Automatic Extraction of Main Thesis Documents Fields Using Decision Trees

Alaa Mahmoud Sobhy
College of Computing and Information Technology
Arab Academy for Science Technology and Maritime Transport
Cairo, Egypt
alaa.mahmoud.sobhy@gmail.com

Yasser M. Kamal
College of Computing and Information Technology
Arab Academy for Science Technology and Maritime Transport
Cairo, Egypt
dr_yaser_omar@yahoo.com

Atef Zaki Ghalwash
Computer Science Department
Helwan University
Cairo, Egypt
atef_ghalwash@yahoo.com

***Abstract*— Thesis documents are underestimated even though they hold large sets of useful information –as they include most of the research information–, but since they are harder to obtain, researchers were lead to depend on research papers even though they have a size limitation and lack elaboration. A lot of time and effort are invested in research, so having a linkage among researchers based on their work would somehow facilitate solving the research problem process. A major step to tackle this goal is to structure thesis documents by extracting some fields such as title, author and abstract. This paper presents a way to structure a semi-structured thesis documents using decision trees in 4 different ways (Simple, Medium, Complex and using KNIME), they scored an overall accuracy of 99.2%.**

Keywords—Structured Data; Semi-structured Data; Thesis Documents; Decision Trees; Machine Learning

I. INTRODUCTION

Unstructured data forms a descent volume compared to structured data. Unstructured data, such as webpages, images and documents, may contain useful information, yet because they lack clear structure and format it is harder to be acquired. Structured data is characterized by having a unified form; an example of structured data would be databases. A database being structured allows information to be easily retrieved and allows the manipulation of the database itself.

Unstructured and semi-structured data are vastly growing in size annually. Projects done by Corporates such as EMC and IDC predict that data size will reach 40 zettabytes by 2020 [1]. Extracting knowledge from such data is highly recommended to enrich the decision making process. However in order to make the data understandable, according to each field in business, the data has to go through a process; extracting information from data automatically in such a way to make it suitable for business use. This is generally achieved by automating the conversion of unstructured or semi-structured data to a structured form.

Researchers face a lot of problems through their research. When this happens some researchers resort to reading other people's work such as research papers, doing such a thing is helpful and gets the researcher an insight on what they should be doing. However not every problem could be solved by reading papers around the same research topic, sometimes a deeper reading would be helpful. Starting from previous discussions, we would like to find a way to describe a researcher's work thoroughly and what better way to describe it, than the researcher's thesis document. The paperless solution should consider having a great deal of details that describe thesis contents, and consider solving faced problems such as structuring. Finding thesis documents for someone's work is harder than finding a paper published with the researchers' work that's why thesis documents are often ignored. In the present paper some thesis documents were gathered, from different sources, to have some structure based on a variety of contents.

Documents used in this research are considered semi-structured because of their similarities, even though some of them may vary in formats and order. The Structuring of thesis documents is done using decision trees in order to make them easier to retrieve data from. Having structured thesis documents would make it easier for researchers who may need more information to find and use the knowledge.

Thesis documents used as datasets were gathered from handed over CDs at the Arab Academy for Science Technology and Maritime Transport, College of Computing and Information Technology, and the digital library [2] of The American University in Cairo, School of Science and Engineering. They were all MSc degrees varying in content, some were masters in computer science and others were masters in information systems.

The data set goes through a preprocessing phase that includes some calculations, statistics and finding some word occurrences. The data is divided into two groups the first group would be used for training a decision tree and creating the machine learning model, the other part of the data would be used for testing these models and comparing between a previously predicted supervised model and the machine learning predicted model.

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.164

This paper is divided into 5 sections. In section II some related work, in structuring different types of documents and using different types of methods, will be discussed, Section III is the proposed method for structuring thesis documents, Section IV is a representation of the experimental results, and finally in Section V conclusion and future work are discussed .

