "difficult" sources, faced the problem of converting tabular data from PDF files in a way that would actually be effective and reliable. The company’s previous tabular data parser did not perform well in practice, returning an error rate of 19.9%.

The PDF format is not friendly to machine processing - it is a combination of a binary and text file, which is far from structured data. In the case of plain text, parsing data from PDF files is often simply time-consuming. However, when documents made available in this format have the structure of tables that often stretch over several pages, with fields filled with text of unpredictable length, sometimes typos, data parsing becomes a real challenge. The number of parser errors affects the costs of obtaining, processing and further use of the information.

As an expert in the field of data processing, **we were hired to build a completely new parser for tabular data from PDF files for the client.**

Type of performed data services: parsing tabular data from PDF files
Year: 2021
Project duration: 1 month

Business needs

The client used a PDF parser extracting data from tables, and the data extracted in this way was later used in an internal web application supporting risk management and quick qualification of B2B leads in accordance with the adopted KYC/KYB procedure. The high level of errors in the parsing process translated into significant business process costs. Incomplete or misleading information generated costs not only at the level of manually correcting data and maintaining database hygiene, but also the price of wrong decisions based on them.

The client's need was therefore to significantly improve the quality of the parsing process. The company decided to cooperate with our data software house due to Transparent Data's extensive experience in technological processing and aggregation of huge sets of economic data.

What did the client expect from the new tabular data parser?

The development of **a new way of parsing data from PDF tables was supposed to minimize errors to a maximum of 4%.** The client counted on us to take care of the whole task comprehensively - from conducting data analysis and tracking down errors and their qualification to writing a new parser whose task is to convert data in PDF format into data in JSON format according to a specific scheme. We also undertook to help the client in implementing a new solution in the company's internal application.

Solution

Key benefits of a new way of parsing table data from PDF files

- 63 x fewer total errors when parsing data (number of errors is only 0.31% on average)
- Complete elimination of such repetitive errors as typos or poorly described PESEL numbers
- The average time to parse a single PDF document is 7 times shorter

What work has Transparent Data done?

The first stage was an **audit of the client's old PDF parser methodology and analysis of data in PDFs in order to catch all possible types of errors**. In this way, we have located that 78% of all errors are digits in names and incorrectly described PESEL identification numbers. The second large pool of errors resulted from the length of the data stored in the tables and the truncation of data contained in subsequent lines.

One of the reasons for these problems was the misguided approach of using the popular pdftotext tool to convert PDF files to text files. The original parser of tabular data from PDF was based on reading information between lines, i.e. on numerous text matches through regular expressions, a kind of “anchors”. The occurrence of these "anchors" was expected and sought after. The found fragment of text between the two "anchors" was then retrieved and saved. This approach gave rise to many problems, in particular in the case of inevitable typos in the analyzed text or the unusual layout of the generated table. In these situations, regular expressions were commonly returning an empty match or a match containing redundant and unwanted characters. Ultimately, such distorted or incomplete data often went directly to the client's business application, reducing its credibility and utility value.

We have created a new data parser from tables from PDF files in a technology that enables a more **analytical approach to the problem to be solved and parallel processing** (i.e. at the same time) of each page of the document. Instead of relying on regular expressions with many conditional statements, the new solution allows the client to provide a configuration scheme (so-called JSON Schema) to the parsed file. It fixes the position of each rectangle and each character inscribed in it. It combines characters into sentences and saves them in a cell, combines cells into rows, rows into tables and, if necessary, merges tables that are sometimes separated into several pages.

With a new approach, the PDF parser analyzes tables by predetermined schemas by detecting the number of rows, columns, and nested subtables. As a result of this action, it produces a structured form of table content in the entire analyzed file. Importantly, the JSON schema is easy to create even for non-technical people and we conducted workshops with the client, during which we jointly created schemas tailored to the needs of its processes.

In addition to the additional performance gain, the new data parsing solution is more versatile, generic and configurable, and easier to adapt to other places that work with tabular values in PDF. Accepting a certain margin of error (the so-called Levenshtein distance), text processing algorithms allow the correct matching of tables based on their headers, even if they contain minor errors or typos.

The biggest challenges of the project

- The biggest challenge was the extensive analysis of the data contained in the PDF files to create a new typology of 60 different, repeatable data patterns. The original documents showed:
    - No internal structure on which data parsing mechanisms can be based (PDF is by definition a format used to present text and graphic content - it is not a good way to exchange data between IT systems)
    - No rules for repeated breaking of pages and text in tables
    - No rules for the filling in an individual cells (e.g. missing values in a row or other than expected data type);
- An additional complication was the fact that the tables in the PDF files were automatically generated from unknown software, which in some cases gave rise to unusual and unforeseen minor visual errors. For example, when the table did not fit on one page, it was stretched over many more pages of the document. These visual imperfections were directly translated into the text matching process, lowering the quality of the data provided. At the same time, being a dynamically generated table, it did not have a homogeneous structure. Both the number of table rows and the number of nested subtables were variable and dynamic depending on the file provided;
- In addition to the change in the way data is parsed, the project entailed the creation of a completely new database, including the development of a new data structure and integration of the client's old systems with the new database. This required comprehensive actions in many areas.

Technologies we used

- Go (Golang)
- PHP 8.0/8.1
- RabbitMQ
- Redis
- Laravel (Lumen)

- Kubernetes
- Docker
- Google Cloud Platform