A semi-supervised clustering method for payload extraction

by

Kyu Seok Lee

B.S., Korea University, 2008

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Electrical & Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2020

Approved by:

Co-Major Professor
William H. Hsu

Approved by:

Co-Major Professor
Don Gruenbacher

Copyright

Abstract

This thesis addresses *payload extraction*, the information extraction task of capturing the text of an article from a formatted document such as a PDF file, and focuses on the application and improvement of density-based clustering algorithms as an alternative or supplement to rule-based methods for this task domain. While supervised learning performs well on classification-based subtasks of payload extraction such as relevance filtering of documents or sections in a collection, the labeled data which it requires for training are often prohibitively expensive (in terms of the time resources of annotators and developers) to obtain. On the other hand, unlabeled data is often relatively easily available without cost in large quantities, but there have not been many ways to exploit it. Semi-supervised learning addresses this problem by using large amounts of unlabeled data, together with the labeled data, to build better classifiers. In this thesis, I present a semi-supervised learning-driven approach for the analysis of scientific literature which either already contains unlabeled metadata, or from which this metadata can be computed. Furthermore, machine learning-based analysis techniques are exploited to make this system robust and flexible to its data environment. The overall goal of this research is to develop a methodology to support the document analysis functions of layout-based document segmentation and section classification. This is implemented within an information extraction system within which the empirical evaluation and engineering objectives of this work are framed. As an example application, my implementation supports detection and classification of titles, authors, additional author information, abstract, and the titles and body of subsections such as 'Introduction', 'Method', 'Result', 'Discussion', 'Acknowledgement', 'Reference', etc. The novel contribution of this work also includes payload extraction as an intermediate functional stage within a pipeline for procedural information extraction from the scientific literature. My experimental results show that this approach outperforms a state-of-the-field heuristic pattern analysis system on a corpus from the domain of nanomaterials synthesis.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First and foremost, my greatest gratitude goes to my research advisor, William Hsu. He not only encouraged me to expand the knowledge of machine learning but also he has been a good model of a scientist and senior in life. Throughout the time I spent with him, I have also learned how to be a good supervisor, as well as a better engineer. Any conversation with him has great value, enough to expand my limitations.

I would also like to extend my profound thanks to Dr. Don Gruenbacher. I thank him for his wisdom, friendship and advice as a professional mentor. Special thanks to Dr. Caterina Scoglio for being on my thesis committee and for her plentiful advice throughout my MS studies. I deeply value her guidance on network analysis and how to deal with issues in data.

This thesis would not have been possible without the kind support and assistance from colleagues. I would specially thank my friend, Huichen Yang. He has been a good partner to help me cross-validate my hypotheses in research.

Finally, nobody has been more important to me in the pursuit of this project than the members of my family, I would like to especially thank my wife, whose love and assistance are with me in whatever I pursue. Importantly, I wish to thank my two loving, two wonderful children, Haneum and Hanyul, who provide unending inspiration.

Chapter 1 - Introduction

This chapter will first present the current status of technology development in the research area of procedural information extraction and its limitations. To give the reader a better understanding of past and current approaches to the problem domain, background knowledge concerning the theory and methods of my research topic (payload extraction from the scientific literature) are connected with the rationale and practical objectives of my work. Finally, this chapter will explain the goal, enumerate my novel contributions, and lay out criteria for the evaluation of my experimental system on a concrete task.

1.1 Overview

The advent of fast document digitization processes has raised scientific questions about how document analysis can better be automated (Fan, Wang, and Chang, 2001; Hao, Wang, and Ng, 1993; Thoma, 1999). There have been many successful researches of information extraction from the document, which ranges from character recognition to question answering (QA) function (Cheng et al., 2017; Mahajan and Zaver, 2018; Pham and Nguyen, 2014; Te, 2011). Using machine learning, the system has been rapidly developed to be able to provide request-specific answers (Al Chalabi, Ray, and Shaalan, 2015; Jose and Thomas, 2018).

Figure 1 shows the architecture of an end-to-end system for document based question-answering (Baudiš and Šedivý, 2015; Clausner, Pletschacher, and Antonacopoulos, 2011; Soares and Parreiras, 2018). Nowadays, the majority of QA researches are focused on proposing

enhanced methods of model representation for semantic analysis of question and answer (Fu, 2019; Shao et al., 2019; Sun and Xia, 2019).

However, to fully automate an end-to-end system, several technical requirements must be met. The system should be able to collect data sets by itself without human help, which is expensive. In addition, it must clean the collected data set and convert it into an appropriate format so data can be readable and processed by the machine. Moreover, the trimmed data needs to be structured so the machine can extract desired information from it. To mitigate this problem and meet the ultimate goal of end-to-end system construction, some literature has developed the data set crawler as a part of question- answering pipeline (Kadwe and Ardhapurkar, 2017; Rusinol et al., 2011; Shekhar, Agrawal, and Arya, 2010). These approaches implement a data collection process which is one of the requisite functions of a complete data management architecture.

The next step after automated data collection is a problem requiring further research. Once the system collects potential data by crawling the web, it must extract and process the information-bearing text, or *payload*, so the system can further analyze it for an overall desired purpose. This payload extraction stage is an essential bridge towards a fully automated end-to-end system. However, due to the limitations of existing research in payload extraction, research on information extraction systems for scientific literature has been fragmentary rather than holistic. There has been research directed towards automating the extraction of payloads using

algorithmic and rule-based systems (Aguirre et al., 2017, 2018) but these require both human background knowledge and additional effort to crop payload.

This chapter explains the complexities and difficulties of this task, beginning with a review of current prominent related work in the following section.

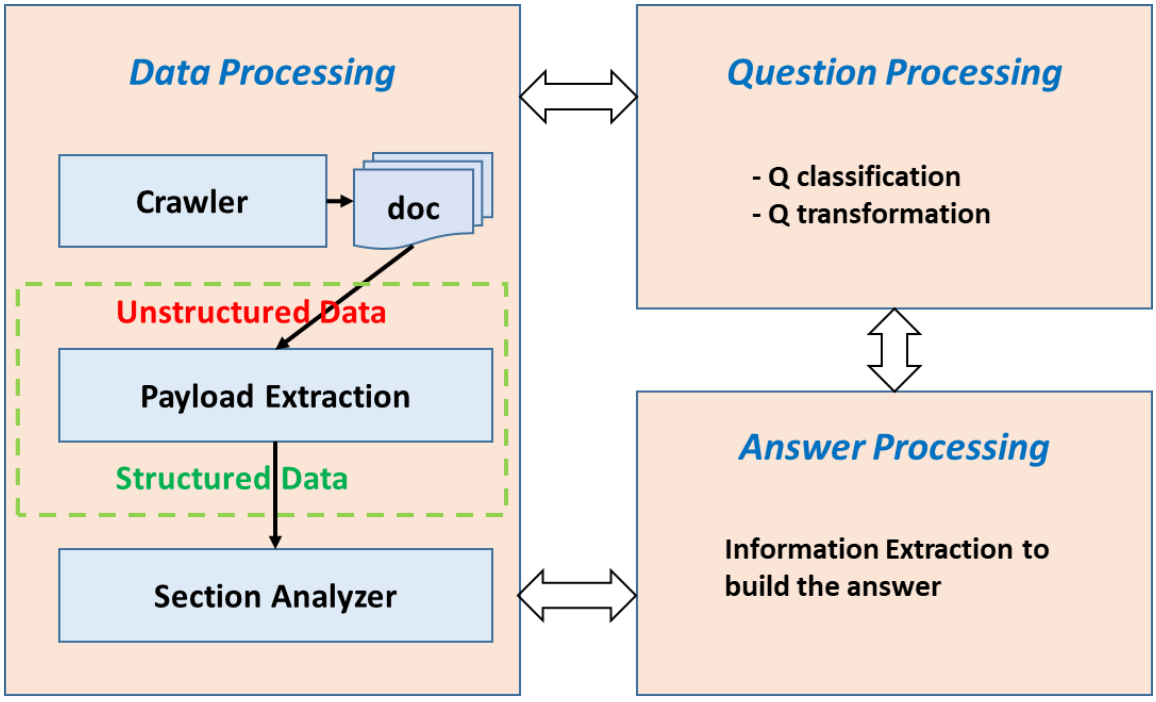


Figure 1. The architecture of three question answering modules. The two right modules have been aggressively researched but the sub-module (in the left module) of payload extraction requires more progress.

1.2 Limitation

In the payload extraction stage of a question-answering (QA) system, free unstructured data is converted to a structured format (Kumar et al., 2011). To be specific, the extracted metadata is analyzed, sorted and organized by criteria. However, the condition of layout and

arrangement varies from document to document(Lam, 1994). For example, a book may have a simple layout with a column format and without a picture while scientific literature shows the complexity of layout information such as a table, figure, page number and logo and so on. Even though we narrow down the domain to reduce the size of the problem into the case of literature, every paper has different layout styles so it is not possible to unify the criteria of structural classification.

In addition to that, the way of document analysis also varies from reviewer to reviewer. Since people have different preferences, intuitions and knowledge, it is not easy to conclude the analysis criteria to get identical ground truth. A person might classify a document into several categories: title, whole body paragraph and reference while the other person divides whole body paragraph more in detail, such as subtitle and its body paragraph by putting importance on a specific subtitle section. Because people have their preferences when they inspect the document and categorize it, their analysis reports are different. The same idea is applied when building an evaluation metric.

We might like to compare two documents in terms of similarity. Let us assume we have two identical scientific documents. If we intentionally remove one line of reference from the document, compare the similarity. Or we might intentionally change one line of title from the document and compare the similarity again. Comparing two cases, we might think the former case shows higher similarity because the part of the reference is less importantly considered in comparison metric or the other way around. Flexible and reasonable evaluation metrics are required to cope with the user's different preferences.

In this section, challenges of payload extraction task have been enumerated: 1. Layout complexity, 2. Ground Truth gathering issue and 3. Evaluation metrics. These challenges are solved in the proposed system. The following chapter introduces related researches.

1.3 Related Work

The most prominent approach for the payload extraction task is MATESC (De La Torre et al., 2018). MATESC classifies section in scientific document in algorithmic way so the output of MATESC can be directly loaded as the input of the section analysis stage. This method tries to cover section classification, but the format of the output is limited to text. Because the output of MATESC does not have spatial coordinate, writing style and so on, the next stage in QA system can only utilize limited information in later analysis. This problem might bring issues in terms of the general purpose of the system. For example, let us assume that our ultimate answer is a mathematical equation. In this case, font type and font style information may provide a useful clue for the construction of mathematical formula, but this information is dropped in MATESC.

Some researches addressed specific section classification, such as title, author, abstract, bibliographic section. GROBID (Lopez, 2009) shows specialty on section classification of title, author and abstract and bibliography section, but has brittleness on the classification of body

paragraph section. In (De La Torre et al., 2018), the table of accuracy comparison is provided to show its brittleness.

In (Clausner, Pletschacher, and Antonacopoulos, 2011), Aletheia demonstrates a ground-truthing system, trying to include spatial information in the method of cropping ground truth. The author presents an exact capture method for bounding boxes, which are expected to be processed later for a user-defined purpose. This work focuses only on semi-automated methods for concrete metadata cropping, to the exclusion of supervised classification learning methods, which constitute a significant facet of research on payload extraction tasks.

1.4 Task Specification

Payload extraction mainly focuses on section classification, so that each section is independently analyzed to produce text input for section analysis, the next stage in our procedural information extraction pipeline. The goal of the proposed payload extraction system reflects challenges mentioned in 1.2, as well as the current limitation of existing work, mentioned in 1.3. This thesis also provides a methodology for building a component-based system for a section classification tool. Therefore, this devised methodology can be adapted to multiple use cases of documentation layout classification. One long-term goal of this research is to demonstrate that this system can be adapted to multiple subject areas by means of incremental transfer learning guided by an end user who is a subject matter expert. I summarize the specification of the proposed system below.

1. The system mainly provides an accurate section classification method.
2. The system provides a platform that supports any type of features from the raw data set, so it is flexible to modify the internal algorithm for different feature selections.
3. The output of the system is represented by the set of segments of original input data; therefore, there is no loss of information.
4. The output of the system is safely fed into the next stage: section extraction..
5. The evaluation metrics are flexible enough to adapt to multiple user criteria.

As a result of the domain-adaptive design of this system, its output is suitable for *extractive* rather than merely *abstractive* summarization. Many extant systems usually take an input document and analyze it and filter out useless parts of the text payload. With this structure, we usually encounter possible issues. What if we want to use original data again which is not included in extracted one by finding a critical feature in input or deficiency of information in the following pipeline? Once we start extracting a part of input as we think it valuable, we might start narrowing down our ability to problem-solving. To keep the generosity and elasticity of the system, I devised the system exposing the inner controller outside. Figure 2 visualizes in-output of the system and open interface for users. This thesis calls the whole system as ‘DBMATE’ which is a shortened form of ‘DBSCAN-based MATESC algorithm. DBMATE is named after like the second version of MATESC since DBMATE displaces a stage in the QA pipeline, which

MATESC has been placed on. Hence, DBMATE takes the same input format and produced the classified sections as output.

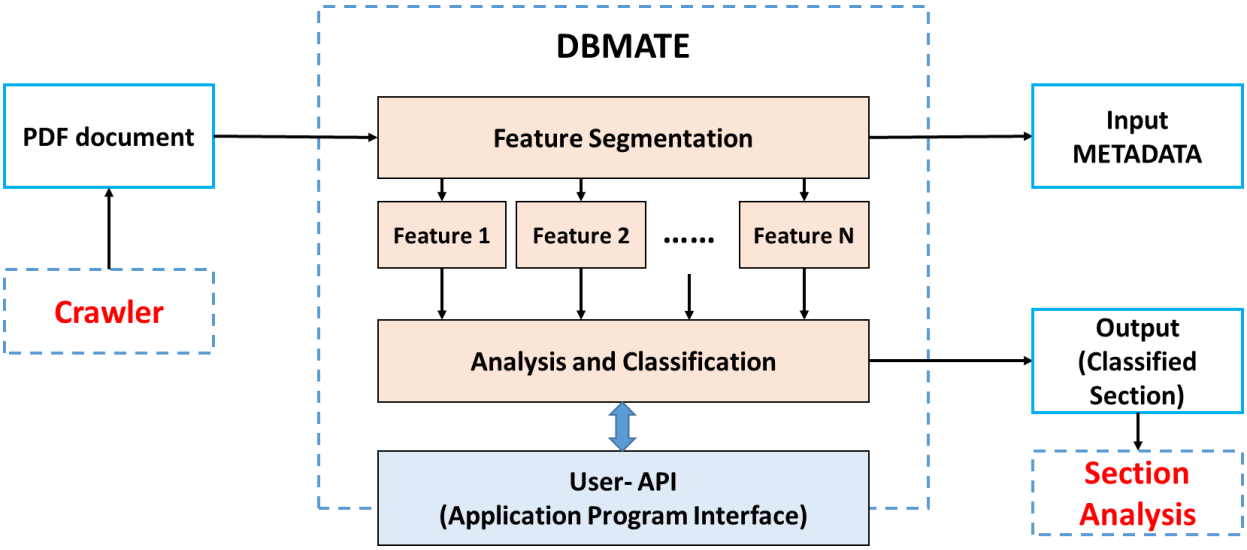


Figure 2. The configuration diagram of DBMATE system. Open API function enables to remodel internal analysis and classification algorithm.

As shown in Figure 2, the user programmable interface provides a function of tuning an internal classification algorithm to the end user. Once the user defines the standardized criteria of classification, he or she can easily tune internal hyperparameters through the API of the system. In Chapter 2, a novel approach to deriving parametric criteria for decision boundaries between desired and extraneous sections will be explained. Also, in the same chapter, the application of classification tuning will be demonstrated.

