Analyzing CV/Resume using Natural Language Processing and Machine Learning

Department of Computer Science and Engineering

School of Engineering and Computer Science

BRAC University

Supervisor: Dr. Jia Uddin

Submitted by

Md. Tanzim Reza – 14101061

Md. Sakib Zaman – 14101171

Declaration

We hereby declare that, this thesis is based on results obtained from our own work. Due acknowledgement has been made in the text to all other material used. This thesis, neither in whole nor in part, has been previously submitted to any other University or Institute for the award of any degree or diploma.

Signature of Supervisor **Signature of Authors**

_____ _____

Dr. Jia Uddin Md. Tanzim Reza (14101061)

_____

Md. Sakib Zaman (14101171)

Acknowledgement

First and foremost, we would like to thank Almighty Allah for enabling us to initiate the research, to put our best efforts and successfully conclude it.

Secondly, we submit our heartiest gratitude to our respected Supervisor Dr. Jia Uddin for his contribution, guidance and support in conducting the research and preparation of the report. Every last involvement of his, starting from instilling in us the deadliest of fears to the kindest words of inspiration has permitted us to effectively complete the thesis. We are truly grateful to him.

We revere the patronage and moral support extended with love, by our parents as well as our friends. They helped us with their direct or indirect suggestions which aided in achieving our goal. We would also like to acknowledge the assistance we received from numerous resources over the Internet especially from fellow researchers' work.

Last but not the least, we thank BRAC University for providing us the opportunity of conducting this research and for giving us the chance to complete our Bachelor degree.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

NLP – Natural language Processing

ML – Machine Learning

CV – Curriculum Vitae

HTML – Hyper Text Markup Language

ANN – Artificial Neural Network

ID3 – Iterative Dichotomiser 3

PDF – Portable Document Format

JSON – JavaScript Object Notation

Abstract

This paper proposes a model of extracting important information from the semi-structured text format in a curriculum vitae or resume and ranking it according to the preference of the associated company and requirements. In order to achieve the desired goal, the entire process has been divided into 3 basic segments. The first segment consists of segmenting the entire CV / Resume based on the topic of each part, the second segment consists of extracting data in structured form from the unstructured data and the final segment consists of evaluating the structured data by decision tree algorithm and training the system. The structured data extraction process is done by segmenting the entire CV / Resume by converting it to HTML. After the conversion to structured data, decision tree algorithm techniques are used to classify the input into different categories based on qualifications and then the data with positive weight is used to train the system for future benefit. Finally, classifier algorithm apart from decision tree such as logistic regression is used to compare the classification result.

CHAPTER 01

Introduction

After completing education the next phase that comes in a person's life is job. However, there are lots of people who start working before completing their formal education. While searching for jobs the most important thing to represent an applicant is Curriculum Vitae (CV) or Resume. In this era of technology, job searching has become more smart and easier at the same time. However, there are more than enough applicants for a single job and it is really tough for an employer to select candidates only based on their CV / Resume. To solve this problem, there are companies who provide specific format for their applicants so that they can make this process a little bit easier. Even after doing that the process is still pretty boring and most of the cases full of errors.

1.1Motivation

There have been lots of work done for job searching process. Whereas, the process of selecting a candidate based on their CV / Resume has not been completely automated. To solve this problem, an approach combined with Natural Language Processing (NLP) and Machine Learning (ML) seems like a feasible opportunity. Nowadays, there are lots of research done in both Natural Language Processing and Machine Learning. Most importantly, these two topics are used in day to day life almost every day while using mail, online shopping etc.

1.2 Contribution Summary

Although there were some research to automate the process in some other way and there were some research to make the process less boring and easier at the same time, but there is still some room for improvement. Many of the natural language processing or machine learning techniques came from the analysis on how brain interprets real life data. For example, Artificial Neural Networks (ANN) is a computer program came from the concept of the biological neural

network in animal brain [1]. Therefore, the first objective of this paper is to analyze how human brain works in case of analyzing a piece of CV/ Resume. Research shows that, 90% of all CVs/ Resumes are checked for less than 2 minutes [2] by the employers. This implies that, in most of the cases the employers only look at the bits of important parts or the points of interest in the CV/ Resumes and ignores the rest. The specific segmentation scheme of a general CV/ Resume makes it far easier to analyze and understand the necessary information. Therefore, the first objective was to segment the CV/ Resume into parts and then separate them in order to figure out the topics of each sentence through analyzing the keywords of each segment.

