Building a Product Taxonomy with Information Extraction (IE) techniques: User Manual for Barentz

Veronika Cherkasova,
Oscar Cheng,
Katherine Tu
BSc Business Analytics, University of Amsterdam

Barentz Contact: Harm Bodewes UvA Supervisor: Dr. Jeroen De Mast Date: 19/06/2024

1. Introduction

1.1 Structure of this Manual

In the introduction section, we will go over the main project objectives, as well as the selected attributes and data model. In Section 2, each of the algorithms will be described and links are provided for the GitHub pages within each relevant subsection. The code contained in these GitHub pages also contains the necessary code to make the final data frames compatible with SQL lite. In Section 3, results are discussed and an evaluation is presented—finally, Section 4 details future recommendations. Additionally, an Excel document is shared separately from this manual. Contacts of the students can be found in the Appendix.

1.2 Project Objectives

The purpose of this thesis project was to convert unstructured product data to structured. To achieve this, two project objectives were defined by the students and corroborated by Barentz. The first objective was to define a data model, specifically the structure of the data model and the attributes included. The second was to create and evaluate various algorithms, which would transfer the data into the data model through information extraction techniques.

1.3 Attribute List and Data Model

1	Product Name	The list was made based on only available information from PDFs for both Roquette and Cargill and consideration for the needs of Barentz. There were many more attributes to consider, but in the context and nature of this assignment, we chose these 9 key attributes which were confirmed with Barentz. The Chosen Model: **Relational Table** - Both a hierarchical and relational model were considered. - Relational was preferred as it is intuitive to use, can be easily changed later if necessary, and most time-efficient to create.
2	Market Segment	
3	CAS-NO	
4	EC NO	
5	Chemical	
6	Supplier ID Description	
7	UN Code	
8	ADR	
9	Hazard	

Table 1. Attribute list.

2. Information Extraction (IE) Technique Algorithms

2.1 Pure Regular Expression

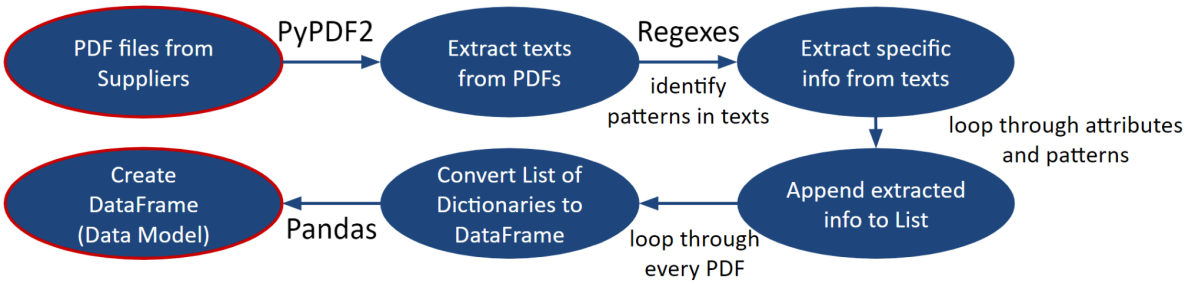

Figure 1. Streamline of pure regular expression algorithm.

The pure regular expression is a ruled-based approach. The Pure Regular Expression approach in both Python notebooks could be considered as follows (see Figure 1):

a. Loading the input data: a large collection of PDF documents with unsystematic product descriptions and no uniformity in layout and naming conventions.
b. Extracting the complete PDF file information to plain text using PyPDF2.
c. Applying the regular expression algorithm to identify specific patterns for information we are looking for, namely, those 9 mandatory attributes (see Table 1). Extract the particular information from the plain text and eventually store the extracted information in the structured $N \times M$ relational data model.

The advantages of using pure regular expressions included high accuracy, short runtime, and ease of understanding. Disadvantages included low scalability, meaning that it requires constant updates for new formats and/or new suppliers to include new patterns as well as the limited flexibility for documents with different languages.

