Machine Learning Techniques for Automatically Extracting Contextual Information from Scientific Publications

Stefan Klampfl and Roman Kern

Know-Center GmbH
Inffeldgasse 13, 8010 Graz, Austria
{sklampfl,rkern}@know-center.at

Abstract. Scholarly publishing increasingly requires automated systems that semantically enrich documents in order to support management and quality assessment of scientific output. However, contextual information, such as the authors' affiliations, references, and funding agencies, is typically hidden within PDF files. To access this information we have developed a processing pipeline that analyses the structure of a PDF document incorporating a diverse set of machine learning techniques. First, unsupervised learning is used to extract contiguous text blocks from the raw character stream as the basic logical units of the article. Next, supervised learning is employed to classify blocks into different meta-data categories, including authors and affiliations. Then, a set of heuristics are applied to detect the reference section at the end of the paper and segment it into individual reference strings. Sequence classification is then utilised to categorise the tokens of individual references to obtain information such as the journal and the year of the reference. Finally, we make use of named entity recognition techniques to extract references to research grants, funding agencies, and EU projects. Our system is modular in nature. Some parts rely on models learnt on training data, and the overall performance scales with the quality of these data sets.

Keywords: PDF extraction, machine learning, named entity recognition

1 Introduction

The constant growth of the volume of scholarly publications makes it increasingly difficult to manage collections of scientific literature and to assess the quality of scientific output. It poses the need for automated processing systems that semantically enrich documents with information that support these tasks. One important aspect of scientific publications is that they are not isolated units, but originate in a specific *context*. In fact, several factors contribute to the development of a paper, for example, the authors' affiliations, funding information, or the venue or journal where a paper was presented or published. Also the list

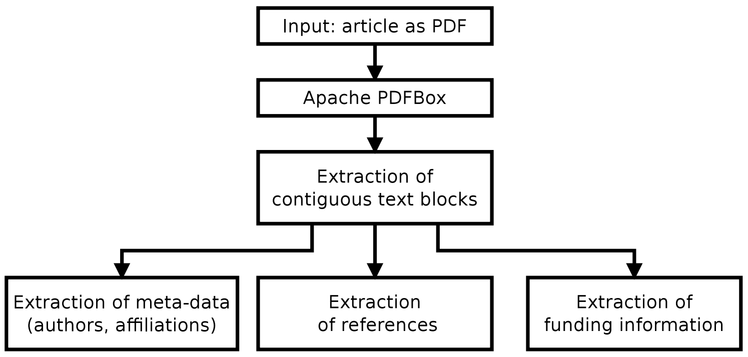

Input: article as PDF
Apache PDFBox
Extraction of contiguous text blocks
Extraction of meta-data (authors, affiliations)
Extraction of references
Extraction of funding information

Fig. 1. Overview of the overall architecture of our system, most of which builds on our previous work. Given a scientific article in PDF, the raw character stream obtained via PDFBox is clustered into contiguous text blocks [6], which serve as the basis for the following stages: the extraction of author and affiliation meta-data [4], the extraction of references [7], and the extraction of funding information [8].

of referenced papers constitutes important contextual information to take into account in order to assess its credibility and relevance.

To that end, the Semantic Publishing Challenge 2015[1] (SemPub 2015) asked participants to automatically annotate these elements within a set of input documents. This paper describes our contribution to Task 2 of this challenge, which focuses on the extraction of contextual information from scientific publications given as PDF files. PDF is the most common format for scholarly articles, however, it is optimised for presentation, but lacks structural information. It only contains information about individual characters and their position on the page, and this information might additionally be noisy. Intelligent and flexible algorithms are required that extract words with correct boundaries in the right order and group these words to lines and contiguous text blocks, which might then be classified to contain a specific type of information. Furthermore, these algorithms have to deal with the large variety of layouts of scholarly articles.