1.3 Thesis Outline

- Chapter 2 focuses on the previous work done in the area and it also describes about the literature review of related works in this particular field
- Chapter 3 discusses about the proposed model along with other necessary details for this system
- Chapter 4 presents the result and experimental analysis on this topic and it also analyzes the system performance
- Chapter 5 concludes the whole paper and also discusses about the future scope of this system

CHAPTER 02

Background Study

2.1 Different Formats of CV / Resume

While in CV / Resume data formats that are used is not completely unstructured, it is still quite challenging to take them into structured format as there is no set in stone rule for writing a CV/ Resume. As a result, many possible way of representing qualifications in a CV/ Resume has been established so far such as chronological CV/ Resume and functional CV/ Resume [2]. Beyond these two, there are many other formats and many people follow their own unique style to make their CV/ Resume stand out from other ones. Additionally, there is a tendency of adding visual elements in a CV/ Resume to make it more interesting to visualize. Opposed to many of the visual elements just being there for aesthetic purpose, there are exceptional cases when someone use visual elements like graphs or charts to represent important information such as their skills because creating and interpreting graphs or charts encourages critical thinking [3] . As in most of the cases these graphs are included in image formats and there is no definitive way to process them without using image processing techniques and these CVs/ Resumes will be kept out of consideration as it is beyond the scope of this paper.

2.2 CV / Resume Analyzing Process

In the past, CVs/Resumes submitted by job seekers used to be manually analyzed and judged by the employers [3]. This method is still followed in the recent times. However, as the big companies often need to deal with hundreds of CVs/Resumes each and every day, it has become very problematic and time consuming to handle such a big number of CVs/Resumes one by one. As a result, many companies started to provide specific formats or forms where the job seekers need to fill up with required information and then the CV/Resume will be analyzed by machine with simple pattern recognition and keywords searching. While this method reduced the workload for the employers, it increased the amount of work for the applicants significantly as they need to

maintain different formats for each job they apply for. Additionally, it also tends to reduce the creativity and the flexibility of writing the skills along with the qualifications in a CV/Resume.

2.3 Natural Language Processing Approach

With all the pros and cons in mind, there has always been an attempt to find an automated method which finds the best of worlds, where the employers can easily select qualified candidates in a short time and where the applicants can also demonstrate their creativity while maintaining just one format to apply in different organizations. The innovation in the field of Natural Language Processing [4] along with Machine Learning [5] has been really helpful in this case. The ability to understand unstructured written language and extract important information from it to teach the machine is exactly what is needed to analyze any written documents such as resume papers just like human being.

2.4 Machine Learning Approach

Along with Natural Language Processing, researchers also used Machine Learning to make their models more accurate and correct. Since, there are various techniques of Machine Learning, therefore, there are various approaches to train a model and solve problems. Logistic regression [6], naive Bayes classifier [7], Decision trees [8] are very commonly used machine learning based techniques that are used to determine whether some is right or wrong, good or bad. They have also been used in the past to determine various diseases, like cancer [9]. As it has been tried to make a decision taking system that determines whether a CV/ Resume is qualified or not, the concept of decision tree will be useful. Moreover, there are different types of decision tree algorithms that exist such as ID3 algorithm [10] and the C4.5 algorithm [11] which is the successor of ID3 algorithm. For this research ID3 algorithm will be used.

2.5 Combined approach with NLP and ML

Other than CVs/Resumes, Natural Language Processing and Machine Learning techniques were also used in various fields such as Essay Grading System [12], Composition Review System[13] etc. These fields used different types of Natural Language Processing and Machine Learning techniques to get better results and to suggest modifications for the system. As a result, these concepts are crucial to give positive weights to CVs/Resumes in our model.

2.6 CV / Resume Processing

There have been different methodologies to process and evaluate CVs/ Resumes in an automated system in the past, but no other work has been done there that does the job while being aware of the layout. So this is an excellent opportunity to reduce the overhead of the system by taking the advantage of layout identification and by doing a lot of work just by the layout aware parsing. The layout aware extraction was done for scientific articles in the past [14], but nothing is done for processing resumes.

In most of the case, CVs/Resumes are usually in two formats: PDF and DOCX. There are libraries in python that can read both the file formats very easily. There is a library in python called PDFminer that can read PDF files and another library called urllib that can convert PDF files to HTML. There is also functionality in a library called beautiful soup that can pull data out of HTML and XML files [15]. It can reverse engineer the HTML file to HTML code [15].

CHAPTER 03

Proposed Model

3.1 System Design

To design this model, various other models [16] and job search theories [17] were analyzed. As a result, the whole system was easily segmented and designed properly to meet the need of both the employers and job seekers. However, the system will be more efficient with the amount of data the system gets.