Regex method GitHub links:

- Cargill notebook: https://github.com/KatherineTu24/Bachelor-s_Thesis_Barentz/blob/4c7dee2a415a9424c39d575d1a2965acbe08d103/Cargill_final_ver.ipynb
- Roquette notebook: https://github.com/KatherineTu24/Bachelor-s_Thesis_Barentz/blob/4c7dee2a415a9424c39d575d1a2965acbe08d103/Roquette_final_ver.ipynb

2.2 Custom Entity Recognition

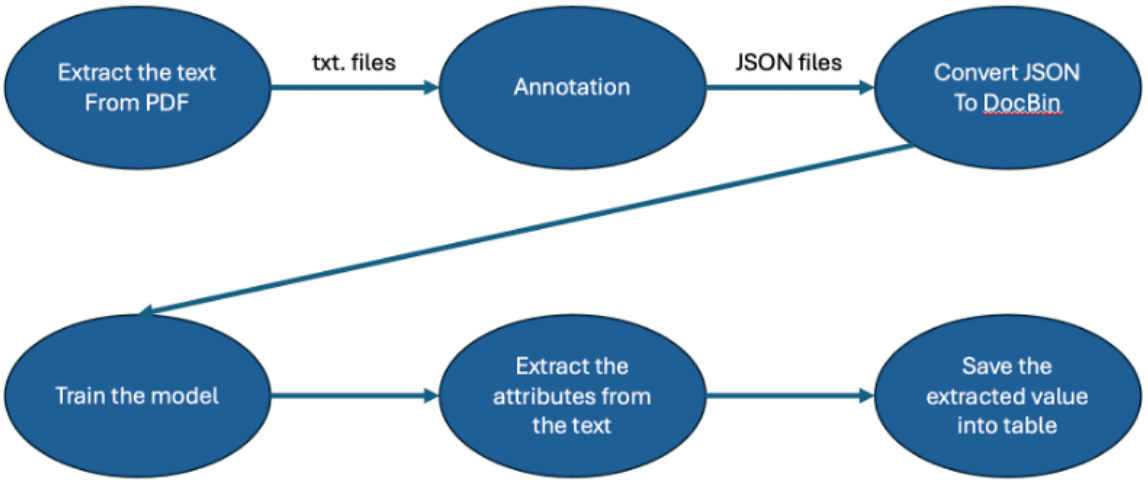

Figure 2. Streamline of the custom entity recognition algorithm.

a. Load the training data and extract the text from the pdf files with PDFReader. (Save it as txt. files)
b. Annotate the training text with NER Annotator for Spacy (https://tecoholic.github.io/ner-annotator/)
c. Go to https://spacy.io/usage/training to download the base_config file.

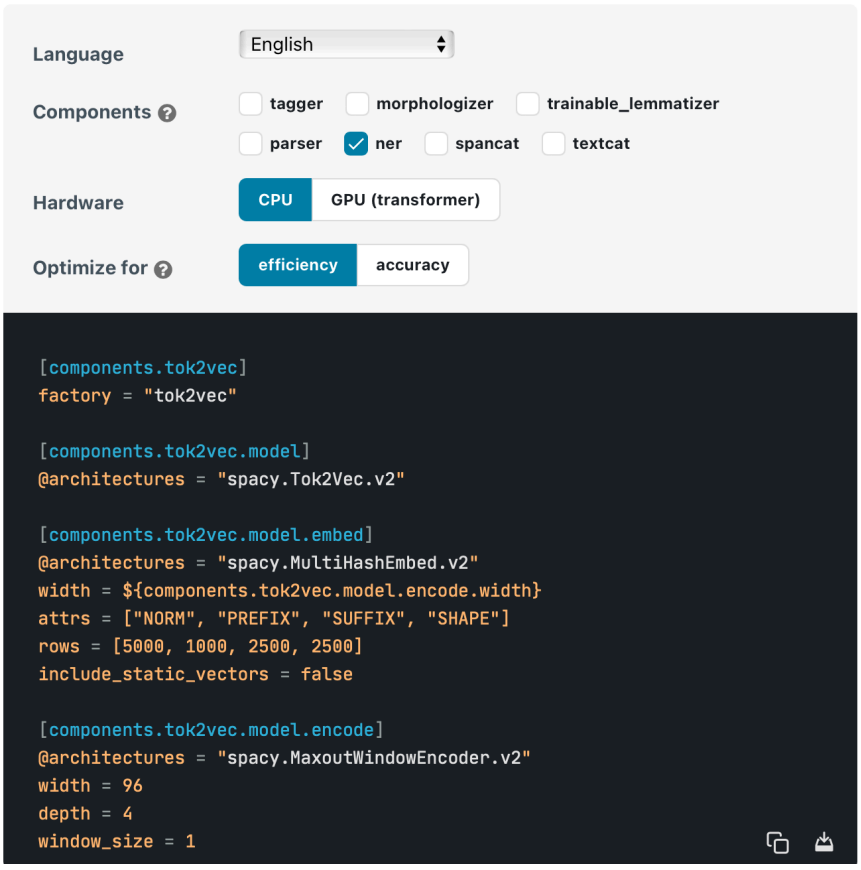

Figure 3.

d. Apply the custom entity recognition model to extract the attributes.