II. Related Work

Structuring documents was done using a variety of methods, such in [3] and [4], on a variety of documents such as XML Documents as in [5] and [6], document images Ref. [7], and PDF documents [8].

One method presented by Ranganathan at [9] was to use GATE to structure the semi structured Word Documents and Excel Sheets but first the document is converted to XML first. GATE works with documents by converting them to plain text then annotating it. GATE is not able to refer back to the original document hence converting to XML as GATE can work with XML but not Word and Excel directly.

Rusu et al. [10] Extracting knowledge from databases KDD (Knowledge Discovery in Databases) aims to find useful information in data, it works in two ways either defining descriptive approximation for the generated data, or creating a predictive model for estimating the value of future cases. The Data mining is done through analysis and discovery algorithms. In order to deal with unstructured data the paper followed some steps. Firstly they extract data from its current form for example if it's in a webpage or an image. As for HTML documents they get formatted into XML. Secondly the extracted data goes into a syntactic analysis phase which generates a parse tree for each sentence dividing it into subject verb phrase. Thirdly a classification algorithm is used to classify and categorize the data.

Zhang [11] used medical documents to be clustered using ontology based term similarity measures, first by indexing terms in medical documents, the ontology based term similarity measure calculated the weight of terms, then the clustering was done by spherical L-means.

Huang [12] introduced text categorization technique called VSM_WN_TM which is a combination of Vector Space Model (VSM), WordNet ontology, and Probabilistic Latent Semantic Analysis (PLSA) topic modeling. It also used the support vector machine for classification purposes. The technique started by creating a VSM model for a document, and based on observation they created a weighing scheme, afterwards they used WordNet with VSM to categorize text, then they incorporated VSM with PLSA to choose topics from documents, and then finally it generated a hybrid VSM model for classification.

III. Proposed Method

A. Dataset and Preprocessing

Total number of gathered thesis documents is 65; which were divided into a group of 36 thesis documents to be used for training the classifier and the remaining 29 were used for testing and prediction. The total number of pages for all theses is 8,346 pages, averaging 128.4 pages per thesis. Most thesis documents were originally in PDF format and some were Word, for the sake of unification, the documents were all converted into Word Documents.

B. Features

Each thesis document has a catalogue of features that was extracted to describe it. Features extraction was done by identifying global features that must exist in each thesis document regardless of them having a standard format or not i.e. (semi-structured or not structured). For each thesis 14 features were extracted per page. Features were stored as a matrix; columns represent features while rows represent the value of each feature per page. Features were extracted using a C# code and NetOffice assemblies for accessing Word documents.

Each category of features was chosen separately. At the beginning the interest was finding where each part of the thesis was, the concept was similar to bookmarks, which is why all extracted features are related to a page number. The first category was the positions of certain words in the documents; these words must exist in almost all thesis documents as a subtitle to describe its contents. The position of these words reflected all the occurrences of the word itself regardless of its actual location. That led to the second category; it is necessary to know the relation between the word and the next few pages, if they were related or not. Which would clarify, on average, pages that are connected to each other; this was done for the important words not for the whole document. Furthermore a column to describe whether this page has a subtitle or not, the column would make it easier to get the actual occurrence of the word and ignore it if it's just mentioned randomly. The final category was added to try some general statistics and test if the count of paragraphs, words and digits would be of help in the structuring process, or would it be ignored by the machine learning system.

Extracted features, as previously mentioned, were divided into three categories of features, as shown in fig. 1, the first feature represents page number's count for each thesis document, and the page number refers to the count in Word regardless of the numbering done by the author.

The first category of features represents the occurrences of some chosen words, it is described by either 0 if the feature doesn't exist in the page or 1 if the feature does exist, the redundancy of the word in the page is ignored. In the first category the selected words were, "thesis" which was chosen to focus on its first mentioning in the document, the first occurrence of the thesis word represents the first page contents which contain important data such as (title, year, author,… etc.), other words

positioning and occurrences were recorded: “supervisor”, “acknowledgement”, “Abstract”, “Contents”, “Conclusion” and “References/Bibliography”.