We have developed a system that exploits the flexibility of a variety of supervised and unsupervised machine learning techniques to deal with these challenges. It builds upon the open-source Apache PDFBox[2] library and processes a given PDF file in a number of individual processing modules (see Figure 1). First, it uses unsupervised learning (clustering) to analyse the physical layout of a scientific article by extracting contiguous text blocks, which we consider as the basic building blocks of a PDF document (section 2). Next, these text blocks on the first page of the article are classified into different meta-data categories, including authors and affiliations, using supervised learning (section 3). Then, heuristics are applied to detect the reference section at the end of the paper and to segment it into individual reference strings. The tokens of these reference

[1] https://github.com/ceurws/lod/wiki/SemPub2015
[2] http://pdfbox.apache.org/

strings are further categorised using sequence classification to obtain information such as the journal and the year of the reference (section 4). Finally, we use basic named entity recognition techniques to extract information about research grants, funding agencies, and EU projects (section 5). Parts of our system have already been described in [4, 7, 6, 5, 8]. A demonstration of the system can be accessed online[3], and the source code is available under an open source license[4].

2 Unsupervised extraction of contiguous text blocks as basic units of a PDF

Before we can extract contextual information from a scholarly article we have to process the low-level character stream of the PDF file to obtain logical units such as words or lines. The stream obtained through PDFBox consists of a list of characters, their bounding boxes (x and y position on the page, as well as their width and height), and information about their font. In our system, we consider contiguous text blocks as the basic building blocks of a PDF document. Each block consists of several lines, each of which is composed of a number of words, which themselves consist of multiple characters. The main challenge here is that the information provided by PDFBox might be unreliable: for example, height and width information might be slightly wrong, or information about the font of some characters might be missing. We therefore require algorithms which are flexible enough to deal at the same time with both this noisy data and the variety of layouts of scientific publications.

We use methods from unsupervised machine learning, in particular clustering, to to iteratively combine individual characters to words, lines, and blocks of text in a bottom-up manner. We employ a sequence of alternating *Merge* and *Split* steps: Each *Merge* step is implemented by hierarchical agglomerative clustering (HAC) with Euclidean distance measure and Single Linkage. In the first *Merge* step individual characters are merged to words: pairs of characters with increasing distance to each other are combined into clusters, until a maximum distance threshold is reached. Since the resulting clusters of characters might now encompass multiple words, a *Split* step is incorporated in the form of standard k-means clustering on the horizontal distances between characters ($k = 2$). Ideally, this partitions the spaces between characters into spaces between words and spaces within words, yielding the final set of words. This *Split* step can also be understood as an outlier detection which removes too large inter-character distances from the words obtained in the *Merge* step.

Another pair of *Merge* and *Split* steps is used to combine words to lines and lines to blocks. First words are merged to lines by combining pairs of words with increasing Euclidean distance to each other. This typically yields lines spanning multiple columns, which is resolved in the *Split* step that separates word spaces within columns from inter-column spaces. Finally, lines are merged to blocks,

[3] http://code-annotator.know-center.at
[4] https://svn.know-center.tugraz.at/opensource/projects/code/trunk

again by first combining them until a maximum distance threshold is reached, and then by splitting the resulting clusters at large vertical distances.

PDFBox already uses its own mechanisms for detecting word, line, and paragraph boundaries for the conversion to plain text. These methods are based on simple heuristics depending on the relative position of neighbouring characters. However, we decided to build our own generic text block extractor and did not reuse existing approaches provided by PDF parsing libraries, mainly because we want to leave open the possibility to apply our system also to other input formats, for example the output of OCR software. Another reason for not using PDFBox for the extraction of text blocks is that it does not provide any geometric information about these compound objects. It might also be desirable to extend our block extractor by incorporating font information or special rules such as the splitting of words at superscripts or subscripts.

