





Automatically Extracting Insurance Contract Knowledge Using NLP

Alexandre Goossens^(✉) , Laure Berth, Emilia Decoene,
Ziboud Van Veldhoven, and Jan Vanthienen 

Katholieke Universiteit Leuven, Naamsestraat 69, Leuven, Belgium
alexandre.goossens@kuleuven.be

Abstract. Vanbreda Risk & Benefits, a large Belgian insurance broker and risk consultant, allocates a substantial amount of time and resources to answer contract related questions from customers. This requires employees to manually search the relevant parameters in the contracts. In this paper, a solution is proposed and evaluated that automatically extracts insurance parameters from contracts using regular expressions and Natural Language Processing. While Natural Language Processing has been used in insurance for optimising premiums, detecting fraudulent claims, or underwriting, limited work has been done regarding parameter extraction. The proposed solution has been developed on 127 different contracts and two different contract types in terms of accuracy and time performance. Moreover, the automatic parameter extraction has been compared to manual parameter extraction. We conclude that automatic parameter extraction using regular expressions achieves better accuracy than manual extraction on top of being significantly faster, allowing Vanbreda Risk & Benefits to invest more time into providing better customer service.

Keywords: Service automation · Insurance industry · NLP · Regular expressions

1 Introduction

The development and improvement of text mining algorithms have given rise to a number of new applications. More specifically in the insurance industry, natural language processing (NLP) is used nowadays for example to fine-tune premiums [1], detect fraudulent claims [2], or perform sentiment analysis on tweets [3]. These applications deal with optimising profit for insurance companies. Insurance companies also need to provide information to customers related to their products. Customers need to know under which conditions they are insured or the rate at which they are covered in case something happens. These questions occur without necessarily having an insurance claim filed by the customers.

This paper will deal with the automatic extraction of insurance parameters from contracts using regular expressions and has been implemented at Vanbreda Risk &

Benefits¹ a large Belgian insurance broker and risk consultant. A solution based on regular expressions is proposed and formally evaluated on a test set of 42 real contracts in terms of accuracy and time-performance compared to manual work.

We proceed as followed: Sect. 2 provides background information concerning social security in Belgium and the company where the solution has been implemented. Section 3 states the problem statement. In Sect. 4, related work is discussed and in Sect. 5 the methodology is explained. The results are presented in Sect. 6 and the evaluation and future work are presented in Sect. 7. We conclude the paper in Sect. 8.

2 Context

2.1 Social Security in Belgium

In Belgium, the social security provisions consist of three parts or pillars. Each one of these pillars is written down in forms, regulations, or contracts describing under which conditions a person is eligible for provisions. Due to their complex nature, these forms or contracts are long and not straightforward to extract all relevant information by a non-expert. The three pillars of Belgian social security are: statutory provisions such as child benefits, illness benefits, retirement, unemployment; supplementary provisions provided by the employer or also known as employee benefits [4] and mainly exists because the first pillar alone is not able to guarantee living standards; individual provisions allowing individuals to build up their provisions on their own.

2.2 Current Situation at a European Insurance Broker Vanbreda Risk and Benefits

In the subsequent paragraphs, the current parameter extraction process at Vanbreda Risk & Benefits will be explained. Vanbreda Risk & Benefits is an independent insurance broker and risk consultant mainly operating within Belgium. Vanbreda Risk & Benefits has a department called Employee Benefits providing assistance in analysing and managing employee insurances of other companies. Due to the complex nature of these contracts, expertise knowledge is required to interpret these contracts. Whenever a client employee or customer company requires information (simple or complex), they can contact Employee Benefits to ask their questions. These questions require the employees to analyse the contracts of the customer. Vanbreda Risk & Benefits currently has approximately 2000 contracts of 25 pages long on average without counting the attachments.

Recently, a technical summary of a contract has been introduced in the company. This technical summary summarises the most relevant parameters of a contract such as a formula, a definition, or a date. A technical summary consists of a parameter column and a value column.

The employees of Employee Benefits can use the data of these technical summaries to answer most of the queries. Thanks to this technical summary, the question-answering

¹ <https://www.vanbreda.be/en/>.

process is sped up. Whenever a new contract is received, a technical summary is created. The creation of a technical summary requires between 15 and 20 min for a single contract. Since these technical summaries have been introduced recently, not all contracts have a technical summary yet. The extraction of parameters from contracts has always been manual work at Vanbreda Risk & Benefits whether the parameters were extracted directly from the contracts or more recently with the introduction of technical summaries.