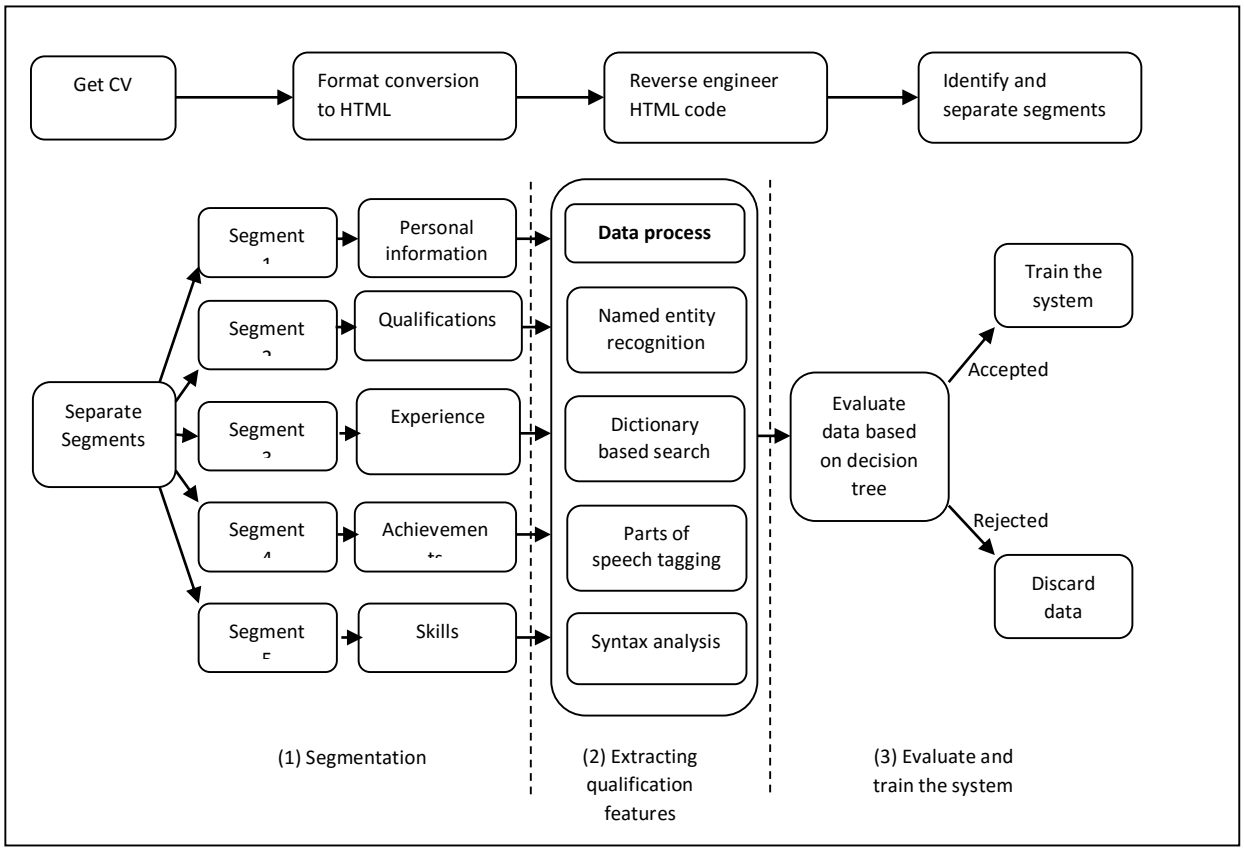

Fig. 3.1.1 Block diagram of the whole system

To build the decision tree model and the parser it is really important to have sufficient amount of data so that the results are as accurate as possible. However, this sufficient amount should always be as much as possible so that the whole system can be trained properly. Fig. 3.1.1 describes the proposed model with a block diagram.

While segmenting a CV/Resume, a part that plays a huge role in segmentations is the font information of the CV/Resume. As mentioned previously, except some specific cases most of the CV/Resume are written in semi-structured form. In a generally structured CV/Resume, the segments usually starts with a heading, a bolded headline, a headline with bigger font than the other texts or something along the line with that. Therefore, it is clear that the font information do carry a lot of weight in segmenting a CV/ Resume.

However, when a text is read by computer, it is read as a plain text and in the process, it misses a huge amount of information. On the other hand segmenting based on line breaks does not work here either because segments with bullet points may carry a lot of extra line breaks in the same segment. Additionally, the segmentation algorithms can be performance hungry and may not achieve the desired accuracy in segmenting a piece of CV/ Resume. As a result, if the font information is preserved, it can be really helpful in segmentation. Fig. 3.1.2 explains the HTML conversion and HTML converted information for the CV/Resume.