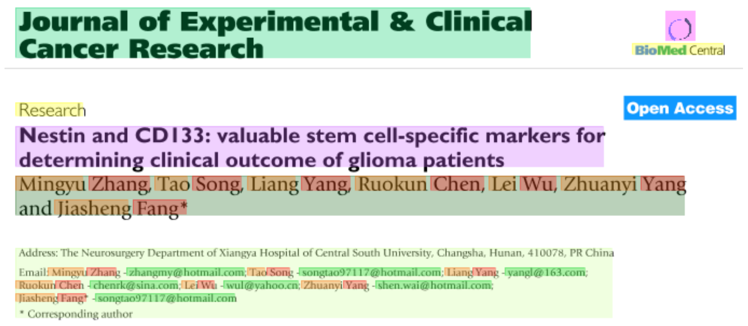
The result of this stage is a hierarchical data structure containing the geometrical information of blocks, lines, and words, as well as the reading order of blocks within the document [1]. Most importantly, this block structure effectively provides a segmentation of the text into single columns, a fact that is particularly helpful for extracting contextual information from references. We have presented a more detailed description of our algorithms as well as an evaluation of the block extraction in [6].

3 Supervised classification of author and affiliation meta-data

Major factors that directly contribute to the origin and development of a paper are the research institutions the authors of a scientific article are affiliated to, the venue where a paper was presented or the journal in which it was published. These meta-data thus constitute an important aspect of the context in which the paper was written.

For the extraction of meta-data from scientific articles we employed supervised machine learning techniques which use labelled training examples to learn a classification scheme for the individual text elements of an article. This stage directly builds upon the output of the text block extraction stage and consists of two phases. We first classify the text blocks extracted from the first page of the article into multiple meta-data categories: apart from author related information (names, e-mail addresses, and affiliations) we also categorise the title (and optional subtitle) of the article, the name of the journal, conference, or venue, abstract and keywords. For author-related blocks we then re-apply the classification to the tokens of these blocks in order to obtain given names, surnames, and affiliations.

As a supervised learning mechanism we use Maximum Entropy (ME) [2] combined with Beam Search [10], which incorporates sequential information by taking into account the classification results of preceding instances in order to avoid unlikely label sequences. Both algorithms are included in the open-source


Journal of Experimental & Clinical Cancer Research

BioMed Central

Research

Open Access

Nestin and CD133: valuable stem cell-specific markers for determining clinical outcome of glioma patients

Mingyu Zhang, Tao Song, Liang Yang, Ruokun Chen, Lei Wu, Zhuanyi Yang and Jiasheng Fang*

Address: The Neurosurgery Department of Xiangya Hospital of Central South University, Changsha, Hunan, 410078, PR China

Email: Mingyu Zhang - zhangmy@hotmail.com; Tao Song - songtao97117@hotmail.com; Liang Yang - yangl@163.com; Ruokun Chen - chenrk@sina.com; Lei Wu - wul@yahoo.cn; Zhuanyi Yang - shen.wai@hotmail.com; Jiasheng Fang* - songtao97117@hotmail.com

* Corresponding author

Fig. 2. Snapshot of a sample paper with text blocks on the first page classified into different meta-data categories, indicated by different colours, including journal, title, authors, and affiliations. The classification is further applied to the tokens of the author and affiliation block, yielding given names, surnames, and emails.

library OpenNLP[5]. The features used for classification are derived from the layout, the formatting, the words within and around a text block, and common name lists for detecting author names. Since the classification method is restricted to binary features, this information needs to be mapped to binary values.

We have called this the TeamBeam algorithm for extracting meta-data information. It has been described and evaluated in [4], where it was shown to achieve a satisfactory performance on a number of different datasets. Figure 2 shows a snapshot from a sample paper from the biomedical domain where title, author names, emails, and affiliations are correctly classified.

3.1 Classification of text blocks

In the first phase the text blocks are classified into the following labels: *Title*, *Subtitle*, *Journal*, *Abstract*, *Keywords*, *Author*, *E-Mail*, *Affiliation*, *Author-Mixed* and *Other*. Author related information might appear in separate text blocks, or different meta-data types might be combined in a single block (such as e-mail addresses and affiliations). In the latter case, this block would be labelled with *Author-Mixed*. The *Other* class is assigned to all blocks without any meta-data information.