3 Problem Statement: From Contract to Automatic Technical Summary

As stated previously, the creation of technical summaries is currently time-consuming. The employee queries the contract in the correct database and manually extracts the required parameters. In case the parameter has an assigned list of default values, one of them is chosen. If not, the parameter values do not have a standard form. This lack of standardisation means that a technical summary also depends of the employee creating it and therefore some variation exists between technical summaries. Moreover, contracts are subject to changes due to changing interest rates, changing market conditions, preferences and such.

It would be highly beneficial for insurance companies to have a digital solution to (partially) automate contracts into technical summaries. Hence, this research is interested in a scalable and applicable solution for different types of contracts (in this case: guaranteed income and waiver of premiums) a company is dealing with. Moreover, the proposed solution needs to be independent of the insurance provider. In short, the solution has to reduce the creation time of a technical summary and capable of handling different contract types and different insurance providers.

4 Related Work

4.1 Applications of NLP and AI in the Insurance Industry

The impact and applications of AI in data-intensive domains such as finance, insurance, and public services is a widely studied domain, e.g. [5–8] For the insurance industry, the authors of [7] say that robots and people can fully create benefits when people focus on building customer relationships and robots perform the repetitive tasks. This principle can indeed be seen as one of the main drivers for research in this direction by insurance companies.

There are numerous potential NLP applications in the insurance industry [9]. One application is the cost predictions of insurance claims using text mining on the injury and incident descriptions in conjunction with more structured data such as demographics [10]. NLP can also be used to fine-tune premiums [1] or to detect fraudulent claims [2]. Furthermore, these techniques can be used to gain customer insights for example by analysing customer calls [11].

With the rise of social media platforms, insurance companies have much more data at hand. In [3], tweets are analysed revealing the most common topics and their feelings towards an insurance company. This provides the insurance company the ability to provide better customer service and to reach potential new customers. The use of chatbots in the insurance industry has also been investigated in [12]. The so-called Intellibot is able to answer specific questions dealing with insurance provided it is given the correct insurance knowledge.

In short, current NLP applications mainly deal with fine-tuning premiums, detecting fraudulent claims, sentiment analysis or customer interaction, and so forth. However, limited academical attention has been paid to automatically extracting insurance parameters from contracts using NLP.

4.2 Document Segmentation

Within the widely studied field of NLP, information extraction is a possible application [13]. The goal of information extraction is to extract structured information from (semi)-unstructured documents.

In most organisations, legal documents in a digital format are often available. Unfortunately, this is often in a semi-structured form. Even though humans can interpret such documents easily, it remains a challenge for machines. This limitation inhibits the performance of information extraction [14].

Table 1. Overview of segmentation techniques.

Contract Segmentation Techniques			
Rules-based approach	Paragraph boundary detection	Conditional random fields	Generic named entity recognition
Search start of new paragraph by set of rules	Search paragraph boundaries by sentence detection	Classifying words by looking at the features the previous and next word (sequential classifier)	Identifying and recognizing specific entities in a text (names of persons, percentages, indications of time, etc.)

Some research has already been done to segment documents. The authors of [14] propose to segment documents into smaller parts to process legal documents easier hence allowing the program to have knowledge of the contract structure. These smaller parts could be sentences, paragraphs, or pages. The authors of [15] provide an overview of segmentation techniques. These are shortly summarised in Table 1. Since contracts are legal documents, knowing the structure of a contract can help in extracting contract elements [16].

5 Methodology

To start the automatic summarisation process, the contract must be read in. Every contract is received in a format called smart PDF. To read these contracts, the Python package PyMuPDF [17] or PDFMiner [18] allows for the text extraction from the PDF contract, next the text is stored as a string variable allowing us to perform the three necessary processing operations. Firstly, the contract is segmented, next the relevant parameters are searched and extracted, to finally be filled into the technical summary. The segmentation and extraction methods can both be used for other contract types. The parameter extraction needs to be fine-tuned to fit the correct purpose.

5.1 Segmentation

In general, a contract is segmented into two parts. The general terms and the special terms. The special terms deal with the insurance conditions and premiums and such. Every parameter that needs to be extracted is located in the special terms of the contract. Prior to the extraction, these special terms are divided into different parts allowing for the extraction to be more efficient. Dividing a contract into smaller parts allows for a more targeted search of a parameter instead of going through the full contract. In short, this means that the contract will be segmented into paragraphs. An overview of the segmentation steps is given in Fig. 1.