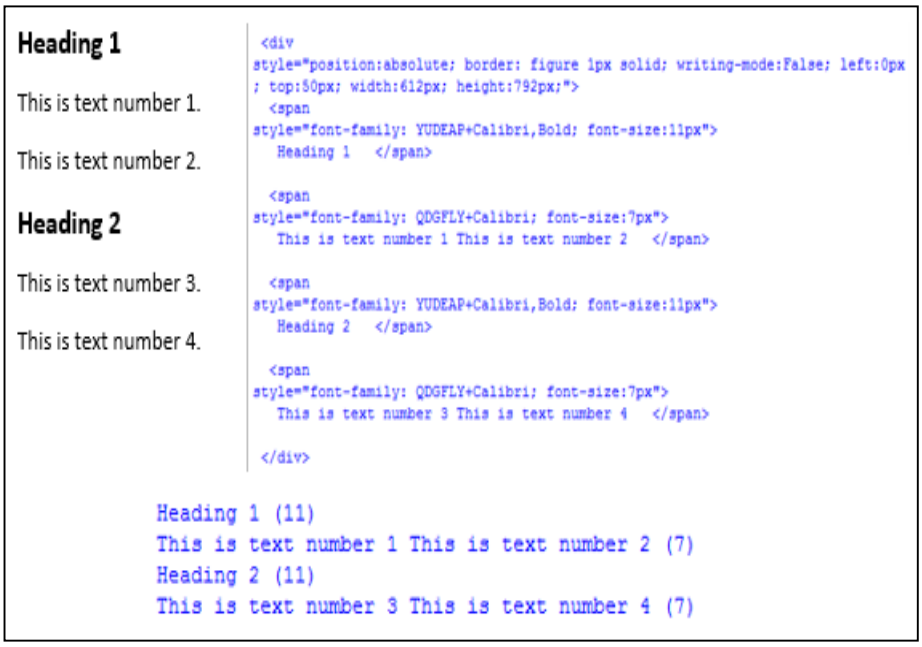

Fig. 3.1.2 Sample data converted into HTML

3.2 Segmentation

In our proposed model, a way to keep the font information can be converting the CV / Resume format to HTML and the reverse engineer the HTML files to HTML code. An HTML code carries information like if a text is bold or not, or the font is bigger or not etc. From this information it will be tried to detect when a segment is starting. Moreover, a way to detect headline through analyzing from a parse tree will be there, the font analyzing technique will work as a layer on top of that.

3.2.1 Finalizing Segment Detection

While processing a CV/Resume, detecting the font only may not be enough as this brings lots of ambiguity. First of all, there are specific CVs / Resumes that may not start each segment with bold alphabets at all. Additionally, some bold worded line with big font may get detected as a heading or start of a segment where the subjected line resembles something entirely different. Therefore, to make the segmentation method more concrete, headings of some already existing CVs / Resumes will be taken which will work as the sample data of our work and will be tried to build a parse tree from it. Then it will be tried to extract certain information from it to indicate the possible structural information of a headline. Luckily, most of the headings in CV / Resume are pretty short for creating and extracting from a parse tree is a pretty simple task.

3.3 Extracting Qualification Features

Once a CV / Resume is segmented, extracting the relevant information becomes easier since it can be worked based on the specific topic that each of the segments has. Likewise, one segment may contain personal information, another segment may carry information related to career and some other segment may describe experience and so on. Now, identifying information with tangible value such as CGPA, grade or years of experience is relatively easy and can mostly be done with simple pattern recognition in python or any other language. However, extraction of more complex data will require multinomial logistic regression model. Additionally, extracting and evaluating information with intangible values such as institution name (different institute have different

values), degree can be tricky. Furthermore, tangible values may stay associated with intangible values where the actual value may be judged only in pair. For example, checking CGPA or grade as an atomic value may not be enough, depending on the institution from which the CGPA or the grade came, the value may vary a lot. In this case, mapping the intangible values into tangible values helps a lot in terms of weighting the values of these qualifications. Therefore, a large table with intangible values mapped with tangible weight comes into use here. Based on the scope, the table can become huge so instead of making the table manually, giving the machine sample data in a small table and let the system expand the table by itself through learning more is useful.

3.4 Training the System

Once a CV / Resume is analyzed and weighted into values, there will a total marks given to a CV / Resume and it will be graded based on how much qualification the candidate has. Just like any other machine learning based model, the success rate of the system will depend on the amount of pre-existing data the machine has which can be used to compare and evaluate new information. As a result, when a CV / Resume is weighted positively, if the CV / Resume carries any new information or something that can enrich or solidify the existing information even more, the system will learn it and expand the scope of the training data so that it can be used for future purpose.