The second category of features consists of two columns; the first one is called positive proximity which is the distance of each word from the upcoming few pages. For example if the table of contents starts from page 10 through page 15 the positive proximity in page 10 would be 1, because it’s where table of contents begins the next page would be 2 and so on till we reach 5. This is the representation of the relationship between each subtitle and the next few pages that contain the content of that title. The second column is a true or false reflection of whether the page has a subtitle in it or not (1 is for true and 0 is for false). Furthermore if the subtitle exists in a page but it is not the actual subtitle content it would be recorded as a 0. The last category of features is the calculation of some statistics per page; these are the total number of paragraphs, the total number of digits and the total number of words.

C. Response Column

In machine learning a result needs to be specified to be able to train the classifier. A response column was created to describe each row of features. The last column is integers within the range of 1 to 7. Fig. 2 Phase two shows the outline of creating a response column or a result column. The integer 1 describes the existence of metadata in the page i.e. (Title, Author, Supervisor, and Year). 2 for abstract pages, 3 is the integer representing table of content pages, 4 represents acknowledgement pages, all of the previous are considered the first partition of the document, 5 refers to conclusion pages, 6 references or bibliography pages, 7 is for any other pages such as chapters, publications, declaration, dedication and appendices.

After those 3 steps an inverted file is complete for insertion into the decision tree for training and testing. Table I shows a sample of the inverted file.

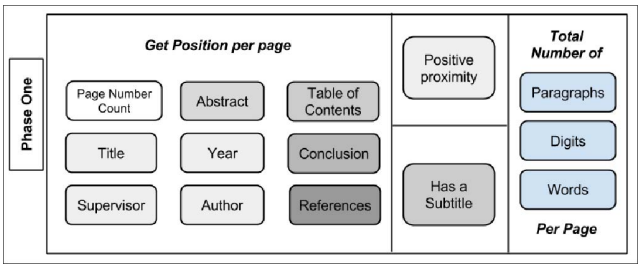

Fig. 1. Phase 1 Extracted Features from Thesis Documents

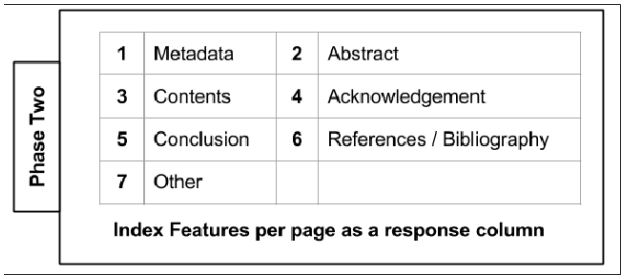

Fig. 2. Phase 2 Response Column

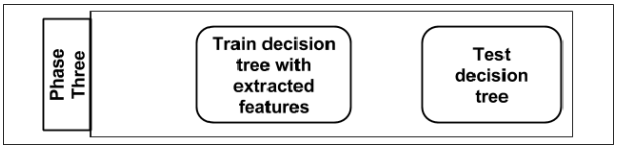

Fig. 3. Phase 3 Decision Tree

D. Decision Trees

Decision trees were chosen because initially a code was written to structure thesis documents in C#, the code ended up having a lot of “if and else” clauses which suggested, by default, having a binary decision tree due to all of the conditions that were written. Fig. 3 describes that extracted features go through training first then through testing the decision tree, which is the last phase. MATLAB was used for decision tree training and testing.

Total rows of features for training were a little over 5,000 while a about 3,200 rows of features were used for testing and validation. Cross validation were used to prevent overfitting, as the training data set was divided into randomly partitioned and almost same size folds which are called *k-folds*, the number of folds is 5. Each fold is used for training except one which is used for validating the trained subsets and this process is repeated 5 times [13].