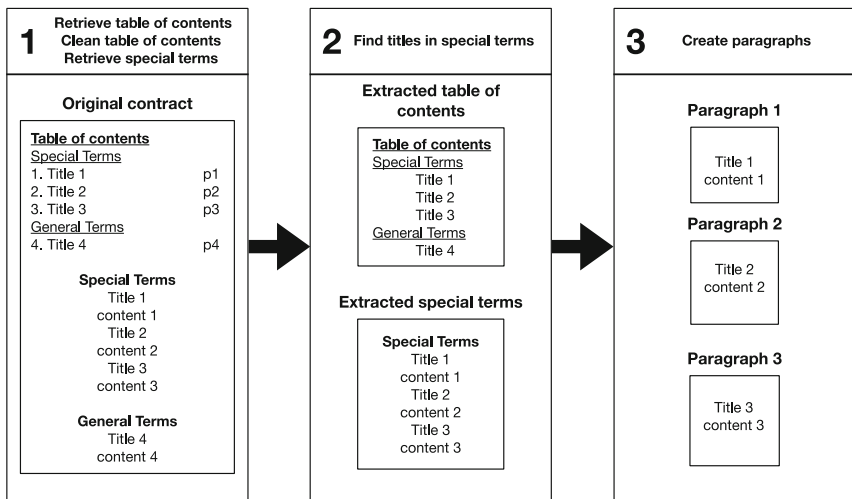


Fig. 1. Overview of the segmentation process.

Retrieve the Table of Contents: A contract structure remains more or less the same for all contract types and insurance providers for Vanbreda Risk & Benefits. In our case, the table of contents itself will be located between the words ‘table of contents’ and the very first title that reoccurs after the table of contents.

These exact words are looked up in the contract string by using regular expressions. The authors of [19] define a regular expression or regex as “a sequence of characters (letters, numbers and special characters) that form a pattern that can be used to search text to see if that text contains sequences of characters that match the pattern (p.257)”. For each matching pattern, the start indices are saved resulting in a substring that only consists of the table of contents.

Clean up the Table of Contents: Only the titles are needed in our solution but a table of contents also contains the page number, title number and so forth. Therefore, the table of contents needs to be cleaned. Each line of the table of contents is placed in a list and considered as a different element. For each list element, the title number, page number, and other punctuation marks are removed, leaving only the title in the list. At the end of this step, only a list of contract titles remains. These steps are also visualised in Fig. 1.

Retrieve the Special Terms: The same procedure as for the table of contents is used to retrieve the special terms. The special terms are located between “special terms” and “general terms” and these words are used as patterns to further search the contract. A separate substring is created that contains these special terms.

Find Titles in Special Terms: The extracted titles are put into regular expressions. These exact titles or patterns are then searched for in the complete special terms substring. Whenever a match is found between the pattern (the title) and a part of the substring (the special terms), the start index of the title is saved in a variable. This process is repeated until all start indices of titles in the contract are stored in a list.

Create Paragraphs: Finally, a paragraph dictionary is created. For each match found in the previous step a dictionary entry is created. The title extracted from the table of contents is a dictionary key and the belonging value is the according paragraph. Each paragraph is delimited by the start index of the paragraph title and by the start index of the next paragraph title.

5.2 Parameter Extraction

The parameter extraction process is visualised in Fig. 2. This part needs to be finetuned for each parameter.

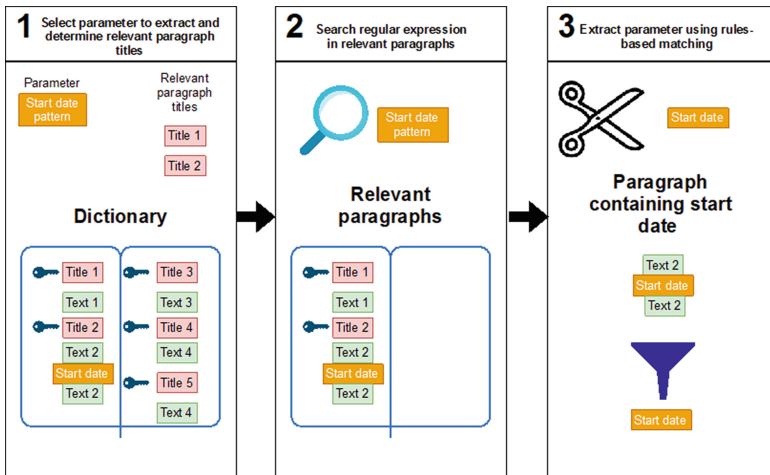


Fig. 2. Parameter extraction process.