1.5 Significance

DBMATE is a new system that combines metadata-based pattern analysis with unsupervised learning (density-based clustering) for a payload extraction task. Overall, it applies

a semi-supervised clustering algorithm to improve on the state of the field. Also, it uses the name entity recognition method (NER), which brings a deeper analysis of data. I have planned the incorporation of a neural network algorithm into a DBMATE system, a technical contribution that will be introduced later in the future work section.

DBMATE provides a platform for layout analysis and classification in literature. Because DBMATE proposes the flexible feature engineering method to prevent information loss, any system that has the objective of layout analysis can incorporate DBMATE as a system component. It provides an accurate method for payload extraction. Our results show approximately 80% accuracy compared to ground truth. Compared to MATESC which attained 50% accuracy, DBMATE significantly improves the state of the field baseline.

Finally, DBMATE provides flexible output including ancillary metadata for multiple analytics, information extraction, and information retrieval. DBMATE produces output with classified text, together with spatial layout information, font type and font size while all the other methods only produce semantic information. These characteristics enable the system to increase accuracy in any evaluation metrics which require more information.

Chapter 2 - Learning Method

This chapter explains several algorithms or techniques for machine learning. It consists of a density-based clustering algorithm and section classification. This thesis first introduces the requirements and objectives for each task and explains how these tasks are solved using machine learning in a payload extraction system.

2.1 Density-Based Clustering Algorithm

This section mainly explains the density-based clustering algorithm, which I used for this research. To connect with the general clustering concept and to introduce one of its sub-categories, DBSCAN, a brief introduction of clustering is explained at first.

2.1.1 Clustering

Clustering analysis encompasses many different methods for grouping objects. A common objective of clustering faces to answer of how to reveal useful structures or meanings from given a collection of data. Because researchers have different goals and desired different output formats for analysis, various analysis methods for clustering have been studied (Cui and Potok, 2005; Zhao and Karypis, 2001).

There are four main clustering approaches: partitioning methods, hierarchical methods, density-based methods and grid-based methods (Han, Pei, and Kamber, 2011). I used a density-based method for this research since the given data set shows similar distance patterns between

adjacent objects (Ester et al., 1996). This thesis covers the practical application of this algorithm in detail in the next section.

2.1.2 Density-Based Clustering Algorithm

The basic idea of density-based clustering is that clusters are dense regions in the data space, separated by regions of lower object density. Therefore, points in a cluster have unique characteristics, that all points are density-connected points. The set of points $N_{eps}(q)$ in a cluster with a radius threshold *eps* is defined as in (Murphy, 2012):

$$N_{eps}(q) = \{\, q \; in \; D \mid dist(p,q) \leq eps \,\} \tag{1.1}$$

The cluster is defined in terms of radius parameter *eps* and density parameter *minpts*, representing the density-connectedness threshold within radius *eps*. *Eps* means the distance from core points (p) so every point within *eps* distance from *p* is clustered in the same group. *Minpts* is used to from cluster separately from an outlier. Density-based clustering detects outliers by using *eps and minpts*. Figure 1 shows an illustration of outlier detection. In Figure 3A (left), the red point is considered outlier or noise point since the number of directly reachable points within *k* distance is 2. By increasing the *eps* value, the red point is considered a core point of the cluster since the number of reachable points is 4, which is the same value as *minpts*. The algorithm keeps *eps* this process to other points and puts them in the same cluster if there is a point within the reachable distance threshold.

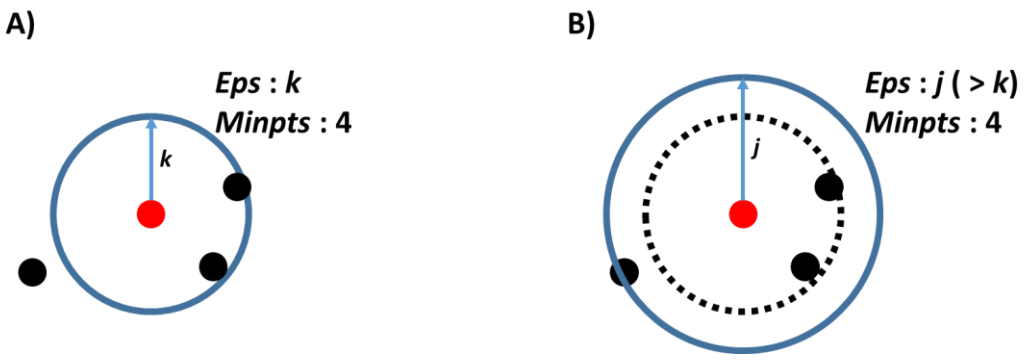


Figure 3. Illustration of clustering process A) core distance is smaller than B) core distance. The red point in A) is detected as an outlier since it only has 3 points within a radius of *eps*. On the contrary, the red point in B) is considered in a cluster because it meets the criterion of *minpts* = 4 within the *eps* boundary.

How does DBSCAN algorithm search and label clusters?

To utilize the DBSCAN algorithm in practical applications, the method of searching objects is explained. Initially, DBSCAN marks all objects in an input data set *unvisited*. Next, DBSCAN randomly chooses an object *A* at first and converts its status from *unvisited* to *visited*. After that, it inspects the other objects that are located within the distance *eps* from *A* and counts the number of those objects. If the number is greater than or equal to *minpts* value, the object *A* and its adjacent objects are labeled as a new cluster C, otherwise, they are considered outliers. DBSCAN repeats this process iteratively with other objects in clusters and remaining objects in the instance space, which are both marked *unvisited*. The pseudocode for this algorithm is given below.

Pseudocode for DBSCAN algorithm

mark all objects as *unvisited*;
DO
randomly choose a 'unvisited object *A*;
mark *A* as *visited*
IF the number of adjacent objects A' within distance *eps* from *A* is at least *minpts*
create a new cluster C, add *A* to C
let N be the set of objects of A'
FOREACH A'
IF A' is *unvisited*
mark A' as *visited*
IF the number of adjacent object A'' within distance *eps* from A' has at least *minpts*, add A'' to N
IF A' is not any member of a cluster, add A' to C
END FOREACH
output C
ELSE mark *A* as 'outlier'
REPEAT until no object is in *unvisited* status

2.1.3 DBSCAN in Payload Extraction

This section presents the optimized application of the DBSCAN algorithm in payload extraction. The objective of DBSCAN in payload extraction is to segment a page layout. In other words, DBSCAN groups unstructured metadata into meaningfully-structured clusters. Here," meaningfully-structured" reflects that the output of DBSCAN can be used as input to a text classification component. In the next stage, *section classification*, each input (the output of DBSCAN) is analyzed and classified into one of the following categories.

1. Title
2. Author
3. Author information
4. Abstract
5. Subtitle

6. A part of subtitle’s paragraph

8. Noisy data (header, footer, journal logo, page number and etc)

The clustering algorithm DBSCAN provides the basic functionality for single section extraction. However, in practical applications, each section has different distance values between members within the section. In the payload extraction task, the DBSCAN algorithm should include every span in a section into the same cluster, but separate spans from other sections. However, the variety of distance values between members in sections makes this task challenging. Figure. 4 shows the illustration of the challenge of section classification using DBSCAN. In the figure, the subtitle section consists of two spans: one is squared special character and the other one is the text of the Introduction section. The paragraph below is considered a body paragraph under the Introduction subtitle. The shadowed background square expresses the bounding box of the span. In this figure, a magnified circle shows the measured distance between spans. Distance (a) is the smallest distance between the subtitle section and its body paragraph, and distance (b) is the distance between spans in the subtitle section.

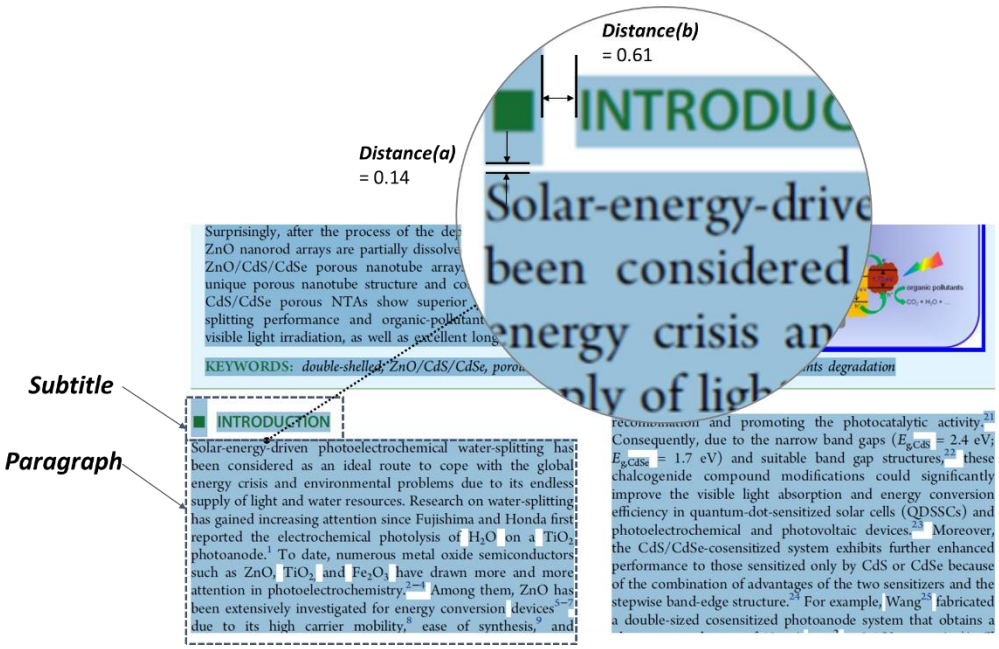


Figure 4. Illustration of distance measurement between section a) shows prospective section classification and b) shows the distance between 1) spans in 'subtitle' and span in 'paragraph' and 2) spans in 'subtitle'

If the DBSCAN algorithm uses an *eps* value of 0.14, the subtitle will be merged into the same cluster of the body paragraph, so it will fail to separate subtitle spans from the paragraph cluster. So inter-span distances below *eps* = 0.14 should be considered. However, any *eps* value smaller than 0.61 also splits the subtitle section into two clusters: a cluster of special character and text cluster of 'introduction'. Therefore, in this case, any *eps* value generates failure results.

To tackle this issue, DBMATE manipulates scaling parameters of other features, such as font size, font type and page number. For example, many subtitles have a bigger font size than its general font size of the paragraph. This uniqueness is utilized to separate subtitles from its body paragraph even when the spatial distance between sections is close enough so DBSCAN merges two sections in the same group.

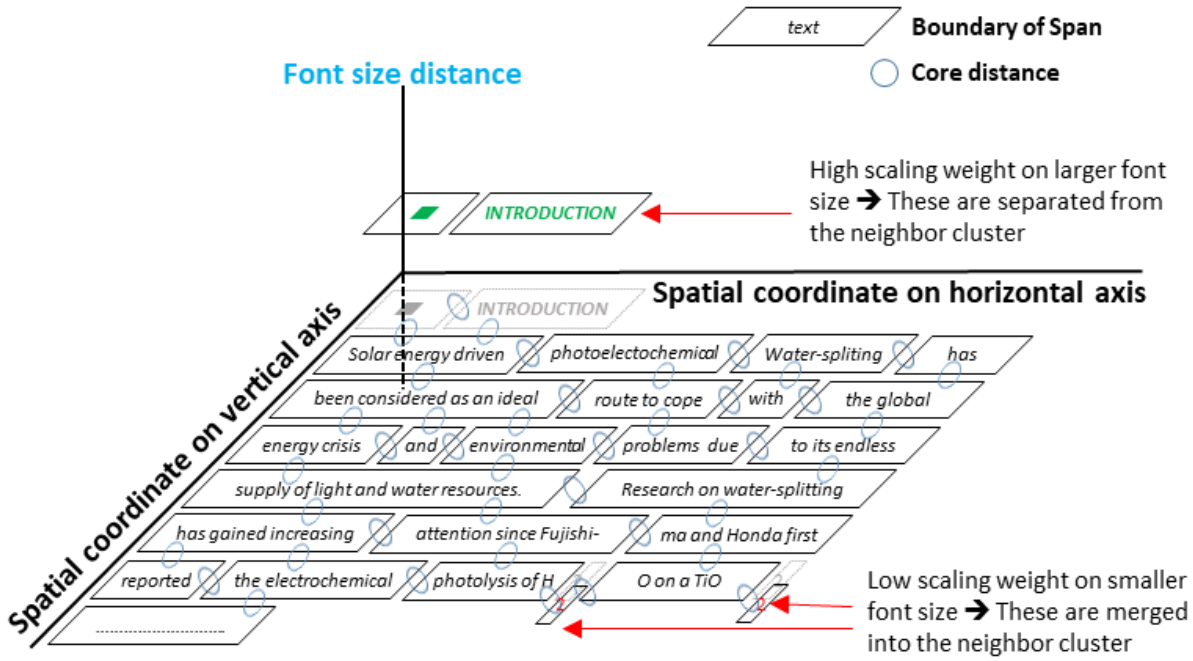


Figure 5. Example of feature manipulation on font size. DBSCAN controls distance on each feature axis by putting weight value.

Figure 5 visualizes the technique of feature manipulation in DBSCAN. The parallelograms with green text represent the span unit in subtitles shown in Figure 4 and other parallelograms with black text represent the spans of an adjacent body paragraph. In Figure 4 above, the subtitle cannot be distinguished from body paragraphs no matter what value of *eps* DBSCAN uses. By including the feature of font size, however, DBSCAN becomes capable of controlling the distance between feature domains of the font size. For example, DBSCAN can multiply constant value when font size is greater than the mode value so span with larger font size can be distant from neighbor spans. The same technique is applied to the smaller font sizes. The two red arrows at the bottom of figure 5 show the spans that have smaller font size values. Usually, the smaller font size in body paragraphs represents subscript, superscript or a part of the mathematical equation. These spans are part of paragraphs, so they should not be excluded from the neighbor cluster. By putting the small value on the scaler parameter on smaller font sizes, DBSCAN protects those spans from separation.

The above parameter control technique of DBSCAN must be calibrated, which we do by parameter updating based on training data, i.e., document corpora. If the value of the chosen weight parameter is wrong, the DBSCAN algorithm may cause over-fitting or under-fitting problem. To keep the most general algorithm, weight parameter variation on the scaler should be understood by the statistical proof. Figure 6 shows an example of statistical proof for scaling parameter selection. The font size used in the subtitle is relatively bigger than the font size used in body paragraphs. This data characteristic gives a clue of scaler parameter variation when the user intends to separate subtitle from its body. We can conclude, it is advantageous to have more distance value when a span has a larger font size than neighbors. With this statistical analysis providing evidence of parameter decision, DBMATE can safely make the most general hypothesis for clustering.