The following features are generated for the text block classification:

- **Language model features** For each text block type, a language model is calculated by counting the frequency of words within this block type in the training set. This is used to generate features encoding the most probable block type for the words within the block.
- **Layout features** describe the position of a single block within a page: *isFirstBlock*, *isLastBlock*, *isLeftHalf*, *isRightHalf*, *isTopHalf*, *isBottomHalf*, *isRight*, *isLeft*, *isTop*, *isBottom*, *isCenter*.

[5] http://opennlp.apache.org

- **Formatting features** encode font and text flow: *isBigFont*, *isBiggerFont*, *isSmallFont*, *isSmallerFont*, *isLeftAligned*, *isRightAligned*. *Big/Small* and *Bigger/Smaller* are set if the deviation of the font size from the average font size exceeds ±1 SD and ±1.5 SD, respectively.
- **Dictionary features** consist of features *containsGivenName* and *containsSurname*, which are set depending on whether the block contains a word found in one of the common name lists: a list of 7,133 common first names taken from the GATE project[6], and a list of 88,799 most common surnames of the US Census[7].
- **Heuristic features** contains simple features such as *containsEMail*, *containsAtChar*, *containsDigits*, *containsPunctuation*, *containsDOI*, and *containsISSN*.
- **Term features** include all words within a block, as well as the first and last word of neighbouring blocks.

3.2 Classification of tokens

In the second phase of meta-data extraction the text blocks labelled with one of the author related types are further processed. The individual tokens of these text blocks are further classified into the following labels: *GivenName*, *MiddleName*, *Surname*, *Index*, *Separator*, *E-Mail*, *Affiliation-Start*, *Affiliation*, and *Other*. Because affiliations are often written in a sequence and often start with a common word, such as “University” or “Institute”, affiliation tokens are split into two parts, one for the initial word of an affiliation. The *Index* class is used for special characters linking authors to their affiliation and/or e-mail address, typically an asterisk or superscript numbers. Multiple index characters are separated by a token labelled as *Separator*, usually a comma.

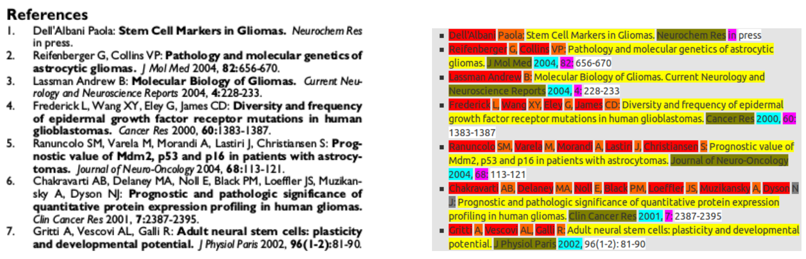
The following features are generated for the token classification:

- **Language model features** reflect the relative frequencies of words: *isCommonWord* (> 0.1), *isInfrequentWord* (< 0.01), *isRareWord* (< 0.001).
- **Layout features** encode the token’s position: *isFirstInLine*, *isFirst*, *isLast*.
- **Formatting features** include the font size compared to the average font size (*isBigger*, *isSmaller*), as well as the number of characters within the token.
- **Dictionary features** are set if the token occurs in one of the common name lists (*containsGivenName*, *containsSurname*).
- **Heuristic features** are reused from the block classification, including a feature for initials (upper-case character followed by a dot).
- **Term features** include a normalized version of the token itself as a feature.