Select Parameter to Extract: Determine which parameters need to be extracted for the creation of a technical summary.

Determine Relevant Paragraphs: In this step, the program needs to know in advance which paragraphs might contain the parameter. The most common relevant paragraph titles are given to the parameter extraction function. To increase the robustness, the title of the previous paragraph (according to the table of contents) is given in case there has been a spelling mistake. This is not a problem as for each insurance type and insurance provider the contract follows the same scheme. Once the pattern is determined it is scalable to all other contracts of the same type and insurance provider.

Extract Relevant Paragraphs: The required paragraphs have already been extracted following the steps explained in Sect. 5.1 and have been stored in a string variable.

Transform Text: In this step, the natural language toolkit (NLTK) is required [20]. The NLTK library allows us to preprocess a string to perform further analysis. The extracted paragraphs are parsed through a so-called sentence tokenizer. This will transform a given text into a list of sentences. In this case study, the Punkt sentence tokenizer has been chosen and as the contracts are in Dutch, the Dutch sentence tokenizer is used to transform the text.

Rule-Based Matching: To retrieve the parameters, rule-based matching is used. The idea of rule-based matching is to use regular expressions and to return a value matching the pattern in a given text. Suppose the desired parameter value is the start date of a contract. A date can either be written as DD/MM/YYYY or as DD-MM-YYYY. The following regular expression is able to detect both of these formats: $\backslashd\{1,2\}[\backslash-]\d\{1,2\}[\backslash-]\d\{4\}$. This process of rule-based matching is repeated until all desired parameters have been extracted. To finalise, a pandas dataframe [21, 22] is constructed. This

dataframe consists of two columns. One column is the parameter name and the second column contains the parameter value.

Sentence Retrieval: In some situations, the complete sentence is required and not just one parameter. In that case, the same procedure as explained above can be used except that when the patterns find a match, the whole sentence is returned instead of only the matching pattern.

5.3 Attachment Extraction

As is often the case with contracts, changes can be made to what was previously agreed upon. These changes are put in attachments and range from new premium rates to new clauses. In this part, the goal is to provide a summary of each attachment by giving the title, start date, and relevant sentences. This is illustrated in Fig. 3.

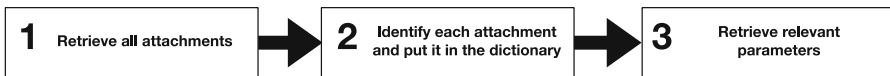


Fig. 3. Overview extraction attachments.

Retrieve All Attachments: For the contracts of Vanbreda Risk & Benefits, the attachments can always be found just before the contract. But it is safe to assume for other problems that the attachments will either be before or after the contract. Once the attachments have been found, a subset is created consisting of only the attachments.

Identify Each Attachment and Put It in the Dictionary: Once again regular expressions are used to find the title of each attachment. This regular expression will consist of the words “attachment”, “nr” and a digit. Once a match is found, the corresponding index is stored in a list. These indices allow for the identification of each attachment and the creation of a corresponding dictionary. Once again the keys equal the title of attachment and the corresponding value the content of the attachment.

Retrieve Relevant Parameters: Each title of an attachment is considered a parameter and the second parameter is the start date on which the attachment is valid. Regular expressions are used to find the start dates. In case more information is required later, the body of the attachment is also retrieved and stored as a third parameter. This subset starts at the end of the introduction and ends at the start of the conclusion. To not always retrieve the full content of the attachments, it is possible to identify and store key phrases containing words such as ‘the following has been changed’.

Creation of the Technical Summary: During the extraction process all parameters are stored in a dataframe. At the end of the process, the contents of this dataframe are automatically put into a spreadsheet file such as an Excel file. This Excel file consists of two columns containing parameter name and parameter value. For each contract, one technical summary is created containing the extracted parameters.

6 Experiments

6.1 Dataset

The proposed approach has been trained on 127 real-life contracts and has been evaluated on the two contract types: guaranteed income and waiver of premiums. The amount of time necessary for the creation of a technical summary by an employee was provided by Vanbreda Risk & Benefits. To determine whether a parameter has been extracted correctly, an expert has checked each one of them individually by going through each contract of the test set. The technical summaries created by humans have also been subject to control by the same expert.