The advantage of using a custom entity recognition model is its flexibility and scalability. Once the model is trained well, the model can deal with most data from the suppliers if the documents from that supplier are in the training sets. However, the biggest drawback of this model is the tradeoff between speed and performance. To have a well-performing model, the training takes a long time. On the other hand, training the model with less documents will lead to a worse performance.

Custom entity recognition model Github link: https://github.com/ChengOscar/Thesis

2.3 Rule-based if loop + Regular Expression

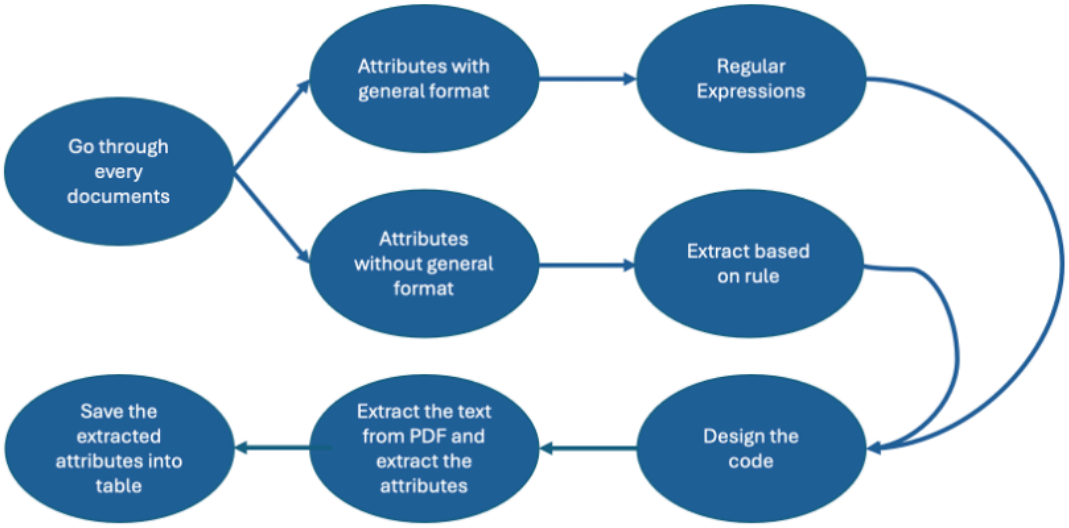

Figure 4. Streamline of the rule-based if loop + regular expression algorithm.

a. Extract the text from PDF documents with PDFReader
b. Separate attributes into two categories- with or without uniform format
c. Use regular expression to extract attributes with uniform format
d. Go through the whole document to find the rule of attributes without uniform format
e. Design the code based on pattern or rule
f. Directly apply the rule-based model to extract the attributes
g. Save the data into SQL or Excel table

The advantage of the rule-based method is that this model is less complex compared to the custom entity recognition model. However, the model is less flexible and it might not work for documents from every supplier since the model is designed specifically for Cargill and Roquette.
Rule-based if loop + Regular Expression model Github link:

https://github.com/ChengOscar/Thesis

2.4 Drag and Drop Zonal OCR

The code for all the methods from 2.4, 2.5 and 2.6 can be found through the link at the end of subsection 2.6.

First, the PDF is converted into an image. Next, the user has to select an area from which the information will be extracted using the Tesseract package. Figure 5 shows an illustrative example of how the algorithm works with an added grid and 'Name' to make the algorithm simpler to understand. Figure 6 displays how it looks for the user who is selecting the boxes.