4 Detection, segmentation, and tokenisation of references

Another contextual dimension to take into account in order to assess the credibility and relevance of a paper is the network of related papers, for instance,

[6] http://gate.ac.uk
[7] http://www.census.gov/genealogy/www/index.html



References

1. Dell'Albani Paola: **Stem Cell Markers in Gliomas.** *Neurochem Res* in press.
2. Reifenberger G, Collins VP: **Pathology and molecular genetics of astrocytic gliomas.** *J Mol Med* 2004, **82:**656-670.
3. Lassman Andrew B: **Molecular Biology of Gliomas.** *Current Neurology and Neuroscience Reports* 2004, **4:**228-233.
4. Frederick L, Wang XY, Eley G, James CD: **Diversity and frequency of epidermal growth factor receptor mutations in human glioblastomas.** *Cancer Res* 2000, **60:**1383-1387.
5. Ranuncolo SM, Varela M, Morandi A, Lastiri J, Christiansen S: **Prognostic value of Mdm2, p53 and p16 in patients with astrocytomas.** *Journal of Neuro-Oncology* 2004, **68:**113-121.
6. Chakravarti AB, Delaney MA, Noll E, Black PM, Loeffler JS, Muzikansky A, Dyson NJ: **Prognostic and pathologic significance of quantitative protein expression profiling in human gliomas.** *Clin Cancer Res* 2001, **7:**2387-2395.
7. Gritti A, Vescovi AL, Galli R: **Adult neural stem cells: plasticity and developmental potential.** *J Physiol Paris* 2002, **96(1-2):**81-90.

- Dell'Albani Paola: Stem Cell Markers in Gliomas. Neurochem Res in press
- Reifenberger G, Collins VP: Pathology and molecular genetics of astrocytic gliomas. J Mol Med 2004, 82: 656-670
- Lassman Andrew B: Molecular Biology of Gliomas. Current Neurology and Neuroscience Reports 2004, 4: 228-233
- Frederick L, Wang XY, Eley G, James CD: Diversity and frequency of epidermal growth factor receptor mutations in human glioblastomas. Cancer Res 2000, 60: 1383-1387
- Ranuncolo SM, Varela M, Morandi A, Lastiri J, Christiansen S: Prognostic value of Mdm2, p53 and p16 in patients with astrocytomas. Journal of Neuro-Oncology 2004, 68: 113-121
- Chakravarti AB, Delaney MA, Noll E, Black PM, Loeffler JS, Muzikansky A, Dyson N J: Prognostic and pathologic significance of quantitative protein expression profiling in human gliomas. Clin Cancer Res 2001, 7: 2387-2395
- Gritti A, Vescovi AL, Galli R: Adult neural stem cells: plasticity and developmental potential. J Physiol Paris 2002, 96(1-2): 81-90

Fig. 3. Snapshot of the beginning of the reference section of a sample paper (left) and extracted reference strings with classified tokens (right). The extraction of references involves the detection of the reference section within the paper and the segmentation of individual reference strings (bullet points). Individual tokens of the reference strings are classified into different categories (indicated by different colours), including given names and surnames of authors, titles, journals, years, volumes, and pages.

those that cite or are cited by a given one, or those that address similar issues. Researchers often use this information to search for literature that is relevant for their own work or for a specific field of research. Here, we focus on the information contained in the reference section, a section scientific articles usually conclude with and which acknowledges relevant and related work in the form of a list of citations or references.

The automated extraction of this type of contextual information requires the detection of the reference section within a paper, the segmentation of individual reference strings, and the labelling of single tokens within each string as to which field they belong (e.g., author, title, year, journal). We use heuristics to detect and segment references within a scientific article, and supervised sequence classification to assign labels to the tokens within each reference string. This part of our system has been described in [7], where we have shown that the extension of ParsCit [3], an existing state-of-the-art reference extraction system, with additional formatting and layout information improves the extraction of references. In particular we are able to correctly segment references with an F1 of about 0.94 and detect most reference token types with an F1 of at least 0.9 on a dataset from PubMed[8]. Figure 3 shows samples of extracted references, including segmented reference strings and categorised tokens, from an example paper.

4.1 Reference line extraction

The first step is to detect the reference section within a scientific article and directly builds upon the output of the text block extraction stage. We look for a specific heading that indicates the beginning of the reference section, which is usually one of “References”, “Bibliography”, “References and Notes”, “Literature cited”, and common variations of those strings (e.g., upper-case variants).