CHAPTER 04

Experimental Result and Analysis

4.1 Algorithm

```
Segmentation(File pdf, List heading_tree)
  1. Read pdf file
  2. Convert file to HTML
  3. If fail, return error
  4. Else, turn the file into HTML code
  5. Remove fillers, e.g. angle braces, irrelevant HTML tags
  6. Read all types of font sizes from code
  7. Average = summation of font sizes / number of different font size
  8. Initialize heading candidate queue
  9. Load first line of text
  10. Read line
  11. If font size of line is greater than average
  12. Push into heading candidate queue
  13. Go to next line
  14. If next line! = null, go back to 10, else go to 15
  15. If candidate queue is empty, go to line 22
  16. Candidate ← candidate queue.pop()
  17. Parse the sentence through heading parser
  18. If parsing successful, make a segment of text till the next
  heading
  19. Determine segment topic
  20. Segment queue ← push segment
  21. Go back to 15
  22. return segment queue
```

Based on the algorithm the whole system will rank the candidates according to the requirement given by the employer. Then based on the limitations and constraints the system will provide top candidates for the next phase of the recruitment process.

4.2 Application of Algorithm

In this section, it has been shown that the segmentation procedure for one CV. After converting the PDF file to HTML, the data that has been received is represented in the following way,

---

Page 1

Md. Sakib Zaman
Flat-A3, 127 West Kafrul, Agargaon,
Taltola, Dhaka – 1207
Mobile: +8801912397694
E-mail: sakib2033@gmail.com

Employment Status

- Currently working as a Student Tutor/Teaching Assistant at Department of Computer Science & Engineering, BRAC University from January 2017
- Currently working as a Student Trainer at Competitive Programming Training Session organized by Department of Computer Science & Engineering, BRAC University and BRAC University ACM Students Chapter from August 2016
- Currently working as a Student Mentor at First Year Advising Team, BRAC University
- Former Intern Software Engineer at Projukti Next from 2nd May 2017 to 31st May 2017

Educational Qualification

- Final year student of Computer Science and Engineering, BRAC University, Dhaka
CGPA: 3.74 in scale of 4.0 (till April, 2017)
- H.S.C. (2013) from Notre Dame College, Dhaka
GPA: 5.0 in scale of 5.0
- S.S.C (2011) from Sher-E-Bangla Nagar Govt. Boys' High School, Dhaka
GPA: 5.0 in scale of 5.0

Technical Skills

- Programming Languages: Java, C, C++, C#
- Operating Systems: Windows, Linux
- Database System: MySQL
- Web: HTML5, CSS3

Achievements in Competitions and Programming

- 1st Runner-up in BRAC University Intra University Programming Contest, Spring 2016
- Participated in DIU ACM ICPC World Finals Warm-Up Contest 2016 and ranked 19th
- Participated in ACM ICPC Dhaka Regional On Site 2015 and ranked 100th
- Participated in EATL - Prothom Alo Apps Contest 2016 and was in top 30
- Participated in BRACATHON 2015 & BRACATHON II 2017
- More than 200 solved problems in UVA Online Judge, UVA user name: Sakib2033
- Participated in various other Online and Onsite Programming Contests

Page 2

Projects

- Currently working on an Online File Server System for Educational Institutions
- Library Management System for Data Structure course
- Hospital Management System for Data Structure course
- Cineplex Management System for Database course
- Dhaka City Management System for Software Engineering course

Academic Awards

- Awarded Performance Based Scholarship to study at BRAC University
- Awarded Lifetime Membership Award and Youth Leadership Award from Notre Dame English Club, Notre Dame College, Dhaka in 2014
- Awarded 100% Attendance Certificate from Notre Dame College, Dhaka in 2013

Synergic Activities

- Member of BRAC University Computer Club and BRAC University ACM Students Chapter since 2014
- Worked as a Judge in National English Carnival 2016 & National English Carnival 2017 organized by Notre Dame English Club, Notre Dame College, Dhaka
- Worked as a Judge in Intra College English Fiesta 2015 organized by Notre Dame English Club, Notre Dame College, Dhaka
- Worked as Assistant General Secretary of Contest and Organization (2012-2013) in Notre Dame English Club, Notre Dame College, Dhaka

(Last updated on 19th May, 2017)

While reverse engineering the HTML code, a very rough conversion into HTML code has been found. After some cleaning up and processing, the following HTML code is received.