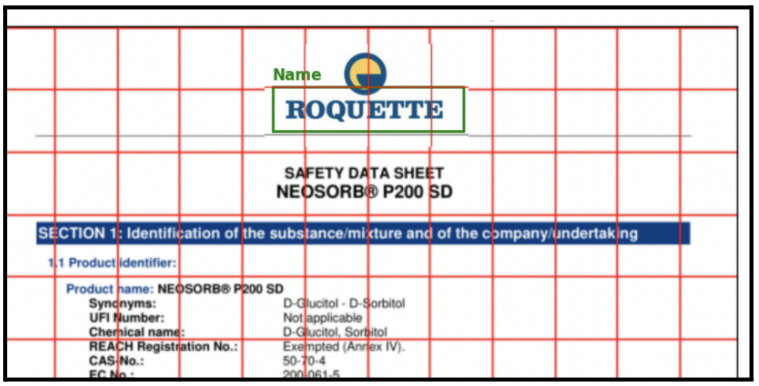

Figure 5. Example of how selection works, with added grid and 'Name'.

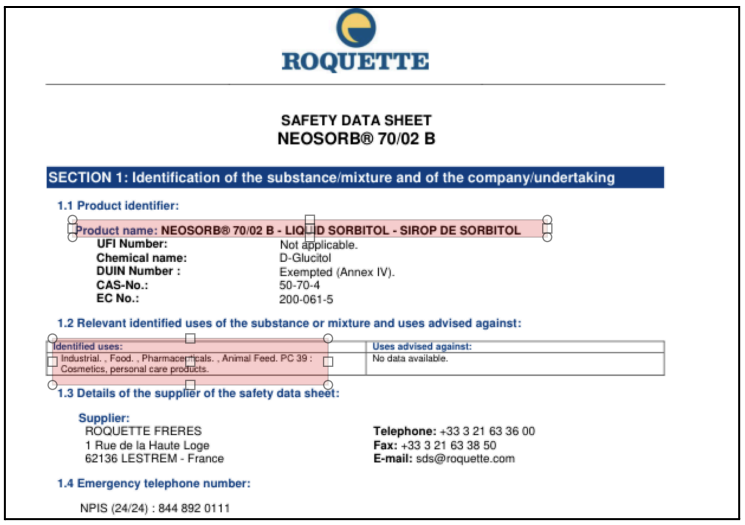

Figure 6. Display for the user selecting the boxes.

After the user 'drags and drops' their pointer, creating a rectangle shape, this image is saved and the coordinates are extracted from it. Finally, from these coordinates, a data frame (Figure 7) is created and code is provided to make it compatible with SQL.

	Product Name	CAS-No.	EC No.	Chemical Name	Identified Uses	Supplier	Hazard?	ADR	UN Number
0	ANFOMUL™ 2000-LQ-(RB)	108-31-6	203-571-6	Distillates (petroleum),solvent-, dewaxed heav...	Raw material	Croda Inc.	No	None	None

Figure 7. The first line of the output data frame.

2.5 Template-based Zonal OCR

The difference between this algorithm and the one in Section 2.4 is that the coordinates are predefined by the algorithm. Hence the following steps are applied:

a. First some additional packages are installed, most importantly Tesseract.
b. Then the template is created by defining the coordinates of the corners of the rectangles, hence creating the 'zones'.
c. The user uploads the PDF documents they are interested in and these PDFs are converted into images.
d. The 'template' is applied. This means using the predefined coordinates to create the rectangles or 'zones' from which the information is extracted. Figure 8 shows an example of how the output from this step could look like.

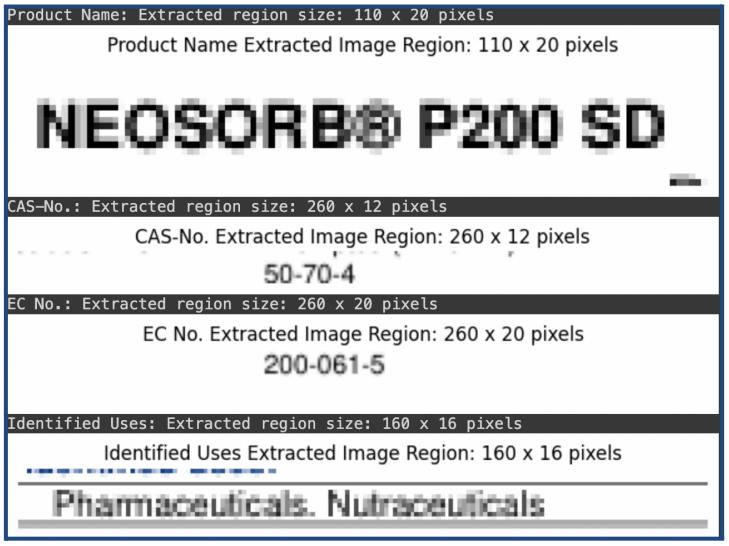

Figure 8. Example of extracted 'zone' images.

e. Then, the textual information is extracted from the images using the Tesseract package.