[8] http://www.ncbi.nlm.nih.gov/pubmed/

We iterate over all blocks in the reading order and use a regular expression to find the reference headers. Then we collect all lines until we encounter either another section heading, starting with "Acknowledgement", "Autobiographical", "Table", "Appendix", "Exhibit", "Annex", "Fig", or "Notes", or the end of the document.

In addition, we incorporate layout information into the reference line extraction in three ways. First, column information is implicitly provided through the consideration of text blocks. Second, we ignored the content of decoration blocks (headers and footers consisting of page numbers, authors, or journal names), which we computed by associating blocks across neighbouring pages based on their content and geometrical position [6]. Finally, we ignored all lines following a vertical gap that is larger than the average gap size plus two times the standard deviation of gap sizes. This criterion has been introduced as the block of references is often followed by footnotes, copyright information or other types of text which is visually separated from the references by a bigger gap.

4.2 Reference segmentation

After the reference lines of the article have been collected, the next step is the segmentation of these lines into individual reference strings. We distinguish three cases how reference strings can be marked: 1) with square or round brackets (e.g., "[1]" or "(1)"), 2) with naked numbers, and 3) strings are unmarked. For cases 1 and 2 the most common marker type is found via regular expressions, and the marker is also used to segment the references. For case 3 we incorporate layout information by looking for start lines which visually stick out from the rest of the lines, e.g., by a negative indentation. Our algorithm uses clustering to separate first lines from the rest of lines, assuming that the first lines will be the minority class.

We inspect each text block containing reference lines. If a block contains just a single line, this line is assumed to be an artefact of the PDF extraction process and is completely ignored. For blocks with more than two lines we cluster the lines using a simple version of the k-means clustering algorithm. The sole feature we use for this is the minimal x-coordinate of a line's bounding box. We set the number of clusters to 2 and initialize the two centroids with the minimal and maximal value of the feature. Then we assign each line to that centroid which is closer to the line's x-coordinate. We stop after a single iteration and update the centroids with the mean of the assigned features. At this stage all lines are assigned to one of the two clusters. Only if two conditions are met the layout based splitting is applied: The minimum cluster must contain fewer lines than the maximum cluster and the centroids differ by at least $0.05 * maxLineWidth$. If this is the case all lines from the minimum cluster are considered to be the first line of a new reference at which the reference lines are split into individual reference strings.

4.3 Reference preprocessing

The task of the reference preprocessing step is to clean the text of the references before the token classification is applied. The preprocessing consists of two parts, dehyphenation and normalization.

In the first part we resolve hyphenations by removing hyphens "-" and concatenating the split word parts if they are the result of a proper English hyphenation. For each line that ends with a hyphen we apply hyphenation on the concatenated word using a list of hyphenation patterns taken from the TEX distribution, and if the line split occurs at one of the proposed split points we resolve the hyphenation.

For normalization, we align the pages information to the form "<number>–<number>", even if there are multiple tokens or different dash characters.

4.4 Reference token classification

The final step is the categorisation of the individual tokens of the extracted and preprocessed reference strings. We used the following token types: *authorGivenName*, *authorSurname*, *authorOther*, *editor*, *title*, *date*, *publisher*, *issueTitle*, *bookTitle*, *pages*, *location*, *conference*, *source*, *volume*, *edition*, *issue*, *url*, *note*, and *other*. The class *authorOthers* is used for intermediate tokens in the author substring, such as "and".

At the core of the reference extraction process lies a supervised sequential machine learning algorithm. We use a conditional random field (CRF) [9], which has also been used in the original ParsCit system [3]. As implementation we use the freely available *crfsuite* software[9]. We use all the original features from the original ParsCit approach (see section 2 in [3] for the complete list). In addition, we incorporate layout and formatting information by a set of binary features specifying whether the font of the tokens inside a sliding window from -2 to +2 tokens is equal to the font of the current token. Two fonts are considered equal if they share the same font name, the same font size, and the same binary attributes specifying whether they are bold or italic.