Decision trees were applied with different tools KNIME was used at first and then MATLAB, and they were compared against each other. Decision trees got the most accurate results of supervised machine learning systems. MATLAB’s simple decision tree, a medium decision tree and a complex decision tree were tested. The difference between the three is the number of leaves in each tree, the more leaves there is the more accurate the class distinction.

IV. Experimental Results

A. Decision Trees

All trained models were tested several times with different dataset ordering and selection. The best results were presented by the medium decision tree; it recorded the most efficient and almost stable results of all models. The root feature that was used to decide the classification was the positive position feature. Fig. 4 is a view of the confusion matrix resulting of the trained tree. Algorithm 1 is the algorithm used to train the decision tree. It describes the

TABLE I. INVERTED FILE SAMPLE

Page Number	Thesis	Supervisor	Ack.	Abstract	TOC	Conc.	Ref.	Biblio.	Num. Parag.	Num. Digits	Num. Words	Proxim. Positive	Has Subtitle	Response
1	1	1	0	0	0	0	0	0	47	4	72	1	1	1
2	1	0	0	0	0	0	0	0	39	0	81	0	0	7
3	1	1	0	0	0	0	0	0	10	0	182	1	1	4
4	1	0	0	1	0	0	0	0	9	16	292	1	1	2
5	1	0	0	0	0	0	0	0	6	3	229	2	0	2
6	1	0	0	0	1	0	0	0	53	97	147	1	1	3

process of reading the dataset and training the decision tree, while algorithm 2 is the actual generated decision tree.

B. Testing and Validation

Trained models were used to predict new data, the testing data were a total of 29 theses with an average of 3,200 thesis document features. Predicted results by models were compared to a supervised observation.

Table II is a collection of the accuracy results given by each model, the most accurate model was given by the medium decision tree which scored an accuracy rate of 99.2% and an error rate of 0.8%.

Table III shows the confusion matrix of the predicted class and the actual classes. Table IV states some statistics of each class, there is a total of 7 classes. For each class the calculations done are True Positive (TP), False Positive (FN), False Negative (FN), True Negative (TN), Precision, Sensitivity and Specificity. The following Equations used to calculate statistics:

True Positive [TP] = Condition Present + Positive result (1)

False Positive [FP] = Condition absent + Positive result [Type I error](2)

False (invalid) Negative [FN] = Condition present + Negative result [Type II error](3)

True (accurate) Negative [TN] = Condition absent + Negative result (4)

Precision(class) = TP(class) / (TP(class) + FP(class))(5)

Sensitivity(class) = Recall(class) = TruePositiveRate(class) = TP(class) / (TP(class) + FN(class))(6) Specificity (mostly used in 2 class problems)=

TrueNegativeRate(class) = TN(class) / (TN(class) + FP(class))(7)

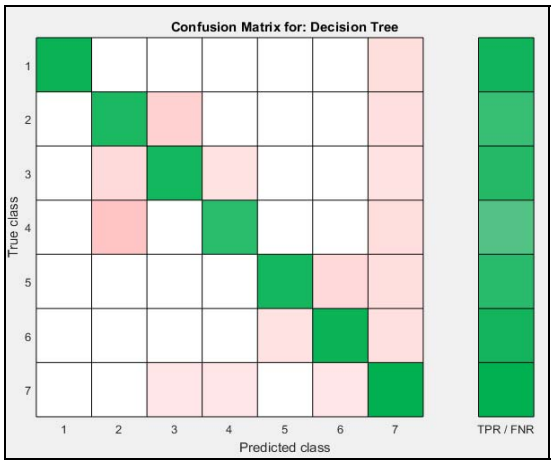

Fig. 4. Confusion Matrix of Decision Tree

TABLE II. RESULTS OF TESTING DIFFERENT MACHINE LEARNING TECHNIQUES AND TOOLS

Classifiers	Overall Accuracy
Simple Tree	96.97%
Medium Tree	99.20%
Complex Tree	99.17%
KNIME Decision Tree	99.14%