It is only possible to perform parameter extraction that was correctly segmented by the program. The test set of type guaranteed income consisted of 21 real contracts out of which 15 were correctly segmented. The test set of type waiver of premiums also consisted of 21 real contracts and 17 contracts were correctly segmented.

6.2 Results

In Table 2, we report the number of parameters for each type of contract, the number and percentage of correctly extracted parameters, and finally the average processing time per contract. These results are reported for both manual and automatic parameter extraction.

Note that the number of parameters differs between automatic and manual extraction because automatic extraction also extracts attachments and some other parameters offering more information for the employees.

Table 2. Performance comparison: manual vs automatic.

	Guaranteed income		Waiver of premiums	
	Manual	Automatic	Manual	Automatic
Number of parameters per contract	14	17	7	10
Average number of parameters extracted correctly	10.8	13.9	4.5	8.1
% parameters correct	77.14	81.57	63.87	81.18
Average processing time per contract	15–20 min	4.91 s	15–20 min	5.23 s

7 Discussion, Limitations, and Future Work

7.1 Discussion

Results Analysis. From Table 2 we can conclude that automatic parameter extraction has achieved promising results. The time required to create technical summaries got reduced by approximately 99%. In addition, a high level of accuracy is maintained as automatic parameter extraction achieves an accuracy of above 80%. This method can

be used by insurance companies to create technical summaries automatically with a high level of accuracy, thereby reducing the workload on the employees with a significant amount.

Overall, we conclude that even though some manual fine-tuning is necessary to get the automatic extraction started, this is offset by the improved time necessary to create the technical summaries. The proposed solution has proven itself to be scalable, provided that most of the contracts follow the same template.

Automatic Extraction Errors. The errors produced by the automatic parameter extraction process are mainly due to the smart scanner Vanbreda Risk & Benefits is using and due to the packages used to read in the PDF file. We identified the cause of each misread parameter individually and found that approximately 10% of the errors were due to wrong scanning or manual handwriting on the original contract. If the PDF files were read in more accurately and if the contracts were not marked with handwriting, it is safe to assume those 10% could be extracted correctly. Other mistakes can be attributed to the fact that specific parameters were not present in the special terms part of the contract but located somewhere else in the contract.

Manual Extraction Errors. The manual extraction errors can mainly be attributed to three reasons. These reasons are inherently human mistakes.

1. Certain parameters were not filled in by the employees in the technical summaries.
2. The extracted parameter did not contain all necessary information to be classified as correctly extracted.
3. The extracted parameter was wrong.

7.2 Limitations

One limitation of our results is that only Dutch documents that already had pdf versions were used. To analyse different languages, the pipeline must be adapted by using different tokenizers for the respective language. Fortunately, this feature is already supported in the used NLTK package for prevalent languages such as English, Spanish, French, and German. It is possible that the results would improve on, for example, English documents as this language is more researched. Hence, this methodology is possible in different languages on the condition that the respective tokenizer exists and that the regular expressions are expressed in the corresponding language.

Both python packages, PyMyPDF and PDFMiner, can be used for reading PDF files and both have advantages and disadvantages. The PyMyPDF package is better at reading regular sentences whilst the PDFMiner packages is better at reading tables for example. Hence, careful consideration must be made when choosing which python package to utilise.

Extracting parameters automatically from attachments is inherently difficult as here the problem shifts from being structured to unstructured. It is challenging to know in advance in which attachments the parameters will be located. Moreover, attachments are not as standardised as contracts. Due to the unstructured nature of attachments, employees are better at capturing the relevant parameters. Therefore, it has been

decided to provide a list of attachments with certain common parameters in the technical summary such as start date or title of the attachment.

7.3 Future Work

As future work, the next steps include fine-tuning the approach to more contract types and insurance providers. We are also planning on improving the digitisation of contracts so that more contracts can be summarised automatically. This will lead to improving customer service at Vanbreda Risk & Benefits.

Moreover, the pattern-based approach works best for structured documents. When dealing with more variable documents, our approach becomes unstable. More research is needed into the usage of complex NLP algorithms to deal with parameter extraction in unstructured documents.

8 Conclusion

In this paper, insurance contract summarisation is performed using NLP. The proposed solution has been implemented at Vanbreda Risk & Benefits a large Belgian insurance broker on 127 contracts. From the results, we conclude that regular expressions are faster and perform as well as the employees at Vanbreda Risk & Benefits. The authors would like to express their gratitude to Vanbreda Risk & Benefits for the collaboration.