[' Md. Sakib Zaman ', ' Flat-A3, 127 West Kafrul, Agargaon,
 ', ' Taltola, Dhaka 1207
 ', ' Mobile: +8801912397694
 ', ' E-mail: sakib2033@gmail.com
 ', ' Employment Status
 ', ' Currently working as a ', ' Student Tutor/Teaching Assistant ', ' at Department of Computer Science
 ', '& Engineering, BRAC University from January 2017
 ', ' Currently working as a ', ' Student ', ' Trainer ', ' at Competitive Programming Training Session
 organized by Department of Computer Science & Engineering, BRAC University and BRAC
 University ACM Students Chapter from August 2016
 ', ' Currently working as a ', ' Student Mentor ', ' at First Year Advising Team, BRAC University
 ', ' Former ', ' Intern Software Engineer ', ' at Projukti Next from 2 ', ' nd ', ' May 2017 to 31 ', 'st ', ' May 2017
 ', ' Educational Qualification
 ', ' Final year student of Computer Science and Engineering, BRAC

University, Dhaka
 ', ' CGPA: ', ' 3.74 ', ' in scale of 4.0 (till April, 2017)
 ', ' H.S.C. (2013) from Notre Dame College, Dhaka
 ', ' GPA: 5.0 in scale of 5.0
 ', ' S.S.C (2011) from Sher-E-Bangla Nagar Govt. Boys High School, Dhaka
 ', ' GPA: 5.0 in scale of 5.0
 ', ' Technical Skills
 ', ' Programming Languages: ', ' Java, C, C++, C#
 ', ' Operating Systems: ', ' Windows, Linux
 ', ' Database System: ', ' MySQL
 ', ' Web: ', ' HTML5, CSS3 ', ' Achievements in Competitions and Programming
 ', ' 1st Runner-up in BRAC University Intra University Programming Contest, Spring 2016
 ', ' Participated in DIU ACM ICPC World Finals Warm-Up Contest 2016 and ranked 19th
 ', ' Participated in ACM ICPC Dhaka Regional On Site 2015 and ranked 100th
 ', ' Participated in EATL - Prothom Alo Apps Contest 2016 and was in top 30
 ', ' Participated in BRACATHON 2015 & BRACATHON II 2017
 ', ' More than 200 solved problems in UVA Online Judge, UVA user name: Sakib2033
 ', ' Participated in various other Online and Onsite Programming Contests
 ', ' Projects
 ', ' Currently working on an Online File Server System for Educational Institutions
 ', ' Library Management System for Data Structure course
 ', ' Hospital Management System for Data Structure course
 ', ' Cineplex Management System for Database course
 ', ' Dhaka City Management System for Software Engineering course
 ', ' Academic Awards
 ', ' Awarded ', ' Performance Based Scholarship ', ' to study at BRAC University
 ', ' Awarded ', ' Lifetime Membership Award ', ' and ', ' Youth Leadership Award ', ' from Notre Dame
 ', '

English Club, Notre Dame College, Dhaka in 2014
 ', ' Awarded ', ' 100% Attendance Certificate ', ' from Notre Dame College, Dhaka in 2013
 ', ' Synergic Activities
 ', ' Member of BRAC University Computer Club and BRAC University ACM Students Chapter since
 ', ' 2014
 ', ' Worked as a Judge in National English Carnival 2016 & National English Carnival 2017
 ', ' organized by Notre Dame English Club, Notre Dame College, Dhaka
 ', ' Worked as a Judge in Intra College English Fiesta 2015 organized by Notre Dame English Club,
 ', ' Notre Dame College, Dhaka
 ', ' Worked as Assistant General Secretary of Contest and Organization (2012-2013) in Notre
 ', ' Dame English Club, Notre Dame College, Dhaka
 ', ' (Last updated on 19 ', ' th ', ' May, 2017)
 ']

The important factor to notice here is that, now all the font information such as font size, font style, line breaks etc has been received that is very crucial for the system. In order to separate the segments, the most important information that is needed is the font size.

Now, some necessary information has been gathered to see that, which font appeared how many times. From the following figures it can be seen that,

---

font 7 has appeared 2 times.

font 8 has appeared 1 times.

font 11 has appeared 64 times.

font 12 has appeared 6 times.

font 14 has appeared 7 times.

font 15 has appeared 1 times.