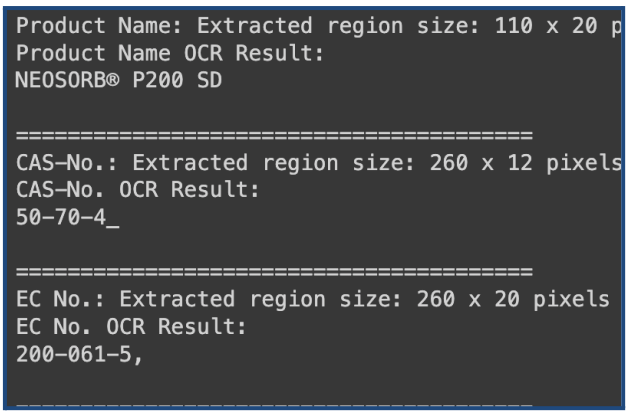

Figure 9. Conversion of the images from Figure 6 into text.

f. Finally, the data frame is created and code is provided to convert it into an SQL compatible format.

2.6 OCR + Regular Expressions

This method relies on the PyMuPDF library and the regular expressions library. In Figure 10 the key steps are outlined.

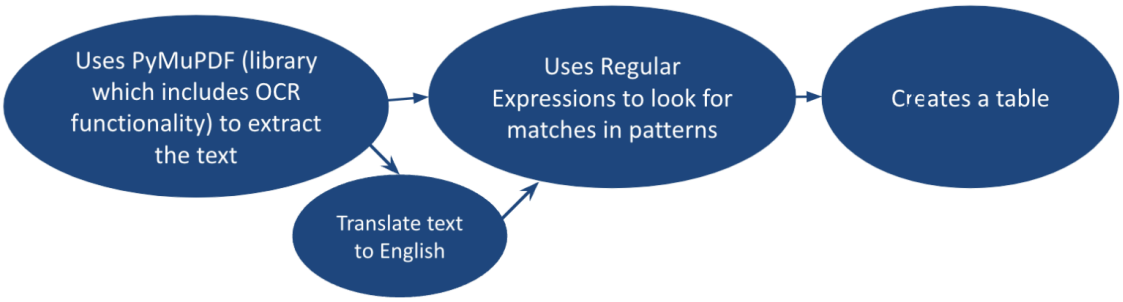

Figure 10. Conversion of the images from Figure 6 into text.

The code includes the following six phases:

a. First, all of the packages are installed
b. Then, the user has to upload the PDF files that they choose. The PyMuPDF library then extracts all the text from the given PDFs.
c. The third phase is optional. It includes the translation of a file from any given language into English.
d. In the fourth step, all the regular expression functions are defined. An example of such a function can be seen in Figure 11. These are custom-built for Roquette and Cargill PDFs, however, should be able to work with other PDFs also, although with a possible lower degree of accuracy

```
def UN(text):
    pattern = r'\bUN\s\d{4}\b'
    match = re.search(pattern, text)
    if match:
        return match.group(0)
    return None
```

Figure 11. Example of a Regular Expression function.

e. The previously defined functions are then applied to the PDFs that the user uploaded and a data frame is created. Refer to Figure 7.
f. Finally, the code is given which created a SQL-compatible data structure

Link to GitHub with full code for sections 2.4, 2.5, and 2.6:
https://github.com/Veronikastudentuva/Thesis-Code.git

3. Evaluation

3.1 Technical Metrics – Quantitative Method

Algorithms were evaluated in two perspectives: technical and fit-to-purpose aspects where each consists of three components. In technical metrics, accuracy represents the percentage of correct extractions out of the total possible extractions. Consistency is measured with the standard deviation to indicate the stability of algorithms. Lately, speed indicates the processing time per document for each algorithm.