References

1. Zappa, D., Borrelli, M., Clemente, G.P., Savelli, N.: Text Mining in Insurance: From Unstructured Data to Meaning. Variance, Press. <https://www.variancejournal.org/articlespress/>. Accessed 23 March 2021 (2019)
2. Wang, Y., Xu, W.: Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud. *Decis. Support Syst.* **105**, 87–95 (2018)
3. Mosley Jr., R.C.: Social media analytics: data mining applied to insurance Twitter posts. In: *Casualty Actuarial Society E-Forum*. p. 1 (2012)
4. Benavides, T.: *Practical Human Resources for Public Managers: A Case Study Approach*. CRC Press (2011)
5. Donepudi, P.K.: AI and machine learning in banking: a systematic literature review. *Asian J. Appl. Sci. Eng.* **6**, 157–162 (2017)
6. Kankanhalli, A., Charalabidis, Y., Mellouli, S.: IoT and AI for smart government: a research agenda. *Gov. Inf. Q.* **36**, 304–309 (2019). <https://doi.org/10.1016/j.giq.2019.02.003>
7. Lamberton, C., Brigo, D., Hoy, D.: Impact of robotics, RPA and AI on the insurance industry: challenges and opportunities. *J. Financ. Perspect.* **4** (2017)
8. Balasubramanian, R., Libarikian, A., McElhaney, D.: *Insurance 2030—The Impact of AI on the Future of Insurance*. McKinsey Co. (2018)
9. Ly, A., Uthayasooriyar, B., Wang, T.: A survey on natural language processing (nlp) and applications in insurance. *arXiv Prepr. arXiv2010.00462* (2020)
10. Kolyshkina, I., Rooyen, M.: Text mining for insurance claim cost prediction. In: Williams, G.J., Simoff, S.J. (eds.) *Data Mining. LNCS (LNAI)*, vol. 3755, pp. 192–202. Springer, Heidelberg (2006). https://doi.org/10.1007/11677437_15

11. Liao, X., Chen, G., Ku, B., Narula, R., Duncan, J.: Text mining methods applied to insurance company customer calls: a case study. *North Am. Actuar. J.* **24**, 153–163 (2020)
12. Nuruzzaman, M., Hussain, O.K.: IntelliBot: a dialogue-based chatbot for the insurance industry. *Knowl.-Based Syst.* **196**, 105810 (2020)
13. Yogish, D., Manjunath, T.N., Hegadi, R.S.: Review on natural language processing trends and techniques using NLTK. In: Santosh, K.C., Hegadi, R.S. (eds.) *Recent Trends in Image Processing and Pattern Recognition*. pp. 589–606. Springer Singapore, Singapore (2019)
14. Loza Mencía, E.: Segmentation of legal documents. In: *Proceedings of the 12th International Conference on Artificial Intelligence and Law*. pp. 88–97. Association for Computing Machinery, New York, NY, USA (2009). <https://doi.org/10.1145/1568234.1568245>
15. Shah, P., Joshi, S., Pandey, A.K.: Legal clause extraction from contract using machine learning with heuristics improvement. In: *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pp. 1–3 (2018)
16. Chalkidis, I., Androutsopoulos, I., Michos, A.: Extracting contract elements. In: *Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law*, pp. 19–28 (2017)
17. McKie, J.X., Liu, R.: PyMuPDF. <https://pypi.org/project/PyMuPDF/>
18. Shinyama, Y., Guglielmetti, P., Marsman, P.: PDFMiner. <https://pdfminersix.readthedocs.io/en/latest/>
19. Hunt, J.: Regular expressions in python. In: *Advanced Guide to Python 3 Programming*. UTCS, pp. 257–271. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25943-3_22
20. Bird, S., Klein, E., Loper, E.: *Natural language processing with Python: analyzing text with the natural language toolkit*. O’Reilly Media, Inc. (2009)
21. McKinney, W.: Data structures for statistical computing in python. In: van der Walt, S. and Millman, J. (eds.) *Proceedings of the 9th Python in Science Conference*, pp. 56–61 (2010). <https://doi.org/10.25080/Majora-92bf1922-00a>
22. pandas development team, T.: pandas-dev/pandas: Pandas (2020). <https://doi.org/10.5281/zenodo.3509134>