TABLE III. CONFUSION MATRIX OF ACTUAL CLASSES VERSUS PREDICTED CLASSES

		Actual Classes						
		1	*2*	*3*	*4*	*5*	*6*	*7*
Predicted Classes	*1*	22	0	0	0	0	0	0
	2	0	33	7	1	0	0	0
	3	0	2	55	0	0	0	1
	4	0	0	1	17	0	0	0
	5	0	0	0	0	112	12	0
	6	0	0	0	0	2	111	0
	7	0	0	0	0	0	1	2858

TABLE IV. STSTISTICS ON EACH CLASS

	Actual Classes						
	1	***2***	***3***	***4***	***5***	***6***	***7***
TP	22	33	55	17	112	111	2858
FP	0	8	2	1	12	2	1
FN	0	2	8	1	2	13	0
TN	3212	3191	3169	3215	3108	3108	375
Precision	100%	80%	96%	94%	90%	98%	100%
Sensitivity	100%	94%	87%	94%	98%	90%	100%
Specificity	99%	99%	99%	99%	99%	99%	99%

Algorithm 1 Train Decision Tree Classifier Algorithm

Input: Dataset to be trained *ThesisData*
O**utput**: Trained Classifier
1: *Data* ← Excel Dataset
2: Convert Data to Table *ThesisData* with header columns *ThesisData*←*PageNumber, ThesisWord, SuperviseWord, AckWord, AbstractWord, ContentsWord, ConclusionWord, ReferencesWord ,BibliographyWord, NumParagraphs, NumDigits, NumWords, PositionPositive, HasSubtitle, Result*
3: Define *Result* Type ← Categorical
4: Define *PredictorNames* ← *PageNumber, ThesisWord, SuperviseWord, AckWord, AbstractWord, ContentsWord, ConclusionWord, ReferencesWord, BibliographyWord, NumParagraphs, NumDigits, NumWords, PositionPositive, HasSubtitle*
5: Convert *PredictorNames* into a Table
6: Convert *PredictorNames* to double
7: *ResponseName* ← ThesisData Result column
8: *ClassNames*← from 1 to 7
9: *MaxNumSplits* ← 20
10: *SplitCriterion* ← gdi
11: *Surrogate* ← off
12: Train Classifier *TrainedDTreeMedium* ← fit *predictors, response, pridictorNames, ResponseName, ClassNames, SplitCriterion, MaxNumSplits, Surrogate*
13: *partitionedModel*← Partition model for cross-validation into 5 folds
14: Compute Validation Accuracy

Algorithm 2 Medium Decision Tree for Classification

0: if *PositionPositive*<0.5 then node 2 elseif *PositionPositive*>=0.5 then node 3 else 7
1: if *PageNumber*<1.5 then node 4 elseif *PageNumber*>=1.5 then node 5 else 7
2: if *PageNumber*<36.5 then node 6 elseif *PageNumber*>=36.5 then node 7 else 6
3: class = 1
4: if *AckWord*<0.5 then node 8 elseif *AckWord*>=0.5 then node 9 else 7
5: if *NumDigits*<24 then node 10 elseif *NumDigits*>=24 then node 11 else 3
6: if *NumDigits*<32 then node 12 elseif *NumDigits*>=32 then node 13 else 6
7: if *NumDigits*<151 then node 14 elseif *NumDigits*>=151 then node 15 else 7
8: class = 7
9: if *PageNumber*<1.5 then node 16 elseif *PageNumber*>=1.5 then node 17 else 2
10: class = 3
11: if *NumWords*<3 then node 18 elseif *NumWords*>=3 then node 19 else 5
12: if *ConclusionWord*<0.5 then node 20 elseif *ConclusionWord*>=0.5 then node 21 else 6
13: if *ConclusionWord*<0.5 then node 22 elseif *ConclusionWord*>=0.5 then node 23 else 7
14: if *NumParagraphs*<25.5 then node 24 elseif *NumParagraphs*>=25.5 then node 25 else 7
15: class = 1
16: if *AbstractWord*<0.5 then node 26 elseif *AbstractWord*>=0.5 then node 27 else 2
17: if *PositionPositive*<10 then node 28 elseif *PositionPositive*>=10 then node 29 else 6
18: if *PositionPositive*<7.5 then node 30 elseif *PositionPositive*>=7.5 then node 31 else 5
19: if *NumParagraphs*<4 then node 32 elseif *NumParagraphs*>=4 then node 33 else 6
20: class = 5
21: class = 7
22: class = 7
23: class = 6
24: class = 7
25: if *PositionPositive*<1.5 then node 34 elseif *PositionPositive*>=1.5 then node 35 else 4
26: class = 2
27: class = 6
28: class = 7
29: class = 5
30: if *PageNumber*<119.5 then node 36 elseif *PageNumber*>=119.5 then node 37 else 6
31: class = 5
32: class = 6
33: class = 4
34: class = 2
35: class = 6
36: class = 5