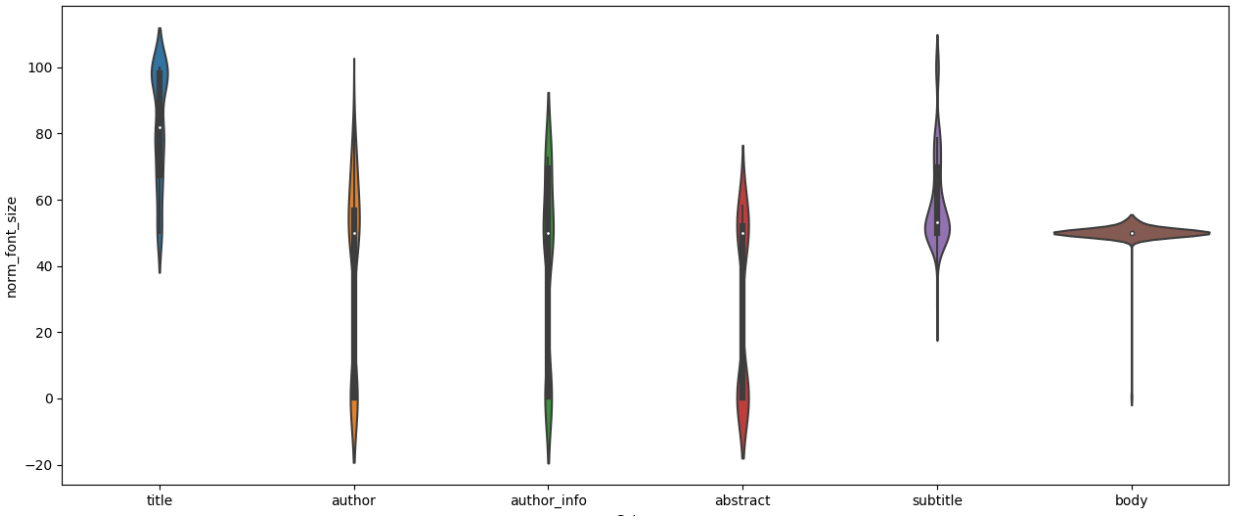


Figure 6.Font size distribution across sections. Font size is normalized to reflect relative increase or decrease between mod font size. Here, mode font sizes are mainly used to represent a characteristic of the body paragraph.

2.2 Section Classification

Section classification is the primary purpose of DBMATE; hence, the accuracy of section classification of scientific literature should be competitive with that of other methods. The same statistical analysis method is used for section classification to get accurate and general classification criteria. The detailed explanation is written in the following section.

2.2.1 Feature Selection and Probabilistic Prediction

Proper feature selection is a key requisite of section classification. The combination of features should contain the unique characteristics of the target class and separate the target class from others. However, in many practical cases, it is hard to find a unique feature pattern for the desired classification. Moreover, there might be noisy data or violation of features that discourage to make concrete classification standard. So it is desirable to show a probability that an element can be classified into a class of *y*. In the mathematical expression, the probability distribution over labels is simply described as $\mathrm{p}(\mathrm{y}|\mathrm{x}, D)$, where x is an input example and D is the training set. By taking symbol *C* as a class label, the predictive classification is computed using the equation below.

$$\hat{y} = argmax\, p(y = c|x, D) \tag{1.2}$$

Equation 1.2 is known as the MAP estimate (where MAP stands for *maximum a posteriori*) (Murphy, 2012). By taking the most probable label on classification, this can minimize classification failure and maximize the generality of the classification method.

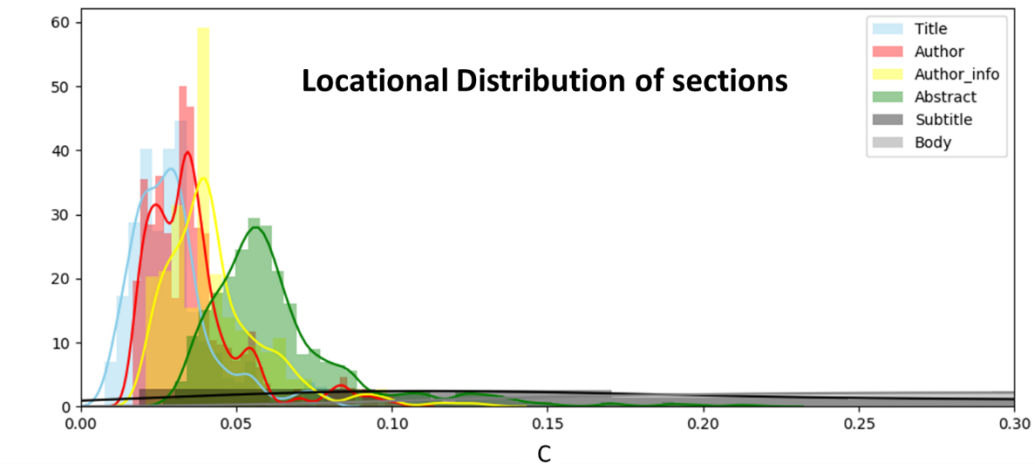


Figure 7. Histogram of spatial information for sections. This shows Title is located first followed by 'author', 'author information',' abstract. 'Subtitle' and 'body paragraph' are distributed all over the area of the papers. The value of C is calculated by normalized y coordinate. For example, when a document has 5 pages and the y coordinate of title section is 20 on the first page, C value is calculated by relative y coordinate against the whole document, so C is 0.2/6 = 0.033.

2.2.2 Input Data Analysis for Payload Extraction

To get a robust classification standard, the data analysis is prepared. As mentioned above, it is desirable to have the classification standard based on probability. Figure 6. shows the histogram by the spatial location of individual sections for 100 document papers. Title, author, author information and abstract sections are relatively located toward the beginning of the paper. This characteristic is used to set up criteria to separate the Title, Author, Author Information, and Abstract from Subtitle and Body Paragraph clusters. For example, any cluster that exists the upper part in the first page is much probably considered to be one of Title, Author and Author Information rather than Subtitle and Body Paragraph.

However, it is not enough to classify sections with only spatial information. Therefore, additional data analysis is required to make a concrete classification standard. Figure 8 is another example of input data analysis. In the histogram of occurrence of the person's name, 'Author' section uniquely shows relatively high frequency distribution. Hence, in the cluster analysis

stage, the cluster that has a high portion of the person's name is considered as 'Author section'. By adding these features to previously collected data analysis, the classification standard becomes more accurate.

DBMATE applies the same idea of data analysis on various features as below.

1. Cluster order
2. Text length
3. Font size
4. Page number
5. Occupied area

In the classification stage, DBMATE measures a cluster using various feature criteria. After that, DBMATE puts different weights on measured items and sums them to get the overall score. For example, when DBMATE classifies a section as the Title section, it analyses all the clusters and measures font size, word length, the order in the paper. In title classification, the occurrence of personal names and area are not considered so DBMATE puts zero weight on name occurrence and area values. After that, the section classifier selects the cluster that has the highest score and labels it as Title. The same idea is applied to other section classification: author, abstract and subtitle section.

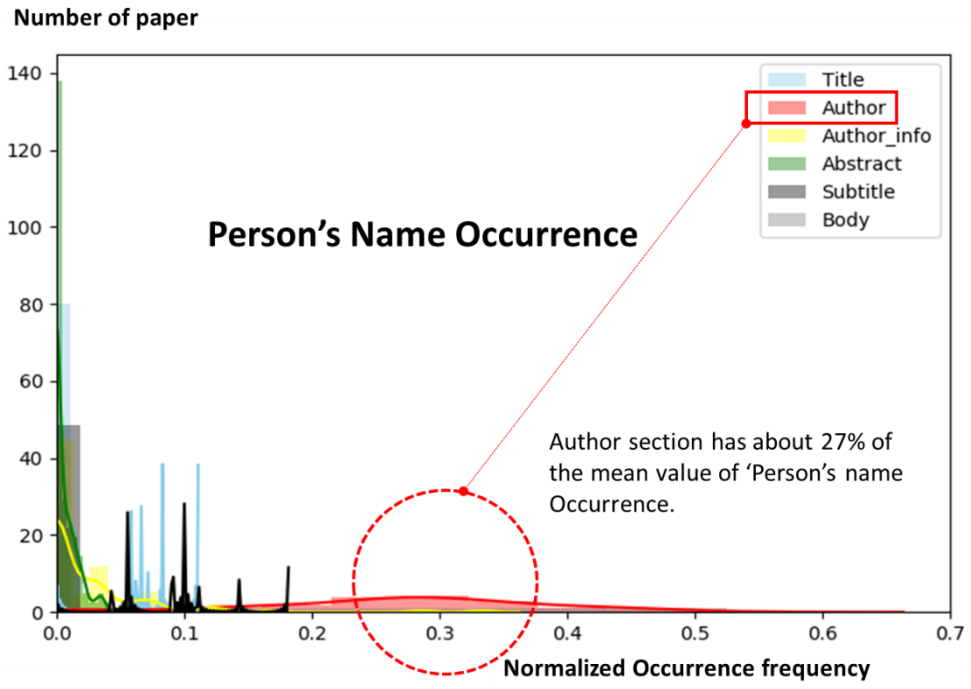


Figure 8. Name Occurrence Frequency of sections. In indicates that in Author section, uniquely it is shown that Person's name is mentioned in a larger portion, than other section domains.

Figure 9 illustrates the overall architecture for section classification in DBMATE. DBMATE varies weight values for different classification purposes. In input later, the green circle represents different analysis functions. The analysis function inspects clustered outputs (output of DBSCAN) and computes score by its functional criterion. The measured score value from cluster analysis is normalized and outputs to the next layer, called an internal layer. In the internal layer, the orange circle re-evaluates the output of the input layer by putting weight value and computing the overall score. For the last stage, the output stage, each classification section combines the previous results and generates the most probable section. Classified sections are labeled with distinct labels using a winner-take-all interpretation of the output, as distinguished from the feedforward outputs of input layers.

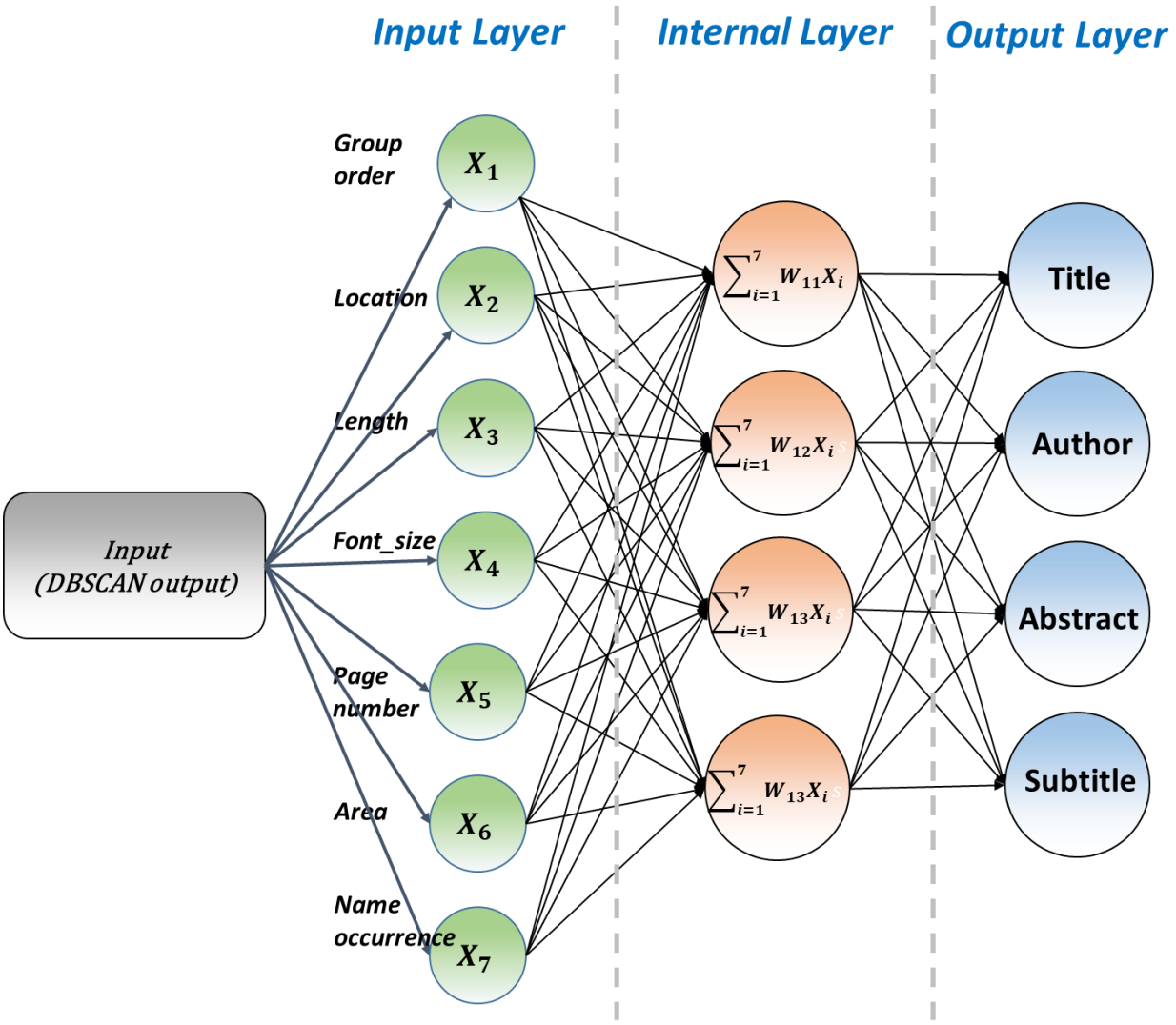


Figure 7. Illustration of classification method in DBMATE. It consists of input layer, internal layer and Output layer.

This architecture gives DBMATE two analytical benefits. First, all section classification is made by a comprehensive understanding of features. Since evaluation in the internal layer consider every possible combination of features so it prevents the construction of evaluation criterion from missing information. Furthermore, by considering the relationship between sections in the generation of the output layer, more complex classification criteria can be made.

Secondly, the parameterization of classification criteria enables DBMATE to accept the flexibility of criteria. By exposing the parameter table to the user-interface, adaptive parameter variation is available. This characteristic is connected to an automated system with other machine learning techniques, semi-supervised clustering method. This advantage is discussed in later sections of this thesis.

2.3 DBMATE Overview

This section summarizes the overall architecture of DBMATE to help readers grab the clear overall understanding of the proposed algorithm. This section first explains the overall architecture and later provides pseudocode so readers can reproduce the output in the same way.

2.3.1 The Overall Configuration

The proposed method is based on a semi-supervised learning technique. The overall architecture of the workflow is illustrated in Figure 8. Once getting the data set, it is split into the training set and the test set. It is desired to use the sampling method to reduce the size of ground truth so we save manual labeling time. Once the labeled data set is ready, *Section Analyzer* starts the analysis of the ground truth section by section. Location, font size, font type, page number, text length, occupying area and the number of name occurrences of every paper are separately extracted and recorded. By taking the range of 99% of each extracted item, criteria ranges used in section classification are decided. These criteria are used to construct the section classification model.