5 Extracting funding information using Named Entity Recognition

Another major aspect in understanding the context in which a paper was written are the funding agencies, EU projects, and research grants that participated in funding a research and that obviously contributed to the development of a publication. These stakeholders are typically mentioned in the Acknowledgement section of a paper. In principle, we use basic techniques from Named Entity Recognition to extract this type of contextual information, however, our work is embedded within the larger goal of ontologically mapping the domain of computer science. We have recently made some initial steps in that direction [8].

[9] http://www.chokkan.org/software/crfsuite/

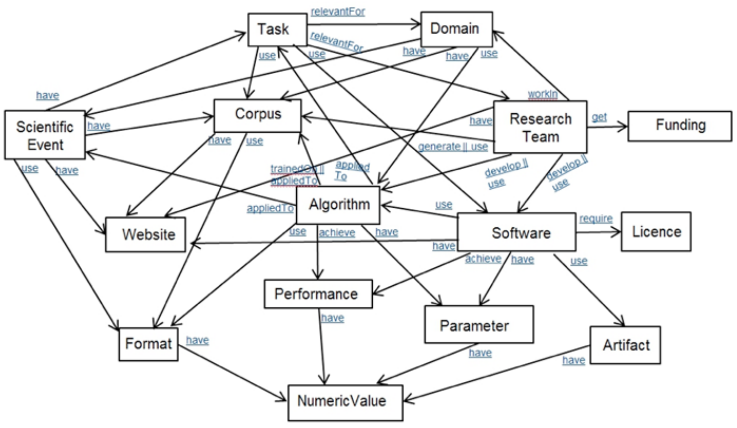


Fig. 4. Ontological description of the computer science domain, with the funding information being one of the concepts. The domain model contains categories (boxes) as well as linkage information in the form of relation categories (arrows).

For many domains ontologies already exist, which help to describe the content of scientific articles. This is in particular true for the biomedical domain; in other domains, such ontologies do not exist. In our recent work we found out that such ontologies are lacking for the domain of computer science [8]. Therefore we devised an ontological structure, which describes the main concepts in computer science literature, including the information about grants and funding agencies. In addition, we modelled the relationship between the concepts, see Figure 4.

Once the ontological structure had been finalised, we manually crafted a ground truth data set by annotating a selection of scientific articles. The final data set contained more than 5,000 manually curated entities and relations. This process shaped our understanding of the complexity of the task and possible ways to automatically infer these annotations.

Equipped with the understanding of what constitutes certain concepts in computer science literature, we realised a heuristics based automatic annotation scheme. In particular, for the funding information, we relied on a set of manually selected trigger phrases. We combined the information of the presence of one of the trigger phrases with the information of the noun phrases of the sentence, containing the trigger phrase. This first, basic approach already provided a performance of 0.79 precision and 0.70 recall in a preliminary evaluation.

Next, we applied machine learning for the automatic annotation of all concepts and relationship found in our ontological structure, including the information on grants and funding agencies. Therefore we utilised a general purpose information extraction pipeline, found in the CODE annotator[10]. This pipeline provides a flexible framework of different feature generation algorithms and highly configurable sequence classification algorithms. The framework itself is

[10] http://code-annotator.know-center.at

thereby not limited to the extraction of information from scientific articles alone, but can be applied on any textual resources. In the case of extracting funding information, it has been sufficient to convert the human annotations into the format suitable as training data for our framework. In this scenario we had only 37 instances of funding information, of which 30 were unique. This is a very low number for a supervised machine learning scenario, even too low to conduct a cross-evaluation. As a point of reference, our framework was able to achieve 0.75 precision and 0.61 recall, when tested on the training data set. This is certainly not representative of the performance, which can be expected in a real world scenario.