V. CONCLUSION AND FUTURE WORK

This paper contributes in structuring thesis documents to help researchers access knowledge easily. This was done by a machine learning technique which is a decision tree; different tools were used to train the decision tree also different types of decision trees were trained and all were compared to each other. The most accurate, efficient and low in variance model was picked which recorded in the validation and testing phase an accuracy of 99.2%.

For future work visual features could be added such as font size, tab spacing, alignment or some other statistical

features that could help with improving results and overall accuracy, also a spelling check would help in this matter to reduce the errors of extractions that were ignored because of misspelled words. Furthermore, extending accepted formats is essential such as PDFs.

REFERENCES

[1] Tomer Shiran, The Future of Hadoop: MapR VP of Product Management.2014.

[2] AUC Libraries, http://library.aucegypt.edu/.

[3] Poullet, Line, Jean-Marie Pinon, and Sylvie Calabretto. "Semantic structuring of documents." Information Technology, 1997. BIWIT'97., Proceedings of the Third Basque International Workshop on. IEEE, 1997.

[4] Gavrilova, Tatiana, and Irina Leshcheva. "Collective Ontologies Design and Development." Complex, Intelligent and Software Intensive Systems (CISIS), 2014 Eighth International Conference on. IEEE, 2014.

[5] Hwang, Jeong Hee, and Mi Sug Gu. "Clustering xml documents based on the weight of frequent structures." Convergence Information Technology, 2007. International Conference on. IEEE, 2007.

[6] Huang, Yin-Fu, and Po-Lun Liou. "Retrieving Representative Structures from XML Documents Using Clustering Techniques." Intelligence and Security Informatics Conference (EISIC), 2011 European. IEEE, 2011.

[7] Chen, Siyuan, et al. "Structured document classification by matching local salient features." Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, 2012.

[8] Hadjar, Karim, et al. "Xed: a new tool for extracting hidden structures from electronic documents." Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on. IEEE, 2004.

[9] Ranganathan, Girish R., Yevgan Biletskiy, and Alexey Kaltchenko. "Semantic annotation of semi-structured documents." Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on. IEEE, 2008.

[10] Rusu, Octavian, et al. "Converting unstructured and semi-structured data into knowledge." Roedunet International Conference (RoEduNet), 2013 11th. IEEE, 2013.

[11] Zhang, Xiaodan, et al. "Medical document clustering using ontology-based term similarity measures." (2008).

[12] Huang, Yinghao, Xipeng Wang, and Yi Lu Murphey. "Text categorization using topic model and ontology networks." Proceedings of the International Conference on Data Mining (DMIN). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2014.

[13] Tomas Borovicka, Marcel Jirina, Jr., Pavel Kordik "Advances in Data Mining Knowledge Discovery and Applications." (2012).