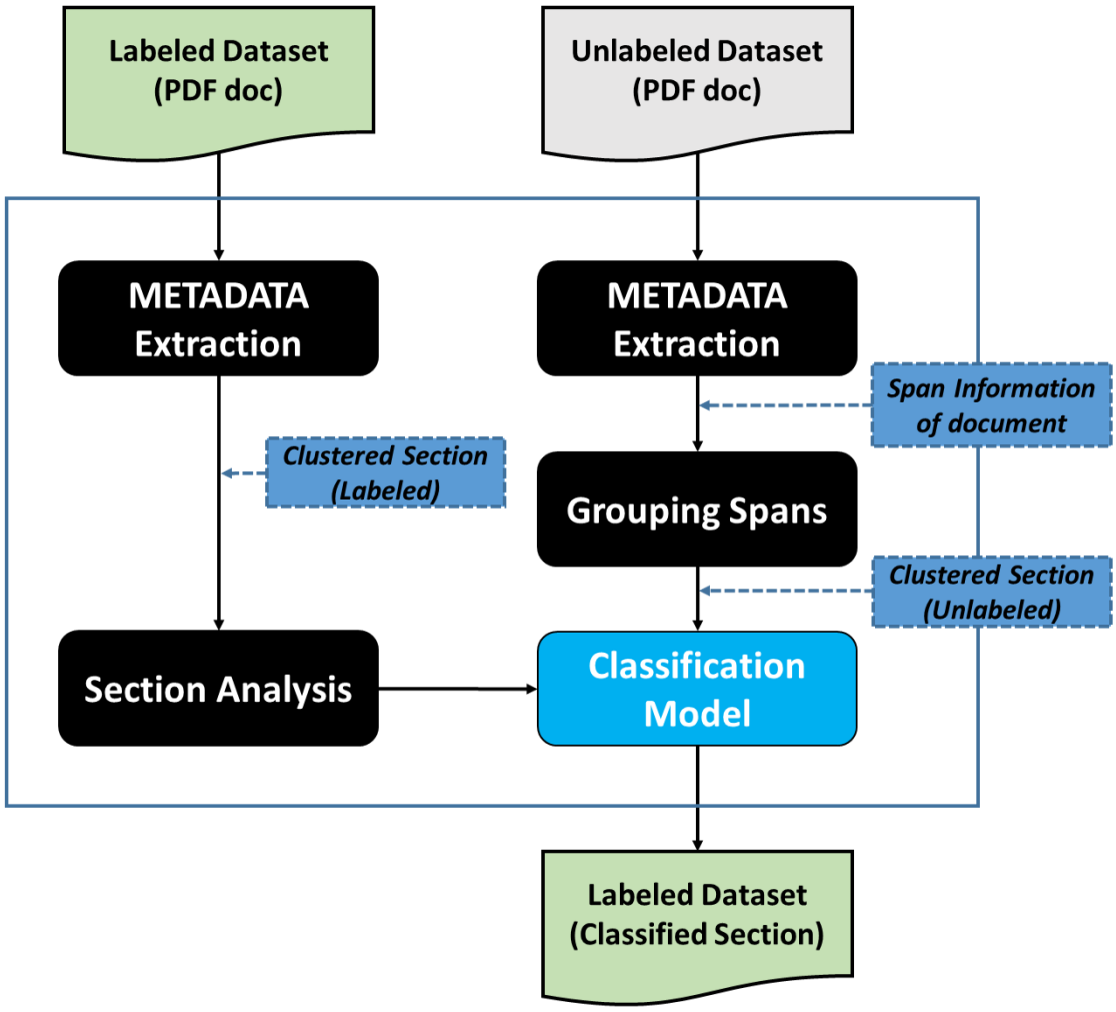


Figure 8. The overall framework of DBMATE: In the above semi-supervised learning framework, the classification model is trained in 'Section Analysis' module. This trained model analysis unlabeled clusters that is obtained from the grouping algorithm.

Section Analyzer also finds the optimized *eps* value and scale parameter for the clustering algorithm. *Section Analyzer* tries the different hyperparameter combination of *eps* and scale parameters of the DBSCAN algorithm and finds the best combination for each different section. This best combination of hyperparameter is later used in the application of DBSCAN algorithm for the unlabeled data set.

Once hyperparameters are decided in *Section Analyzer*, unlabeled raw data are plugged in DBMATE. DBMATE converts the raw PDC documents into the form of metadata. Once the metadata of each document is produced, the character-based metadata is merged into the span unit. After grouping these spans into the unlabeled section paragraph, the corresponding section labels are annotated to the section paragraphs. In Figure 8, it is shown that the classification model which is extracted from *Section Analyzer* is re-used to classify sections generated from the unlabeled data.

2.3.2 Pseudo-Code

This section prepares the pseudocode of both two main diagrams: *Section Analyzer* and the section classifier of DBMATE. *Section Analyzer* first analysis the labeled document and extracts the classification model of the DBMATE. This model is used in the section classification stage of DBMATE.

---

* General algorithm for extraction of Section Classification Criterion in ***Section Analyzer***

Given:

Labeled document set $\boldsymbol{D} = ((\boldsymbol{X^{(1)}}, \boldsymbol{Y^{(1)}}), \dots, (\boldsymbol{X^{(n)}}, \boldsymbol{Y^{(n)}}))$ such that $\boldsymbol{X^{(i)}} = (x_1^{(i)}, \dots, x_q^{(i)})$ and $\boldsymbol{Y^{(i)}} = (y_1^{(i)}, \dots, y_q^{(i)})$ where $x_j^{(i)}$ is a section in pdf document and $y_j^{(i)}$ is its corresponding section label.

Set of features $\boldsymbol{A} = (a_1, \dots, a_l)$ such that $\boldsymbol{A}_j^{(i)}$ is the collection of features of $x_j^{(i)}$

Set of attributes of section label $\boldsymbol{B} = (b_1, \dots, b_t)$ such that b_k corresponds to $y_k^{(i)}$

Set of section criterion $\boldsymbol{C} = (\boldsymbol{c_1}, \dots, \boldsymbol{c_t})$ such that $\boldsymbol{c_r}$ =$(I, \dots I_l)$ and $I_k = (I_k^{lower}, I_k^{upper})$ where I_k^{lower} = lower boundary value of feature a_k *in* $\boldsymbol{G}_r$ and I_k^{upper} = upper boundary value of a_k *in* $\boldsymbol{G}_r$

Algorithm Section Analysis **(*D*, *A*, *B*):**

 # Aggregate blocks with target section into analysis data group: ***G***

for i **= 0** → n **do**

  for j = 0 → q **do**

    for k = 0 → t **do**

      if $b_k = y_j^{(i)}$ **then**

        $\boldsymbol{G}_k \leftarrow \boldsymbol{A}_j^{(i)}$ of $\boldsymbol{X}^{(i)}$

return ***G***

Criterion extraction function from ***G***

for i = 0 → t **do**

  for j **=** 0 → l **do**

    extract i_k of a_j in $\boldsymbol{G}_i$

    $\boldsymbol{C} \leftarrow i_k$

return ***C***

---

* General algorithm for Section Classification in ***DBMATE***

Given:

Unlabeled document set $\boldsymbol{D'} = (\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(n)})$

Criterion matrix $\boldsymbol{C}_{t,l,2}$ obtained from ***Section Analysis***

Classification function Q

Algorithm of Section Classifier in *DBMATE*$(\boldsymbol{C}, \boldsymbol{D'})$:

for i **= 0** → n **do**

  for j = 0 → q **do**

    extract I_j' of a_j of $x_j^{(i)}$

    Classified Section ← $Q(I_j', \boldsymbol{C})$

Chapter 3 - Methodology

This chapter presents my approach to payload extraction within a pipeline developed by the Kansas State University Laboratory for Knowledge Discovery in Databases (KDD Lab). In it, I introduce the hybrid algorithm DBMETA, an extension of the amalgamated heuristics of MATESC (*Metadata-Analytic Test Extractor and Section Classifier for Scientific Publication*), which was previously developed for the payload extraction problem addressed in this thesis, with the general-purpose density-based clustering algorithm DBSCAN. Figure 9 shows the pipeline of DBMATE. My work takes PDF documents as input data. With using collected PDF files, DBMATE extracts metadata and groups the metadata into clusters using DBSCAN, a clustering algorithm of machine learning. The grouped clusters are analyzed and classified within the section classifier and labeled with the predicted section name. This thesis explains the detailed algorithm in the following sub-section.

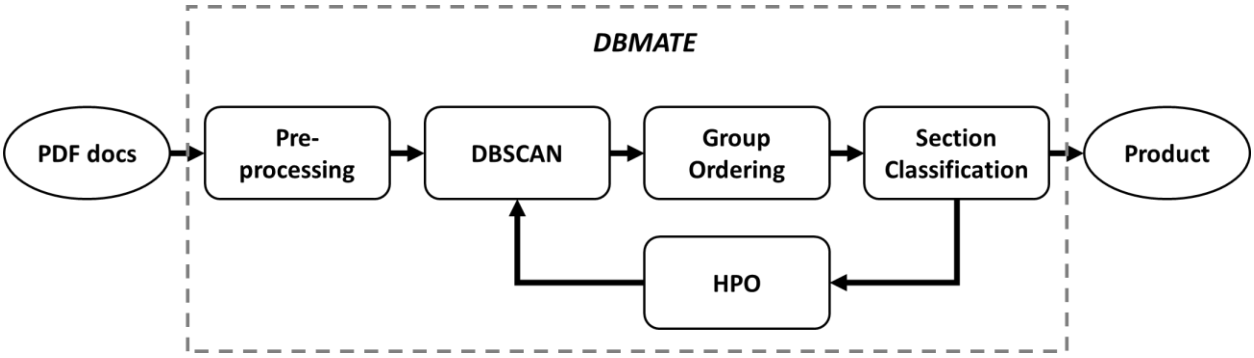


Figure 9. DBMATE Pipeline. DBMATE system takes PDF document as its input format and produces section-classified layout as its output

3.1 Metadata extraction

To analyze PDF files, PDF files must first be converted into the machine-readable format that can be easily processed by a computer. The original PDF file is the only image file that consists of pixel information so it is required to convert high-level format so the computer easily handles to proceed with. To meet this objective, DBMATE converts PDF file into text format using PyMuPDF (Liu and Mckie, 2018), a tool that converts PDF documents into the pre-defined text format, which is called as metadata. The metadata has information on each character, including location coordinates, font type, font size, page number in document and so on.

3.2 Character merging

Previously extracted metadata consists of characters as its basic unit. However, character utilization for machine learning algorithm and classification brings several following issues.

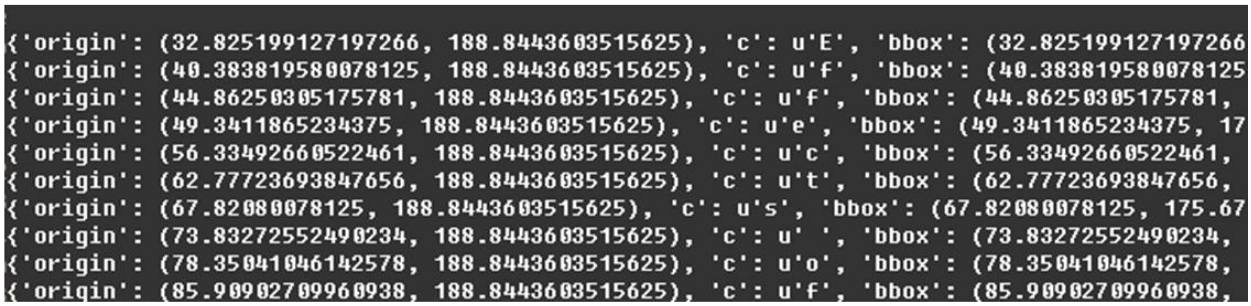
First, the character itself does not contain the word-level meaning. Semantic analysis on word-level is used later in section classification step, therefore the utilization of word-level unit brings a benefit when analysis.

Second, character unit handling causes a memory consumption issue. For example, for a 12-page PDF file, approximately 36000 characters are used. Each character has its own font size, font type, bounding box information and page number so approximately three hundred and thousand variables should be stored to be utilized in the program. Instead, word level unit handling reduces its size of memory allocation by one hundredth with a sufficient amount of

information. DBMATE uses ‘span’ as a basic unit, therefore, as shown in Fig.2, multiple characters are merged into a span if the character has the same font type, font size, page number in the sequence.

Thirdly, the DBMATE algorithm reflects human intuition when it analyzes the document file. The classification method used by DBMAT imitates the classification behavior of a human. When human readers and encounter words of different font sizes, they sense there is more than usual meaning and check if it has further meaning. For example, while one is reading a paper and meet a short phrase with the larger font size and bold font type, one may recognize it as a subtitle or specially emphasized word. For another example, we assume that people suddenly encounter subscriptions or superscription while reading a paper, people think it might be a special character or a word such as annotation number, a part of a chemical compound formula or mathematical equation, instead of plain text. Because any change in font size and font type, special character usage, newline is able to deliver structural or semantic variation, DBMATE slices sentence or word when it recognizes any variation of the above features.

a



b

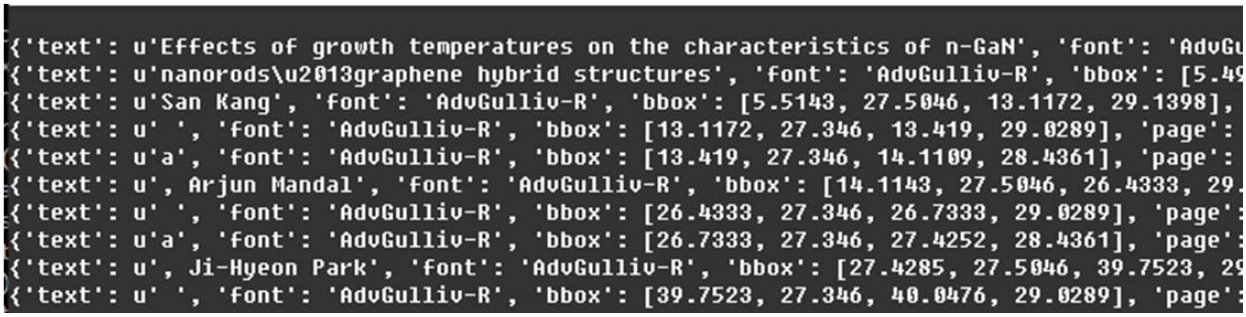


Figure 10. Basic unit conversion. In DBMATE, (a) character -based format is converted to (b) span-based format.

3.3 Clustering (DBSCAN)

The primary goal of the DBMATE algorithm is section classification. Possible classification target values for the section type in scientific papers include Title, Author, Abstract and Body. Body classification is again subdivided into pre-text, subtitle and paragraphs. This thesis explains the method of section classification in detail later. All of this section has a common characteristic. They consist of multiple layout units or spans, and these spans are located adjacent to other members in the same section. Utilizing spatial characteristics of each span, DBSCAN uses a density-based clustering algorithm to group spans into a higher level of category.

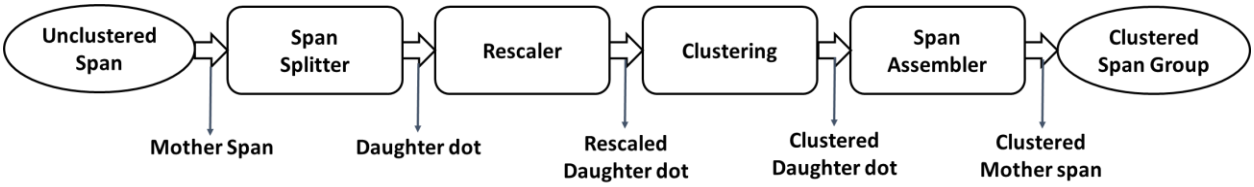


Figure 11. Clustering Process Pipeline

Figure 1. shows the detailed process illustration of the clustering step in the DBMATE algorithm. From the previous character merging step, constructed spans are given. Each span is expressed with font type, font size, page number and bounding box that has location information and occupying area in a rectangular shape in the pdf file.