Except for the "template-based Zonal OCR" method, the other algorithms performed quite well in the categories of accuracy and consistency. Five out of six algorithms achieved more than 85% accuracy rate. Specifically, "drag and drop Zonal OCR" reached almost 97% accuracy, the most among all six algorithms. Their standard deviations were all in the acceptable range $SD = [-2, 2]$. As for the speed, with the limited CPU, custom entity recognition required 40-plus minutes just to train on the dataset. The other algorithms completed the extraction process in less than two minutes for the given 60 PDFs from Cargill and Roquette.

Algorithms	Accuracy	Consistency	Speed
Custom entity recognition	85%	1.09	40+min
Rule-based if loop + Regular Expression	92%	1.14	70 seconds
Pure Regular Expression	94.1%	0.6	18 seconds
Drag and Drop Zonal OCR	96.9%	0.57	110 seconds
Template based Zonal OCR	38.9%	2.48	7 seconds
OCR + regular expressions	91.4%	0.96	5 seconds (no translation) 14 seconds (translated)

Table 2. Technical metrics comparisons between all 6 algorithms.

3.2 Fit-to-Purpose Metrics – Qualitative Method

Complexity, scalability, and flexibility are the three components of fit-to-purpose evaluation metrics. Specifically, complexity indicates ease of use and understanding, scalability demonstrates the feasibility of scaling up when adding new formats, attributes, and suppliers, and flexibility represents the tolerance of algorithms when dealing with multi-language documents. Since it is a qualitative approach to measuring algorithms, we scaled each component from 1 to 5, 1 representing the least complex (i.e. very easy to use)/very scalable/flexible and 5 representing the most complex/very unscalable/inflexible. To emphasise the importance of different components, each qualitative evaluation component is weighted accordingly where complexity weights 20%, and both scalability and flexibility weights 40%, respectively. Essentially, the lowest scores the better.

After calculating the weighted scores, the custom entity recognition approach stands out from the others with a total score of 1.6, while “template Zonal OCR” comes up in second place. The least preferred approach is the “rule-based if loop with regular expression” with a score of 4.6.

Algorithms	Complexity 20%	Scalability 40%	Flexibility 40%	Weighted Scores
Custom entity recognition	4	1	1	1.6
Rule-based if loop and regex	5	5	4	4.6
Pure Regular Expression	1	5	2	3
OCR + Regular Expressions	1	4	3	3
Template based Zonal OCR	1	2	4	2.6
Drag and Drop Zonal OCR	3	4	2	3

Table 3. Fit-to-purpose metrics comparisons between all 6 algorithms.

3.3 Overall Rankings

Combining both technical and fit-to-purpose evaluation approaches, we created a table of overall ranking for all six information extraction methods. According to the overall ranking, the pure regular expression is the most preferable algorithm, followed by the Zonal OCR + regular expression algorithm. The least preferable algorithm is the drag-and-drop Zonal OCR. However, it is worth mentioning that the final ranking is not the single source of truth in which the selection of the algorithms can differ from its purposes.

Algorithms	Technical ranking	Fit-to-Purpose ranking	Overall ranking
Pure Regular Expression	1	4	1
Zonal OCR + Regular Expressions	2	3	2
Custom entity recognition	5	1	3
Template based Zonal OCR	6	2	4
Rule-based if loop + Regular Expression	3	6	5
Drag and Drop Zonal OCR	4	5	6

Table 4. Overall rankings for all 6 algorithms.

4. Future Recommendations

Custom entity recognition:

- To have a more generalised custom entity recognition model, more training data or data from other suppliers is required.
- A better device can reduce the training time of the custom entity recognition model.

Regular Expression

- The strength can be optimised and drawback can be minimised when integrated with other natural language processing techniques, such as Bag of Words (BoW), NLTK (a language detection technique)

Zonal OCR

- A machine learning model can be trained on the templates to make the OCR model more versatile. However, the training is a time-consuming task.

Currently, only 9 attributes are extracted. Future research should focus on extracting all attributes as well as integrating more scalable AI techniques to handle diverse data sources and languages.

Appendix

Contacts:

Katherine Tu, linkedin.com/in/katherine-tu
Oscar Cheng, linkedin.com/in/shiu-che-cheng
Veronika Cherkasova, linkedin.com/in/veronika-cherkasova-vc