If the information is plotted in a histogram, it can be seen that,

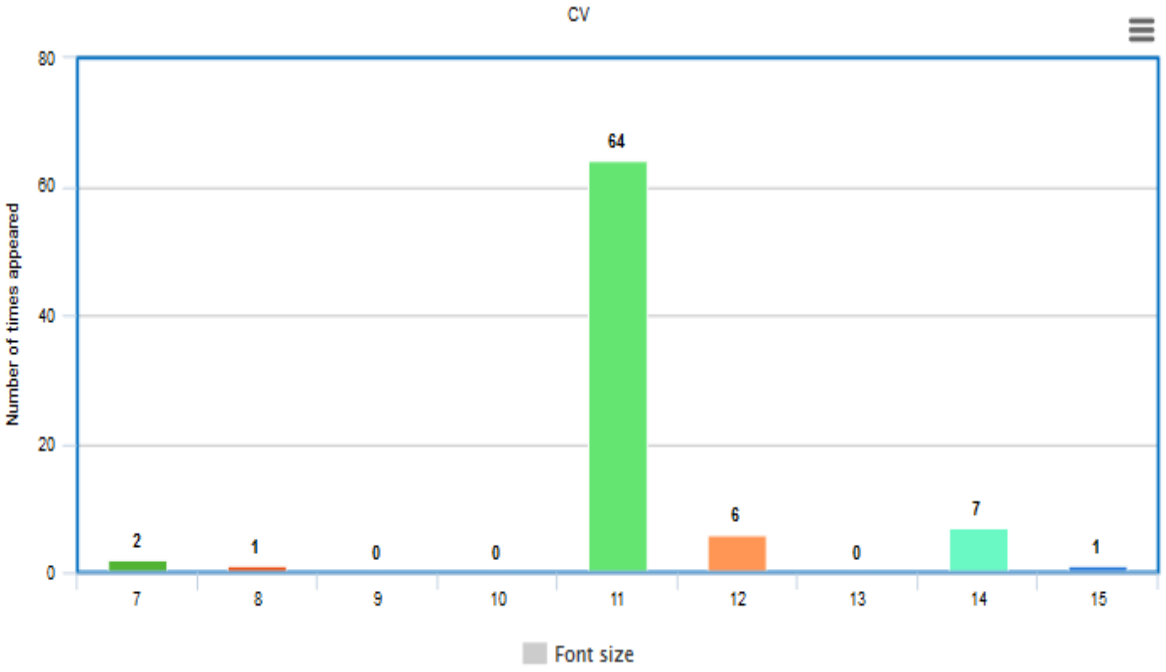

Fig. 4.2.1 Histogram from CV of Sakib

Moreover, here are some more histograms from some other CVs/ Resumes where CV/ Resume information are converted into HTML code and then font information are collected and represented in the following figures.

CV link: (CV of Sample Candidate 1)

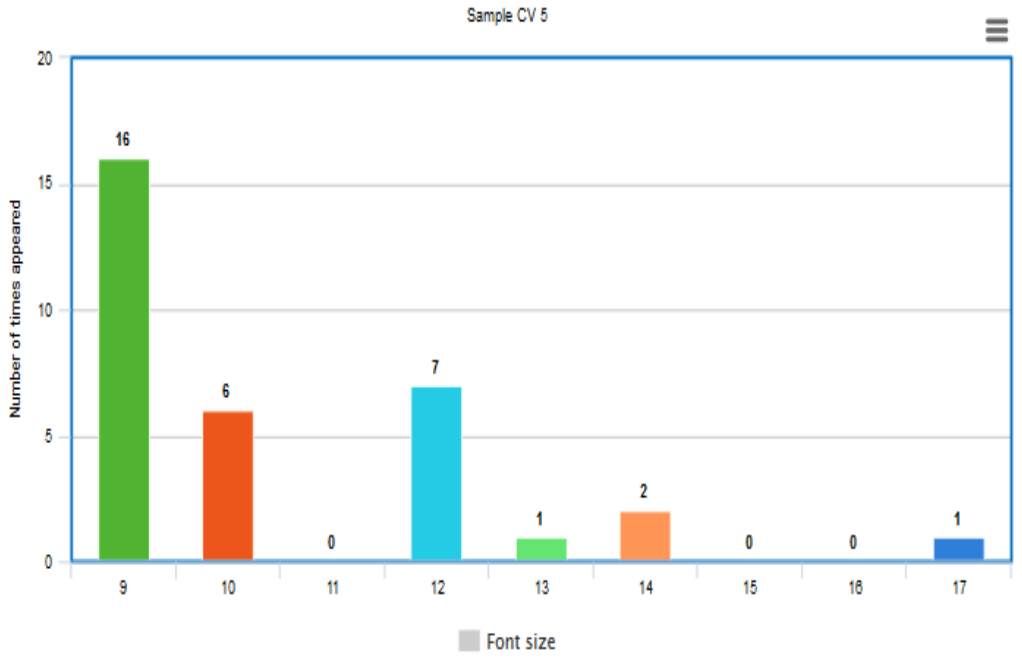

Fig. 4.2.2 Histogram from CV of Sample Candidate 1

CV link: (CV of Sample Candidate 2)