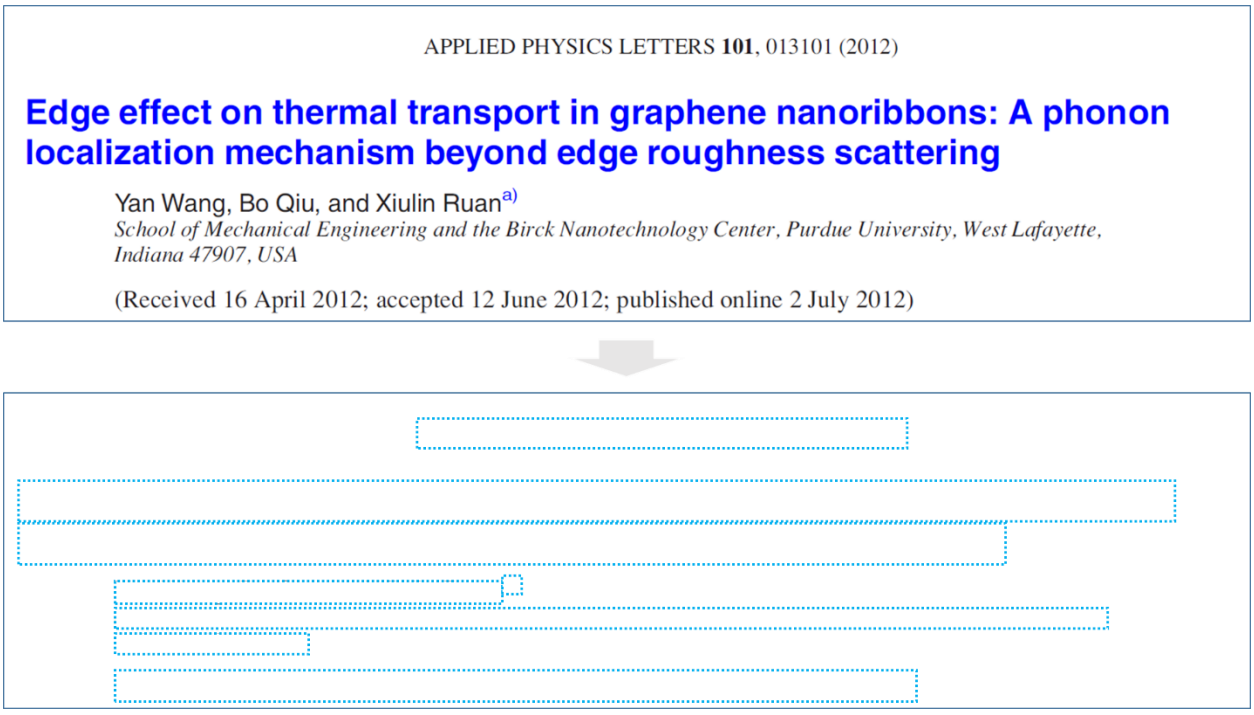


Figure 12. Illustration of input (top) and output (bottom) of the span splitting process

The span splitter converts the bounding box of the span into many small dots to plug in the DBSCAN algorithm as an input. This is an essential process since the DBSCAN algorithm only takes point as the input format for the grouping process. Figure 4 shows the graphical illustration of this process. The bounding boxes of all mother spans in a document are converted into many small daughter dots to represent their spatial information. In this process, small dots have not only information, but also keep the rest of the information of font size, font type and page number that the original mother span has.

The DBSCAN algorithm uses daughter dots and makes groups by measuring the distance between two points. Therefore, if any two dots are close enough, the DBSCAN puts the two points in the same group. DBSCAN uses not only spatial information to measure the distance between dots, it also considers font size, font type, the ratio of x coordinate over y coordinate and page number to measure distance. For example, even though two separate points are close enough for each other to be in the same group if their font sizes are different, DBSCAN puts more value on distance measure so the distance between two points is increased by the font size difference. The same logic is applied to measuring by other features. DBSCAN can put different weights of features when it measures distance, it is called scaling. By this scaling functionality of DBSCAN, we can compute various other features as well as location information.

After the clustering process, grouped dots are used to restore the original span format. In the splitting process, DBMATE built a lookup table including pairs of mother spans and the produced daughter dots. By matching with this table, DBMATE can obtain the clustered mother span.

3.4 Group Ordering

Once DBSCAN outputs clustered span groups, the ordering process is followed. This process is important because later classification algorithm matches the subtitle and its paragraph by the group order. For example, when DBMATE finds the paragraph of a subtitle, DBMATE chooses the next groups until DBMATE finds the next subtitle. Therefore, if the group ordering algorithm is poor, it causes the wrong selection of the subtitle's paragraph. To address this, this thesis presents column-aware group ordering. First, DBMATE scans the whole page of the document along with y coordinate and detects if there are more than two columns. While this process, DBMATE records multi-column ranges and uses it for the next ordering process. Once DBMATE finishes the scanning step, it re-orders every group by the following rule. Figure 13 illustrates an example of this ordering process.

Rule 1. If the group is in single-column range, DBMATE orders groups along with x axis.

Rule 2. If the group is in multi-column range, DBMATE separate groups in ranges by column. For each column, apply rule 1 again.

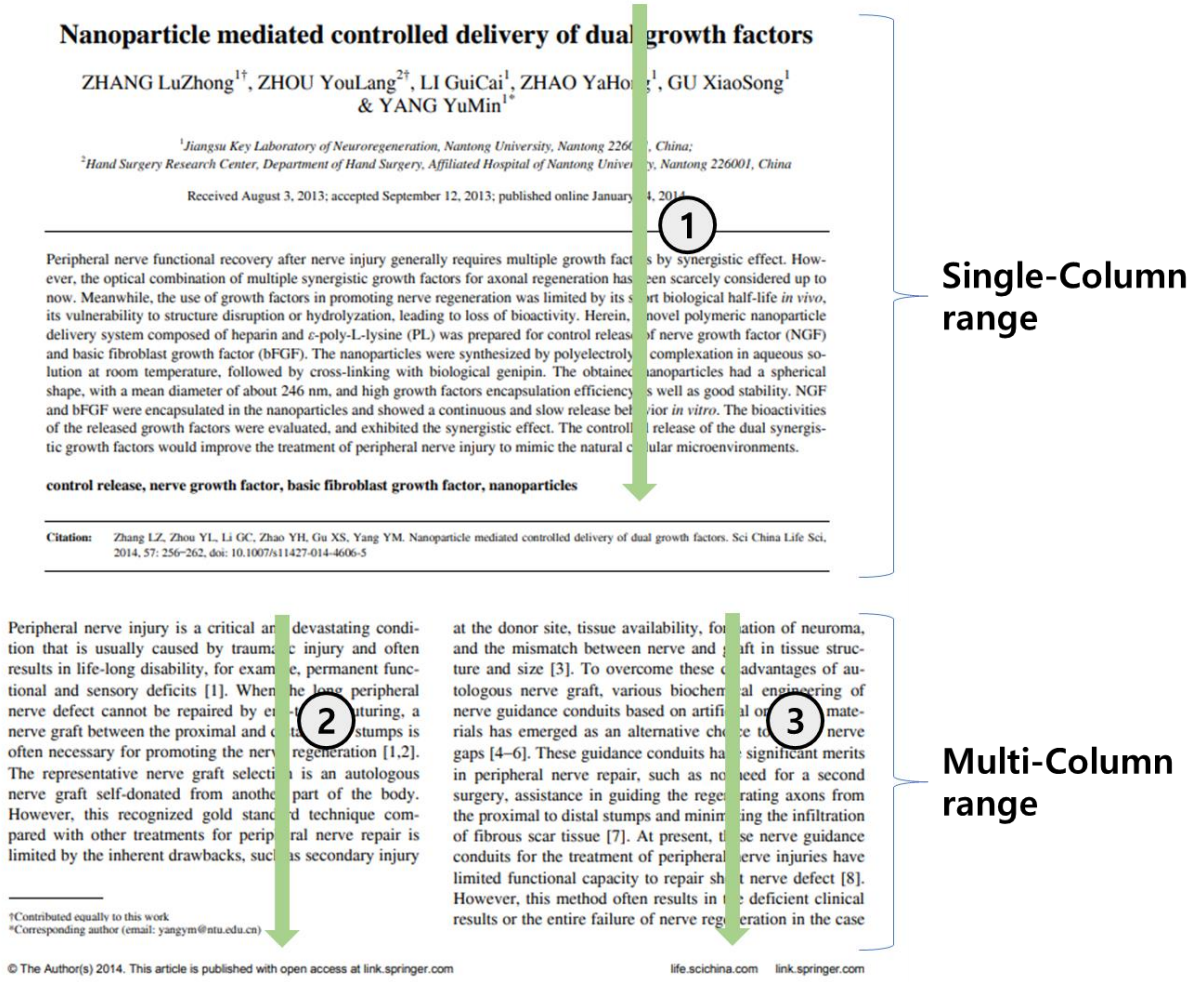


Figure 13. Graphical illustration of the paragraph the ordering process: The circled numbers show ordering procedure. As DBMATE scans a document from top to bottom, if a paragraph is positioned in a single-column range, DBMATE orders paragraphs by vertical center point. However, if a paragraph is positioned in a multi-column range, DBMATE put paragraphs in the corresponding column in multi-column ranges first and orders paragraph in each column by vertical center point later.

The above rules reflect human intuition. In almost every scientific paper, the author writes documents from left to right and from top to bottom. When the author divides the paragraph in columns, he starts writing paper from the leftmost column to the rightmost column. Within the column, authors generally follow consistent conventions of typesetting throughout.

3.5 Section Classification

After group ordering step, DBMATE classifier makes an analysis on every grouped span and put each group in four corresponding categories based on location information, representative font size, length of group, area of the group and its text.

1. **Title:** Title section has a larger font size and poses at the beginning of the page. Because the article's publication name has similar features, a pre-collected article or journal name is excluded from title candidates that DBMATE has found. By scoring using these two features with different weights, one final title is selected.

2. Author: Author section has a larger font size and poses at the beginning of the page. Also, the author section has several person's names, so the name entity recognition method is used to separate from the title section and author information section that usually posed later than the author section. Name entity recognition method, it accommodated by Spacy (Mattew et al., 2015), a tool that provides the category of the given word. By scoring using these three features, one final author is selected.

3. Abstract: The abstract has mainly mode font size, range of area size of the paragraph, poses at the beginning of the paper. It is usually shown as the first text paragraph. By scoring using these three features, one final abstract is selected.

4. Body: In this thesis, the body means the remaining part of the paper except for title, author, and abstract section. Hence, the location of the body section is behind the above three sections. To utilize these characteristics, the body section is analyzed after the above three sections are classified. Plus, the Body section consists of multiple sub-section that usually have the subtitle and its paragraph, such as Introduction, Methodology, Result, Discussion, Reference, and so on. Some paper starts the paragraph without the subtitle at the beginning of body section. In DBMATE, this non-subtitled paragraph is defined ‘pre-test’ to separate from the paragraph that has subtitles. Because the body section consists of multiple subtitles and their own paragraph, subtitle extraction is the main task in body section classification. The subtitle has a small area size and shorter length compare to its paragraph. However, header and footer, small fragment of equation, figure or table group have similar characteristics, so these noisy data are removed from the subtitle candidates. Once subtitles are extracted, their paragraphs are matched. DBMATE assigns a series of groups to the subtitle until it finds the next subtitle.

3.6 Hyperparameter Optimization

DBMATE finds different sets of hyperparameters in DBSCAN for different section classification. As shown in Figure. 1, DBMATE measures the accuracy of a hyperparameter for every iteration and observes the accuracy trends on hyperparameter variation. Once DBMATE chooses the best parameter set for each section classification using the training set, DBMATE applies the hyperparameter to the test set.

The hyperparameter separation method for sections helps DBMATE improve overall accuracy. To be specific, each section has its own optimal *eps* value by its original font size variation. For example, the title and author section usually have bigger font sizes than other sections, therefore their clustering boundary is bigger.

By plugging different boundary values in the DBSCAN algorithm, DBMATE is able to tune internal classifiers differently in sections.

Chapter 4 - Experimental Design

4.1 Data Preparation and Ground Truthing

For input data collection, a set of PDF documents can be collected by two ways: one is collected by human efforts such as visiting the web site of a science journal and downloading PDF file in the domain of interest, and the other one is to collect data by the automated system, which is a so-called "*crawler*". It is obvious that the latter saves time and energy. To use the benefits of the automated system, I used a web crawler developed at the Laboratory for Knowledge Discovery in Databases of Kansas State University (KSU KDD Lab) for the purpose of gathering user-defined domain-specific PDF files. The crawler automates collecting documents in the user-defined domain by utilizing a set of seeds (e.g. URLs). The collected PDF files from a specific domain website are fed to the DBMATE.

To obtain the ground truth, which is the section labels, such as the title, author, additional author information, abstract, and the bodies of subsections (such as 'Introduction', 'Background',' Experimental Design or Method', 'Result', 'Discussion',' Acknowledgement', 'Reference', and etc.), I manually labeled the sections and the subsections to corresponding paragraph blocks in the document.

This ground truth for section classification is based on span unit, therefore the ground truth is described as holistic information of span information, such as font size, font type, the spatial location of the bounding box, text, page number, and etc. To describe ground truth with span unit, I converted a raw document into metadata using PyMuPDF and I extracted spans from the

whole metadata. Since these spans are indexed, I describe ground truth with only the numbers of indexes.

Figure 14. shows the illustration of the cropped ground truth of a document. For example, the ground truth of title section is described as span indexes of ‘1,’ and ‘2,’ and stored in the text format. The file of the ground truth contains noisy data information such as header, footer, email address, publication date and so on as well as ‘Title’, ‘Author’, ‘Abstract’,’ Body’. This is because, in the future, this noisy data information is also analyzed in the system and used to help to filter out useless information for the accurate section classification. However, in the current research, this text file is sorted and structured as ‘DBMATE-specific ground truth’. This ground truth only contains the section information of ‘Title, ‘Author’,’ Abstract’,’ body’ as expressed by ‘span index’, therefore Levenshtein distance between the lists of span index are measured to evaluate the system performance.

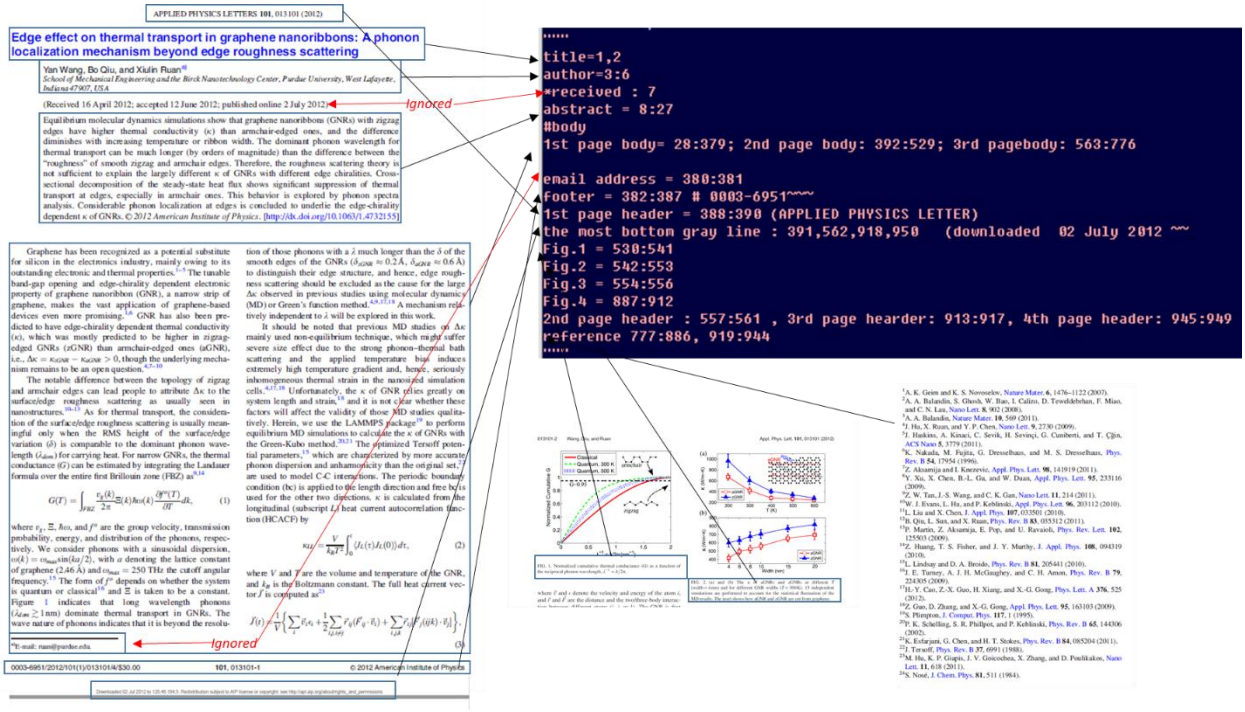


Figure 14. Example of Ground Truthing. I manually classified the sections of ‘Title’, ‘Author’, ‘Abstract’, ‘Body’ and other noisy data of ‘Header and Footer’, ‘Publication reception data’, ‘Email address’ and etc. Classified section information is stored as a ‘span index’

4.2 Evaluation Metrics

Our work aims to classify sections; however, the ultimate purpose of DBMATE is to generate request-specific section paragraph, which is an input of the following pipeline in QA system architecture. Therefore, we calculate distance metrics for their section paragraphs as well as annotated sections, which are an output of DBMATE. The output of DBMATE consists of the title, author, abstract, pre-text and four subtitle sub-section. The overall performance of the DBMATE is considered by the average performance of the section classification, such as title, author, abstract and body section.