6 Conclusion & Discussion

In this work we have presented a system for extracting contextual information from scientific publications that are given as PDF files. It utilises the flexibility of both unsupervised and supervised machine learning techniques (i) to form words, lines, and blocks out of the raw character stream of the PDF, (ii) to classify these blocks into different meta-data categories, such as authors and affiliations, and (iii) to detect and segment reference strings and to classify tokens of these reference strings into different categories such as authors, title, journal, or year of publication. The features for our algorithms are composed of layout information (e.g., the absolute and relative geometrical positioning of text blocks on a page), formatting information (e.g., the type, style, and size of fonts), and textual information. Furthermore, we used techniques from information extraction and named entity recognition to extract information about funding agencies, research grants, and EU projects.

One major problem with PDFBox and other low-level PDF parsing tools is that the information provided about individual characters in the PDF is inherently noisy, for example, height and width information might be wrong, or information about the font of some characters might be missing. This implicit noise affects every stage of our system and thus its overall performance.

Parts of our system rely on models being learnt based on training data; hence the overall performance of our system also scales with the quality and size of these data sets. Our system is flexible and modular in nature and allows a separate training of different stages on different training sets. In many cases we used a subset of the PubMed database as a training set, mainly because it provides a rigorous annotation of the complete content of each document, in particular, meta-data and references. The publications in this database are from a wide variety of journals the biomedical domain, which we consider as representative for the general domain of scientific articles. Still it might not perform well on a specific sub-domain, such as conference publications from computer science. This would have to be addressed by a different training set that is more representative for this type of publications, which to the best of our knowledge does not yet exist in a reasonable size.

In the future we plan to address the aforementioned limitations and further improve the performance of the individual components. In particular, we plan to replace the remaining heuristics and manual rules by more flexible machine learning algorithms. Following this approach should enable us in the future to extend the information that we harvest out of scientific articles even further.

Acknowledgements

The presented work was in part developed within the CODE project (grant no. 296150) and within the EEXCESS project (grant no. 600601) funded by the EU FP7, as well as the TEAM IAPP project (grant no. 251514) within the FP7 People Programme. The Know-Center is funded within the Austrian COMET Program Competence Centers for Excellent Technologies under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. Aiello, M., Monz, C., Todoran, L., Worring, M.: Document understanding for a broad class of documents. International Journal on Document Analysis and Recognition 5(1), 1–16 (2002)
2. Berger, A.L., Pietra, V.J.D., Pietra, S.A.D.: A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics 22(1), 39–71 (1996)
3. Councill, I.G., Giles, C.L., Kan, M.y.: ParsCit: An open-source CRF Reference String Parsing Package. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., Tapias, D. (eds.) Proceedings of LREC. vol. 2008, pp. 661–667. Citeseer, European Language Resources Association (ELRA) (2008)
4. Kern, R., Jack, K., Hristakeva, M., Granitzer, M.: TeamBeam - Meta-Data Extraction from Scientific Literature. In: 1st International Workshop on Mining Scientific Publications (2012)
5. Kern, R., Klampfl, S.: Extraction of References Using Layout and Formatting Information from Scientific Articles. D-Lib Magazine 19(9/10) (Sep 2013)
6. Klampfl, S., Granitzer, M., Jack, K., Kern, R.: Unsupervised document structure analysis of digital scientific articles. International Journal on Digital Libraries 14(3-4), 83–99 (2014)
7. Klampfl, S., Kern, R.: An Unsupervised Machine Learning Approach to Body Text and Table of Contents Extraction from Digital Scientific Articles. In: Research and Advanced Technology for Digital Libraries. pp. 144–155 (2013)
8. Kröll, M., Klampfl, S., Kern, R.: Towards a Marketplace for the Scientific Community: Accessing Knowledge from the Computer Science Domain. D-Lib Magazine 20(11/12) (2014)
9. Lafferty, J., Mccallum, A., Pereira, F., Science, I.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the International Conference on Machine Learning (ICML-2001) (2001)
10. Ratnaparkhi, A.: Maximum Entropy Models for Natural Langual Ambiguity Resolution. Ph.D. thesis (1998)