Regarding body section accuracy measure, this thesis presents a strict method on body evaluation since I emphasize the importance of the extraction of an information-bearing paragraph. For this purpose, the performance of each sub-section is calculated only if the subtitle of DBMATE is the same as ground truth, otherwise, its performance is not calculated. To be specific, when the classified subtitle of DBMATE is not in the subtitle list of ground truth, performance for the subtitle is calculated with zero. In the same way, the performance for the text paragraph is measured when the classified subtitle of DBMATE is in the subtitle list of ground truth.

Figure 16 illustrates an example of an accuracy measure. Title and abstract section are scored of each 100% while author section is scored of 80% since one of five spans is missing. It is shown that the body section of ground truth consists of four sub-section, one is from pre-test and the other three are from titled sub-section. Therefore, average score of one pre-test and three

titled sub-section is considered as the score of the whole body section. Finally, the total score of a document classification is calculated by the average of section scores.

NANO LETTERS

LETTER

pubs.acs.org/NanoLett

Porous One-Dimensional Nanostructures through Confined Cooperative Self-Assembly

Feng Bai,[†,▲] Zaicheng Sun,[†,§] Huimeng Wu,[∥] Raid E. Haddad,[⊥] Eric N. Coker,[∥] Jian Yu Huang,[∥] Mark A. Rodriguez,[∥] and Hongyou Fan*[,†,∥]

[†]Key Laboratory for Special Functional Materials of the Ministry of Education, Henan University, Kaifeng 475004, People's Republic of China

[⊥]Department of Chemical and Nuclear Engineering, The University of New Mexico/NSF Center for Micro-Engineered Materials, Albuquerque, New Mexico 87131, United States

[§]State Key Laboratory of Luminescence and Applications, Changchun Institute of Optics, Fine Mechanics, and Physics, Chinese Academy of Sciences, 3888 East Nanhu Road, Changchun 130033, People's Republic of China

[∥]Advanced Materials Laboratory, Sandia National Laboratories, 1001 University Boulevard SE, Albuquerque, New Mexico 87106, United States

Supporting Information

ABSTRACT: We report a simple confined self-assembly process to synthesize nanoporous one-dimensional photoactive nanostructures. Through surfactant-assisted cooperative interactions (e.g., $\pi-\pi$ stacking, ligand coordination, and so forth) of the macrocyclic building block, zinc meso-tetra (4-pyridyl) porphyrin (ZnTPyP), self-assembled ZnTPyP nanowires and nanorods with controlled diameters and aspect ratios are prepared. Electron microscopy characterization in combination with X-ray diffraction and gas sorption experiments indicate that these materials exhibit stable single-crystalline and high surface area nanoporous frameworks with well-defined external morphology. Optical characterizations using UV−vis spectroscopy and fluorescence imaging and spectroscopy show enhanced collective optical properties over the individual chromophores (ZnTPyP), favorable for exciton formation and transport.

KEYWORDS: Hierarchical nanostructure, nanoporous, self-assembly, one-dimensional nanowire and nanorod, j-aggregate, porphyrin

Porous one-dimensional (1D) nanostructures, such as nanowires and nanorods, are materials of great interest for catalysis, sensing, and nanoelectronics due to their high surface areas and low densities.[1−3] Functionality of these nanomaterials is greatly expanded by incorporation of dye compounds, especially through J-aggregation of dye molecules that leads to strong excitonic coupling.[4] Continuous extensive efforts have been made to develop methods to fabricate 1D porous nanostructures,[1] such as templating methods,[5,6] hydrothermal synthesis,[7] noncovalent self-assembly,[8,9] and coordination modulation growth.[10] Despite these efforts, it is challenging to integrate multifunction into 1D nanostructures at multiple length scales and locations. Molecular self-assembly is one of the powerful methods to fabricate porous hierarchical nanostructures that impart multifunctionality derived from individual starting molecules or molecular building blocks.[3] Pore structures, morphologies, and function can be readily controlled by cooperative self-assembly through specific noncovalent interactions as well as variation of shape and size of starting molecules or building blocks. Harnessing these advantages, we report here a cooperative self-assembly process to synthesize highly ordered networks of 1D nanostructures that exhibit hierarchical structure and enhanced optical activity.

Porphyrins and related compounds, such as chlorophyll and heme, are essential to many biological energy transduction processes including light harvesting and photocatalytic processes.[11−13] Inspired by their utility and function, here we use their relative zinc meso-tetra (4-pyridyl) porphyrin (ZnTPyP) as our molecular building block. Whereas most porphyrins form H-aggregates as they typically self-assemble in a face-to-face manner via π-overlap,[14] ZnTPyP can self-assemble into more favorable arrangements via coordination of its peripheral pyridine groups to its core metal ions. ZnTPyP has well-defined size (~1.6 nm × 1.6 nm × 0.5 nm) and surface chemistry and is highly stable.

The first step in our confined self-assembly synthesis is encapsulation of ZnTPyP within the core of surfactant micelles through an interfacially driven aqueous process.[15] ZnTPyP does not readily dissolve in water, so it is acidified such that its pyridyl

Received: July 18, 2011
Revised: November 7, 2011
Published: November 15, 2011

ACS Publications © 2011 American Chemical Society

5196

dx.doi.org/10.1021/nl203598n | Nano Lett. 2011, 11, 5196–5200

```
# DBMETA Ground Truth (index type)
# file number is 114

# Main Section Ground Truth
       title=[13:14]
       author=[15:39]
       author_info=[40:55]
       abstract=[156:180]

# Body Section Ground Truth ( Total 5 parts)
       pre_text=[59:150,188:316,345:472,
                 528:630,663:674] # no subtitle
       subtitle_list=[675:676,684:685,
                 688:689,710:711] # 4 sub-titles

# Text paragraph for each subtitle element
       [677:683]
       [686:687]
       [690:709]
       [712:914]
# END Ground Truth
~
~
~
```

```
# DBMETA SPANLIST For file number :114

{'index': 1, 'text': u'Published:', 'cluster': 0, 'bbox': [51.9132
, 91.2165, 58.1042, 92.4525], 'font': 'AdvOT51c1769e', 'page': 1,
'size': 8.966300010681152}
{'index': 2, 'text': u'November 15, 2011', 'cluster': 1, 'bbox': [
59.464, 91.2165, 70.609, 92.4525], 'font': 'AdvOT2e364b11', 'page'
: 1, 'size': 8.966300010681152}
{'index': 3, 'text': u'r', 'cluster': 2, 'bbox': [25.2099, 94.6845
, 26.194, 95.3913], 'font': 'AdvEls-ent3', 'page': 1, 'size': 5.97
7700233459473}
{'index': 4, 'text': u' 2011 American Chemical Society', 'cluster'
: 2, 'bbox': [26.194, 94.583, 39.7602, 95.4864], 'font': 'AdvOT46d
cae81', 'page': 1, 'size': 5.97770023345947 3}
{'index': 5, 'text': u'5196', 'cluster': 3, 'bbox': [49.7945, 94.6
001, 52.4072, 95.5227], 'font': 'AdvMyrsemi_B', 'page': 1, 'size':
 7.471799850463867}
{'index': 6, 'text': u'dx.doi.org/10.1021/nl203598n', 'cluster': 4
, 'bbox': [65.5028, 94.6467, 77.7724, 95.4864], 'font': 'AdvOT46dc
ae81', 'page': 1, 'size': 5.977700233459473}
{'index': 7, 'text': u' |', 'cluster': 4, 'bbox': [77.7724, 94.429
, 78.3289, 95.676], 'font': 'AdvOT46dcae81', 'page': 1, 'size': 7.
970280061798096}
{'index': 8, 'text': u' Nano Lett.', 'cluster': 4, 'bbox': [78.328
9, 94.74, 82.6013, 95.4782], 'font': 'AdvMyriad_I', 'page': 1, 'si
ze': 5.977700233459473}
{'index': 9, 'text': u' 2011, 11, 5196', 'cluster': 4, 'bbox': [82
.6013, 94.6467, 88.7817, 95.4864], 'font': 'AdvOT46dcae81', 'page'
: 1, 'size': 5.977700233459473}
{'index': 10, 'text': u'\u2013', 'cluster': 4, 'bbox': [88.7807, 9
```

Figure 15. The left shows an example PDF document for cropping ground truth. The upper right shows an example of ground truth expressed with the span index. It shows title section, author section, abstract section, pre-text, subtitle list, and its text paragraph. The bottom right shows whole span list of a document that contains every span information

4.3 DBMATE vs MATESC

To compare DBMATE with MATESC, I applied the same distance metrics but only on text comparison, not on the holistic information of span. This is because the output of MATESC consists of only text information while DBMATE outputs the index list of spans as output format. To mitigate this gap, I extract the only text of DBMATE output and concatenates them in

a line. Again, the longest common subsequence (LCS) measure on text is used to measure the performance of both algorithms. The output of DBMATE has exclusive information such as location, font size and font type, respectively to output of MATESC.

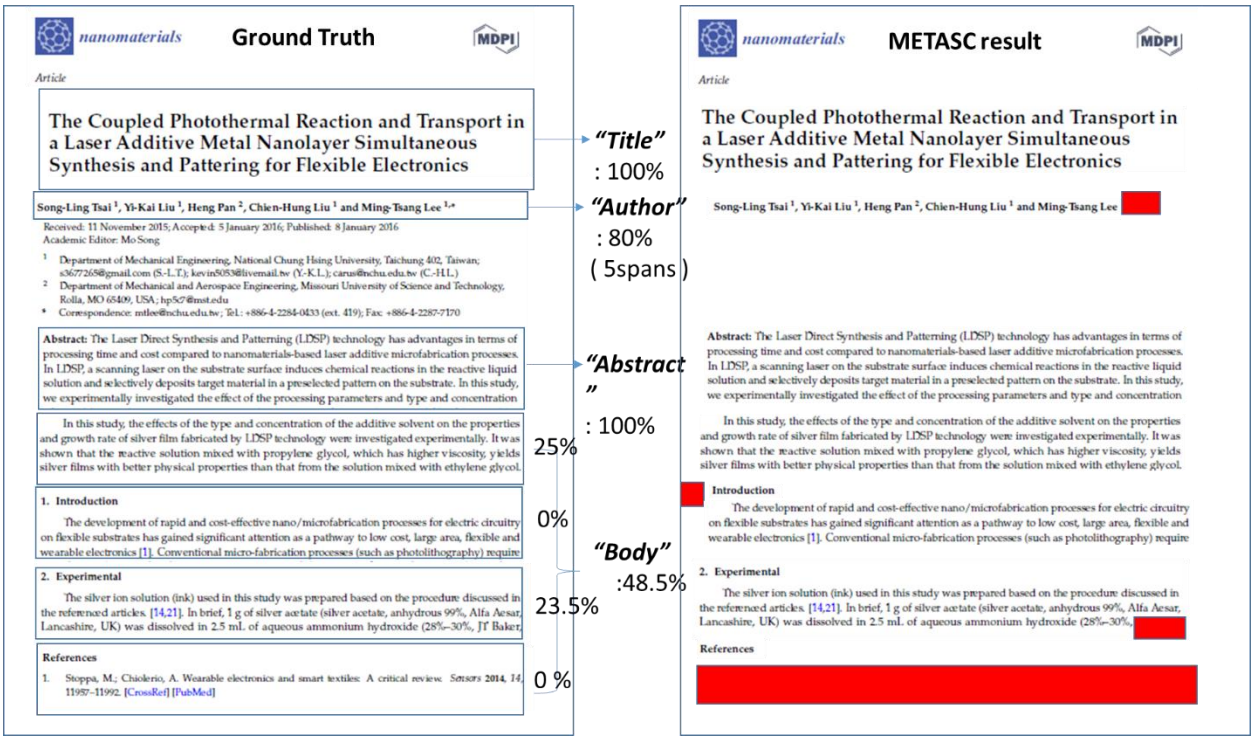


Figure 16. Example of the evaluation method for a document. The left shows ground truth while the right shows produced output of DBMATE. Here, for 'introduction' sub-subsection, the performance score is zero since one of the spans is missing.

4.4 Hybrid Algorithm

As I mentioned above, the ultimate purpose of DBMATE is to generate information-bearing paragraph. Hence, the direction of the research is to accurately detect subtitles and extract their corresponding body paragraphs. This chapter explains a hybrid algorithm that focuses on accurate body paragraph extraction. I have inspected MATESC first and presents DBMATE later. I found that both systems detect sections with a conservatively tight standard. Table2 shows that the body systems show a high precision score as shown in Table 2. For another

supportive material, Figure 21shows the accuracy difference between METASC and DBMATE on the difference between the number of subtitles of both algorithms. In Figure 21, the algorithm that finds the more subtitles shows the tendency to have a better performance result of body classification. I reflect this system characteristic on the new algorithm, so the hybrid algorithm takes one of MATESC and DBMATE if one algorithm shows more subtitles to the other algorithm.

For the scenario when both algorithms produce the same number of subtitles, the hybrid system takes DBMATE with higher priority. Support this strategy Figure 22 shows that DBMATE shows a better performance result when it produces the same number of subtitles as METASC does.

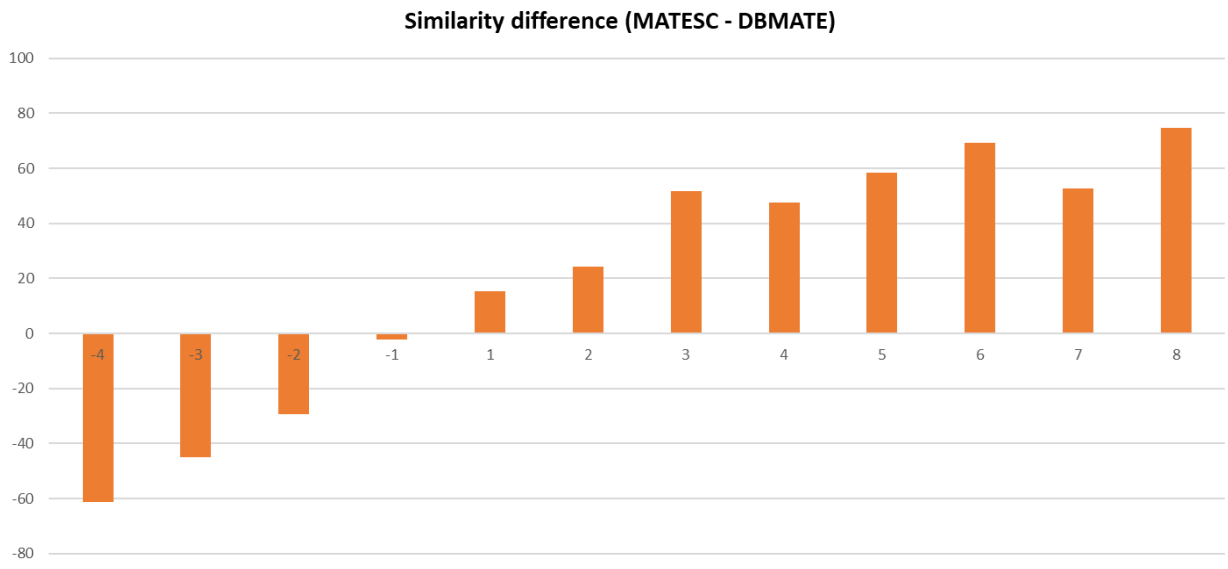


Figure 17 Accuracy trends (similarity) of the body section by subtitle candidates. In both algorithms (MATESC and DBMATE), accuracy tends to increase when an algorithm finds more subtitle candidates. The numbers in the horizontal axis represent the difference in the number of subtitles found in both algorithms.

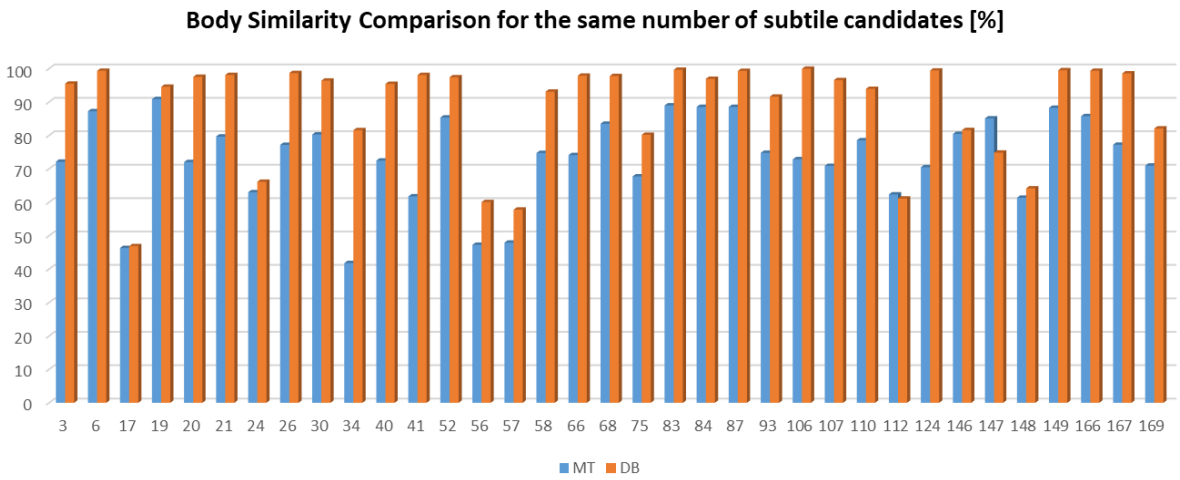


Figure 18 Similarity comparison of body section: DBMATE vs MATESC when both has the equal number of subtitle candidate. For every case of the same number of subtitle candidate, DBMATE show a better result than MATESC.

Chapter 5 - Results

This chapter mainly presents the results of experiment design described in chapter 4. First, the robustness of design is assessed using the results of cross validation experiments. Later actual accuracy of the proposed system which is compared with ground truth will be demonstrated. Finally, performance comparison with the most currently prominent method will be provided to show outstanding achievement.

5.1 Cross Validation

This section presents cross validation result for *eps* values. This cross validation is conducted on each section. Figure 17 - 20 shows cross validation result for each section, such as title, author, abstract and body. Title has *eps* value ranges from 0.4 to 0.6, author has *eps* value ranges from 0.3 to 0.5, abstract has *eps* value ranges from 0.5 to 0.7 and body has *eps* value ranges from 0.3 to 0.5. All *eps* ranges are small and similarity results shows small variation. We can see actual values in Table 1-4.

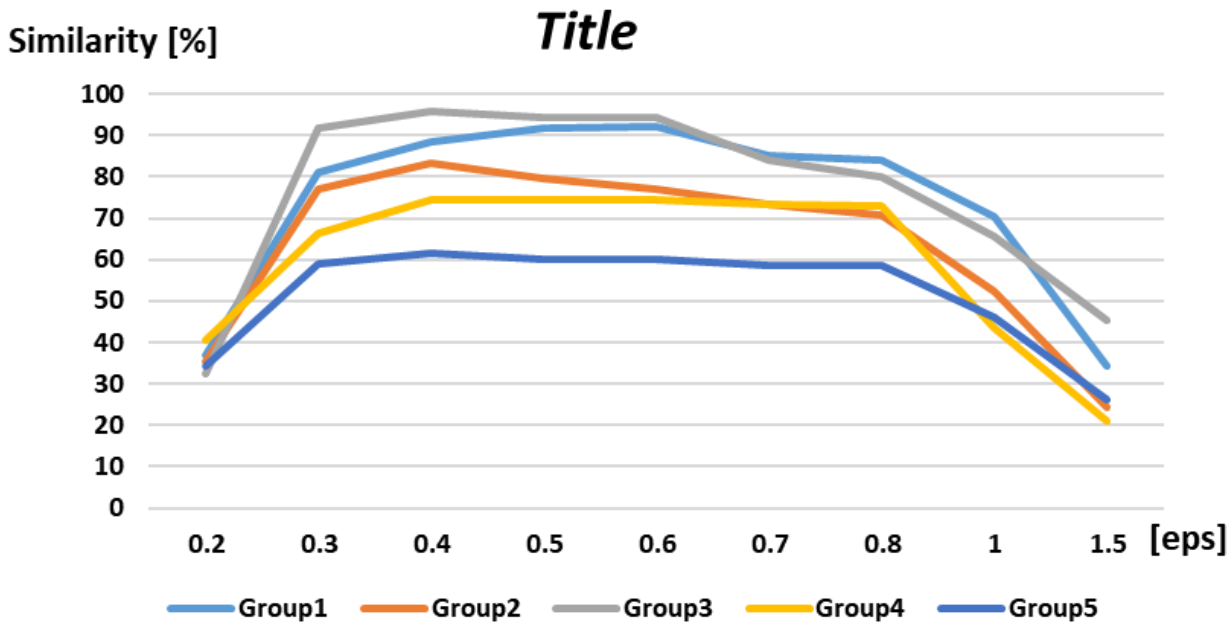


Figure 19. Cross validation result for *Title* section. The vertical axis represents the measured similarity [%] of the output of DBMATE and the horizontal axis represents *eps* values as the hyperparameter of DBSCAN.

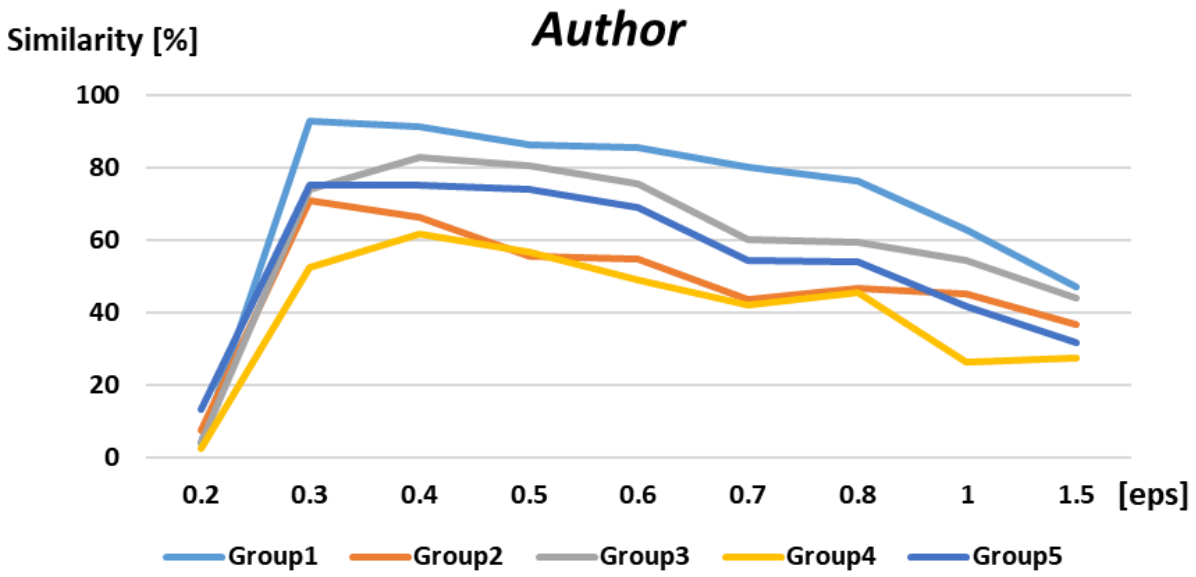


Figure 20. Cross validation result for Author section. The vertical axis represents the measured similarity [%] of the output of DBMATE and the horizontal axis represents *eps* values as the hyperparameter of DBSCAN.

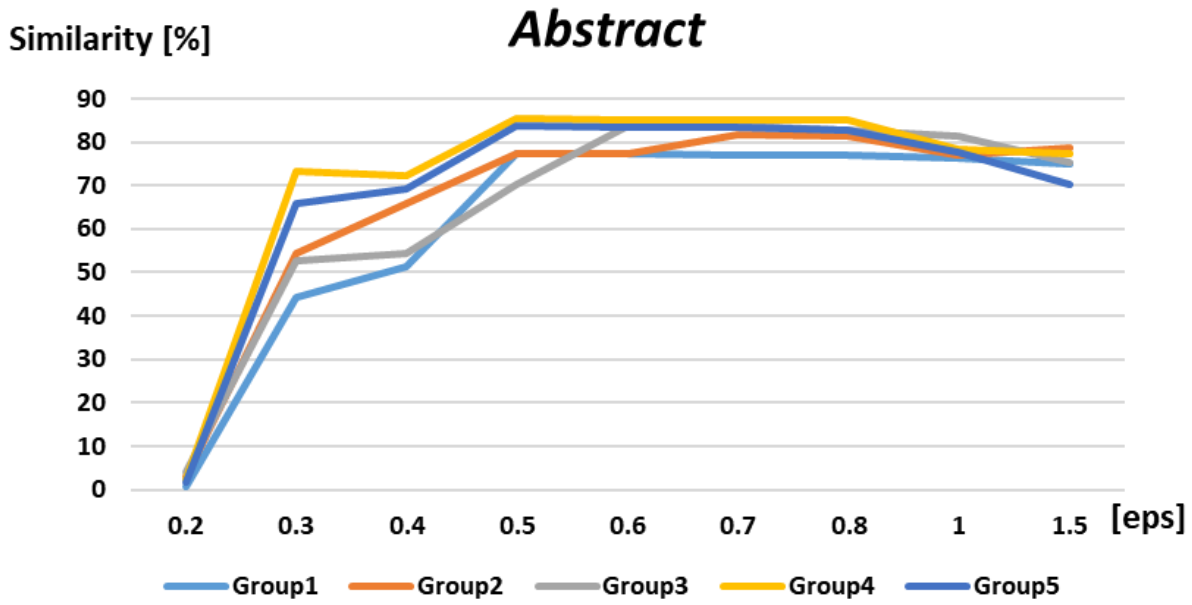


Figure 21. Cross validation result for Abstract section. The vertical axis represents the measured similarity [%] of the output of DBMATE and the horizontal axis represents *eps* values as the hyperparameter of DBSCAN.

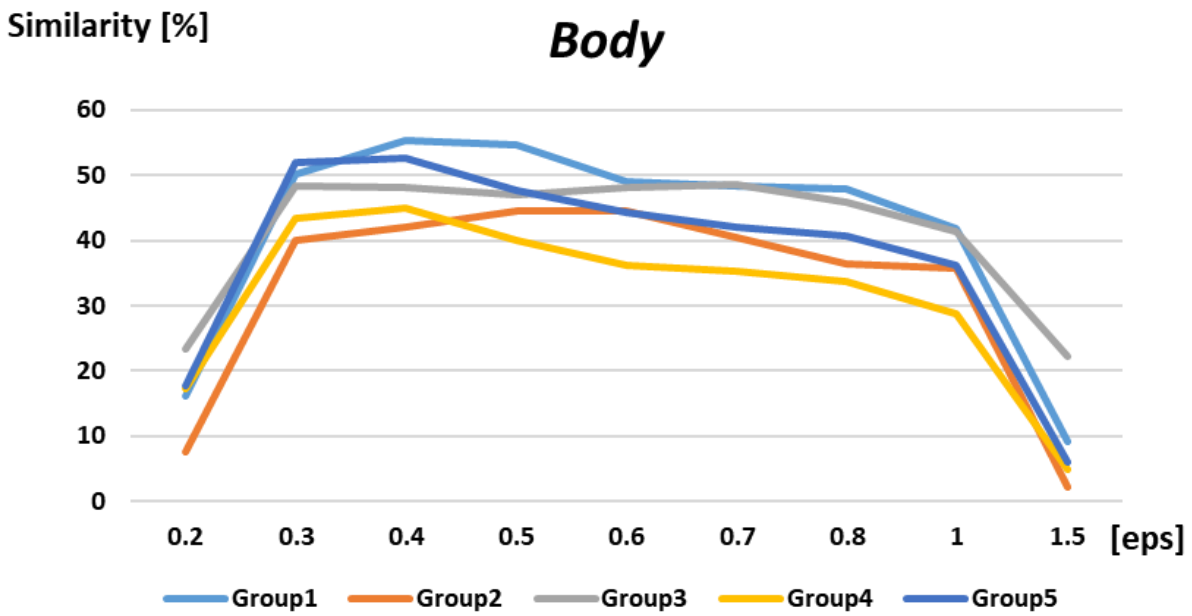


Figure 22. Cross validation result for Body section. The vertical axis represents the measured similarity [%] of the output of DBMATE and the horizontal axis represents *eps* values as the hyperparameter of DBSCAN.

Table 1. The actual value of the calculated similarity of Title section for cross validation of different *eps* values (columns). The shadowed values are marked as highest value.

Similarity [%]	0.2	0.3	0.4	0.5	0.6	0.7	0.8	1.0	1.5
Group1	739.24	1620.82	1770.82	1834.15	1841.85	1700	1677.78	1408.49	684.03
Group2	709.54	1543.68	1669.44	1594.44	1539.9	1468.18	1413	1047.07	486.72
Group3	645.56	1834.77	1913.24	1888.64	1888.64	1680.77	1596.56	1314.84	905.65
Group4	807.33	1324.44	1488.32	1488.32	1488.32	1469.17	1456.67	871.17	421.77
Group5	684.29	1178.05	1228.05	1203.05	1203.05	1173.81	1173.81	919.05	523.41

Table 2. The actual value of the calculated similarity of Author section for cross validation of different *eps* values (columns). The shadowed values are marked as highest value.

Similarity [%]	0.2	0.3	0.4	0.5	0.6	0.7	0.8	1.0	1.5
Group1	79.24	1860.75	1825.43	1725.43	1714.57	1607.87	1530.09	1257.7	940.96
Group2	148.07	1417.48	1326.64	1111.83	1093.65	874.92	937.8	906.26	735.41
Group3	83.83	1480.14	1661.36	1611.36	1511.37	1202.94	1187.15	1093.46	885.23
Group4	51.4	1050.24	1238.84	1138.84	979.09	843.8	909.52	531.31	550.35
Group5	266.24	1501.95	1501.95	1480.52	1380.52	1091.19	1082.87	839.28	634.47

Table 3. The actual value of the calculated similarity of Abstract section for cross validation of different *eps* values (columns). The shadowed values are marked as highest value.

Similarity [%]	0.2	0.3	0.4	0.5	0.6	0.7	0.8	1.0	1.5
Group1	15.34	884.74	1026.1	1550.1	1550.1	1542.96	1542.96	1528.79	1496.9
Group2	60.16	1090.97	1319.01	1545.11	1545.11	1634	1630.97	1542.08	1574.06
Group3	84.21	1057.12	1086.99	1408.23	1676.92	1676.92	1656.77	1627.92	1507.49
Group4	46.98	1468.76	1444.44	1708.3	1700.3	1700.29	1700.29	1567.44	1548.33
Group5	35.83	1314.8	1383.69	1673.03	1667.31	1667.31	1657.1	1551.37	1402.85

Table 4. The actual value of the calculated similarity of Body section for cross validation of different *eps* values (columns). The shadowed values are marked as highest value**.**

Similarity [%]	0.2	0.3	0.4	0.5	0.6	0.7	0.8	1.0	1.5
Group1	324.23	1002.33	1106.3	1094.56	982.59	968.46	956.57	834.64	184.46
Group2	152.44	802.28	842.06	891.52	888.57	811.01	727.64	714.14	44.46
Group3	468.87	968.04	962.98	938.58	963.83	972.38	917.85	825.86	446.91
Group4	344.14	866.65	897.44	799.78	723.17	706.45	674.65	574.57	97.82
Group5	356.2	1040.12	1052.82	952.77	887.11	842.16	811.82	725.67	121.22

5.2 DBMATE vs. Ground Truth

To give the readers the overall picture of system results, the total similarity of sections is provided. Table 5 shows the result. DBMATE shows an accurate performance of title, author and section classification but less accurate performance on body classification. To compare the baseline, the result comparison with MATESC will be shown in the following section.

Table 5. The overall performance result: DBMATE

	Precision	**Recall**	**F1 Score**
Title	0.82	0.95	**0.83**
Author	0.71	0.76	**0.73**
Abstract	0.79	0.83	**0.81**
Body	0.86	0.58	0.64
Avg. Total	0.80	0.78	0.75

5.3 DBMATE vs. MATESC

As I mentioned above, text-based comparison method explained in chapter 4. 2 is used to evaluate and compare DBMATE. MATESC produced the output of the same data set.
Table 6 shows an comparison result. As we see in Table 6, DBMATE successfully improved section classification using machine learning algorithm, comparing to the heuristic algorithm approach of MATESC.

Table 6. The overall performance result: MATESC vs DBMATE.

	Precision		Recall		F1 Score	
Algorithm	MATESC	DBMATE	MATESC	DBMATE	MATESC	DBMATE
Title	0.44	0.93	0.61	0.88	0.46	0.88
Author	0.38	0.83	0.48	0.67	0.36	0.70
Abstract	0.71	0.85	0.81	0.91	0.72	0.84
Body	0.77	0.76	0.77	0.72	0.75	0.74
Avg. Total	0.58	**0.84**	0.67	**0.80**	0.57	**0.80**

5.4 Hybrid Method (DBMATE + MATESC)

I have devised an alternative method, the combined algorithm of DBMATE and MATESC, to improve the performance of body section classification. The new method chooses one of both algorithms by comparing the subtitle numbers in the middle of the process. Meanwhile, this new method chooses DBMATE when the number of subtitles are the same because DBMATE has shown better performance at the same number of subtitles. Figure 21 shows the comparison result when both algorithms find the same number of subtitles. Table 7 shows result comparison including a new method, which has 75.3% of accuracy of body section classification. The newly combined algorithm loses spatial information, font type, font size and page number information so the output only has text information.

Table 7. The overall performance result: DBMATE vs Hybrid algorithm

	Precision		Recall		F1 Score	
Algorithm	DBMATE	Hybrid	DBMATE	Hybrid	DBMATE	Hybrid
Title	0.93	0.93	0.88	0.88	0.88	0.88
Author	0.83	0.83	0.67	0.67	0.70	0.70
Abstract	0.85	0.85	0.91	0.91	0.84	0.84
Body	0.76	**0.83**	0.72	**0.80**	0.74	**0.81**
Avg. Total	0.84	**0.86**	0.80	**0.82**	0.80	**0.81**

Chapter 6 - Conclusion and Future Work

6.1 Summary and Conclusion

I proposed a semi-supervised clustering framework for the payload extraction of documents, which consists of grouping and classification algorithm. Since our approach starts converting pixel information into metadata then merging into span and grouping into sections, the overall system shows greatly reduced training time. For training on 100 documents, it takes 4.1 hours which compares favorably with a deep neural network approach. Also, our system outputs multidimensional information of layout units of the document, so the system gives versatile output for end-users. We found that this advantage contributes to building the end-to-end system of various types of QA specifications.

The preliminary result on the section classification task shows approximately 20% increased F1 score compare to our baseline method. This result tells that DBSCAN as a grouping algorithm and data-driven approach for section annotation contribute to producing more an accurate payload extractor than the current heuristic method. We also provide a hybrid algorithm to improve accuracy on text information attributed to also statistical trends on the by-product of the system. Because a hybrid algorithm achieved improved accuracy in extracting the body paragraphs of subsections, it is expected that the system can be preferably used to QA system, which takes a specific body part as an input of the system.

6.2 Future Work

Several techniques in each pipeline in DBMATE can be studied and applied. Robust and accurate metadata extraction is an urgent task to proceed with the next step. In Figure 23, several alternatives such as PDFTextStream, Pdftotext or PDFBOX are enumerated in future work chart. The coverage of metadata extraction is to be evaluated and compared. The evaluation criteria include below:

1. How many kinds of layout does metadata cover? Does it contain figures, tables, logo information?

2. How accurately does metadata describe layout? Does it provide an accurate boundary box?

Because correctness and coverage of metadata decide the initial quality of layout analysis, accurate and broad coverage of metadata extraction is required. In the proposed research, only PyMuPDF was evaluated and used so other metadata extraction can be considered in future work.

Additional recognition algorithms for publication logo, table, figure or page index have been considered to build a concrete system. In DBMATE, publication logo, page index and picture information of figures are ignored due to the lack of metadata information or increased complexity. Larger coverage of recognition and detection will help to extract pure information, no contaminated by noisy data.

Neural network approaches are likely to improve DBMATE heuristics on noisy data recognition and section recognition. This technique requires a more detailed classification of ground truth. However, it is expected that neural network techniques such as RCNN, will enable significate additional improvements in the precision and recall of payload extraction.

Also, the natural language process can be applied to connect groups of sections in a semantic way. In the group ordering process, DBMATE utilizes only column information to connect groups in the same section. There has been shown that DBMATE incorrectly combines groups in order, so the sentences are broken by the incorrect ordering process. Analysis on semantic association between groups is expected to provide a understanding of the document and is a human-like ordering process, which is more convincing.

Evaluation criteria can be considered in a broad range. In this thesis, in the overall evaluation, the body extraction part is equally measured as other main sections were done. Various combinations of weights should be parametrically evaluated and fine-tuned to the application domain in developing a general and robust system to users.

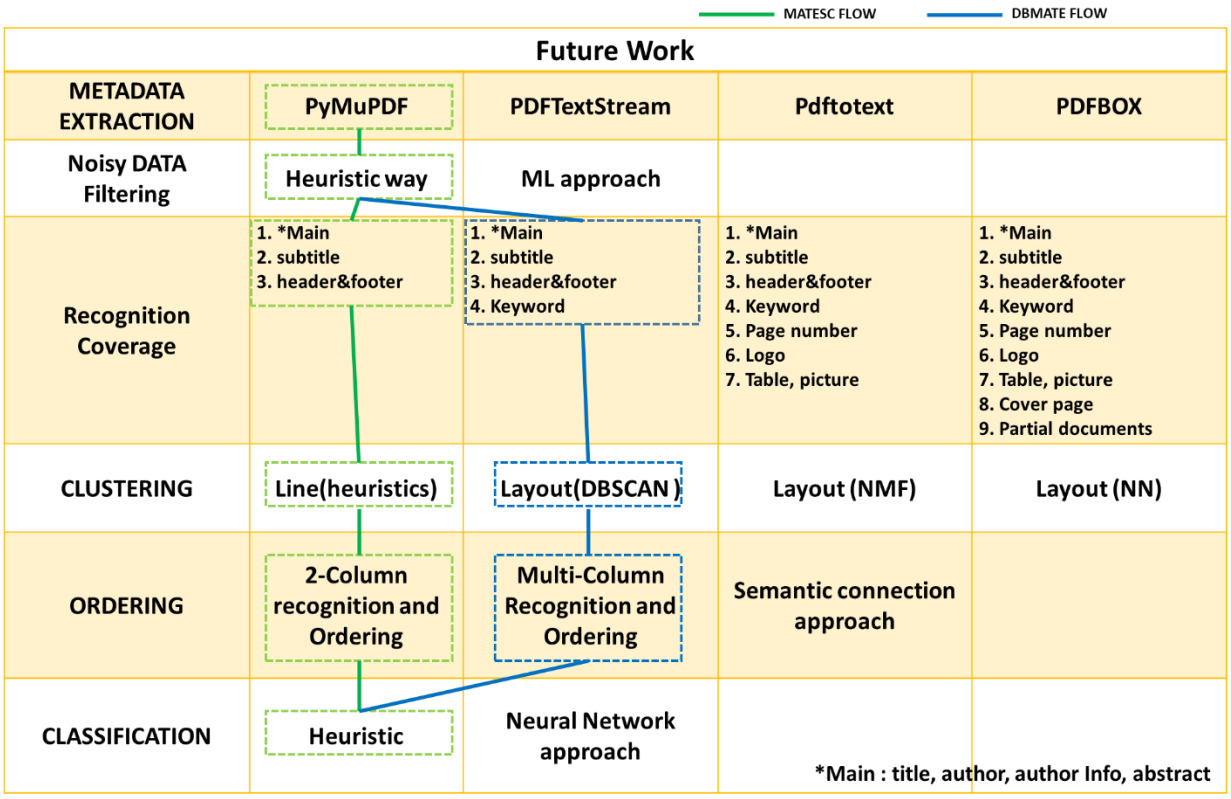


Figure 23. The coverage chart of the possible method in the pipeline of payload extraction. For coverage comparison, both two algorithm flows are represented in colored lines.

Bibliography

Aguirre, Carlos A et al. 2017. “Learning to Filter Documents for Information Extraction Using Rapid Annotation.” In IEEE, 85–90.

———. 2018. “Towards Faster Annotation Interfaces for Learning to Filter in Information Extraction and Search.”

Al Chalabi, Hani Maluf, Santosh Kumar Ray, and Khaled Shaalan. 2015. “Question Classification for Arabic Question Answering Systems.” In IEEE, 310–13.

Baudiš, Petr, and Jan Šedivý. 2015. “Modeling of the Question Answering Task in the Yodaqa System.” In Springer, 222–28.

Cheng, Xiang, Shuguang Zhu, Sen Su, and Gang Chen. 2017. “A Multi-Objective Optimization Approach for Question Routing in Community Question Answering Services.” *IEEE Transactions on Knowledge and Data Engineering* 29(9): 1779–92.

Clausner, Christian, Stefan Pletschacher, and Apostolos Antonacopoulos. 2011. “Aletheia-an Advanced Document Layout and Text Ground-Truthing System for Production Environments.” In IEEE, 48–52.

Cui, Xiaohui, and Thomas E Potok. 2005. “Document Clustering Analysis Based on Hybrid PSO+ K-Means Algorithm.” *Journal of Computer Sciences (special issue)* 27: 33.

Ester, Martin, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.” In , 226–31.

Fan, Kuo-Chin, Yuan-Kai Wang, and Mei-Lin Chang. 2001. “Form Document Identification Using Line Structure Based Features.” In IEEE, 704–8.

Fu, Chaogang. 2019. “Tracking User-Role Evolution via Topic Modeling in Community Question Answering.” *Information Processing & Management* 56(6): 102075.

Han, Jiawei, Jian Pei, and Micheline Kamber. 2011. *Data Mining: Concepts and Techniques*. Elsevier.

Hao, Xiaolong, Jason TL Wang, and Peter A Ng. 1993. “Nested Segmentation: An Approach for Layout Analysis in Document Classification.” In IEEE, 319–22.

Jose, Jeffin Mariam, and Jinu Thomas. 2018. “Finding Best Answer in Community Question Answering Sites: A Review.” In IEEE, 1–5.

Kadwe, Snehal S, and Shrikant Ardhapurkar. 2017. “Implementation of PDF Crawler Using Boolean Inverted Index and N-Gram Model.” In IEEE, 2680–83.

Kumar, KB Shiva, KB Raja, RK Chhotaray, and Sabyasachi Pattnaik. 2011. “Steganography Based on Payload Transformation.” *International Journal of Computer Science Issues (IJCSI)* 8(2): 241.

Lam, Stephen WK. 1994. “A Local-to-Global Approach to Complex Document Layout Analysis.” In , 431–34.

Lopez, Patrice. 2009. “GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications.” In Springer, 473–74.

Mahajan, Rutal S, and Mukesh A Zaver. 2018. “Novel Answer Ranking Approach in Question Answering System Using Compositional Distributional Semantic Model.” In IEEE, 1–5.

Maria, F, Carlos A Aguirre, BreAnn Anshutz, and William H Hsu. 2018. “MATESC: Metadata-Analytic Text Extractor and Section Classifier for PDF Scientific Publications.”

Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. MIT press.

Pham, Son The, and Dang Tuan Nguyen. 2014. “Implementation Method of Answering Engine for Vietnamese Questions in Reading Answering System Model (RASM).” In IEEE, 175–80.

Rusinol, Marcal, David Aldavert, Ricardo Toledo, and Josep Llados. 2011. “Browsing Heterogeneous Document Collections by a Segmentation-Free Word Spotting Method.” In IEEE, 63–67.

Shao, Taihua, Yupu Guo, Honghui Chen, and Zepeng Hao. 2019. “Transformer-Based Neural Network for Answer Selection in Question Answering.” *IEEE Access* 7: 26146–56.

Shekhar, Shashi, Rohit Agrawal, and Karm Veer Arya. 2010. “An Architectural Framework of a Crawler for Retrieving Highly Relevant Web Documents by Filtering Replicated Web Collections.” In IEEE, 29–33.

Soares, Marco Antonio Calijorne, and Fernando Silva Parreiras. 2018. “A Literature Review on Question Answering Techniques, Paradigms and Systems.” *Journal of King Saud University-Computer and Information Sciences*.

Sun, Yuan, and Tianci Xia. 2019. “A Hybrid Network Model for Tibetan Question Answering.” *IEEE Access* 7: 52769–77.

Te, Rou. 2011. “Research on Question Classification Method of Tibetan Online Automatic Question-Answering System.” In IEEE, 211–13.

Thoma, George R. 1999. “Automating Data Entry for an Online Biomedical Database: A Document Image Analysis Application.” In IEEE, 370–73.

Zhao, Ying, and George Karypis. 2001. “Criterion Functions for Document Clustering: Experiments and Analysis.”