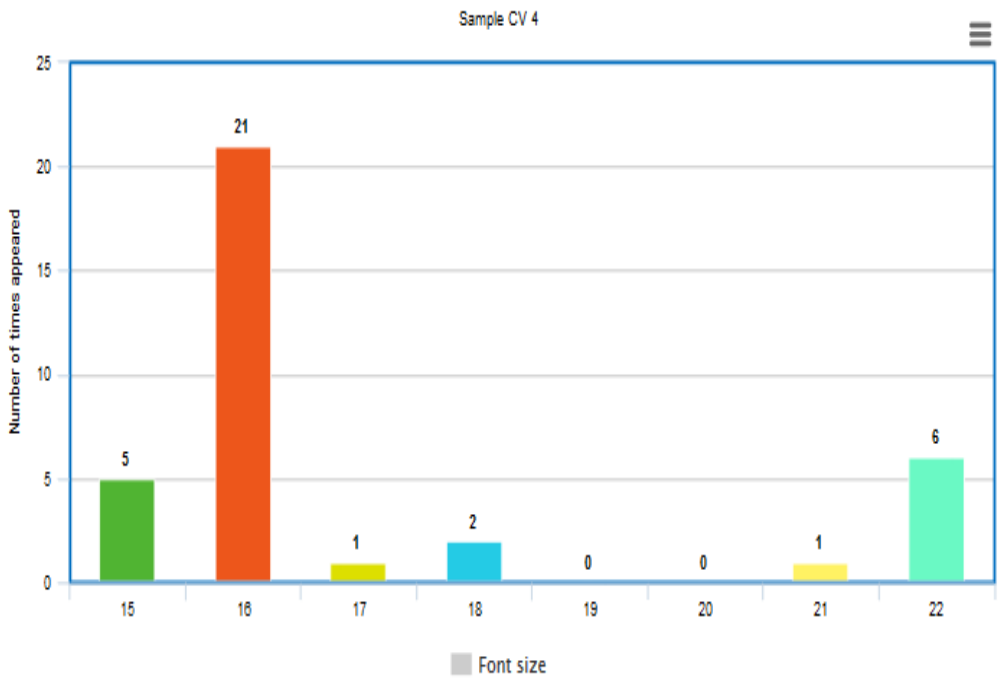

Fig. 4.2.3 Histogram from CV of Sample Candidate 2

CV link: (CV of Sample Candidate 3)

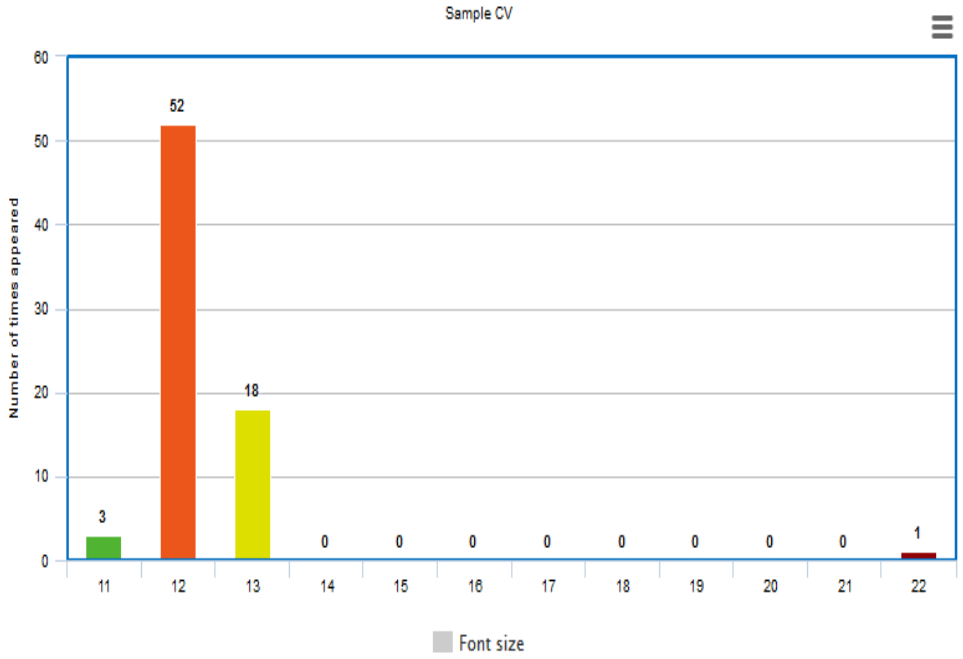

Fig. 4.2.4 Histogram from CV of Sample Candidate 3

CV link: (CV of Sample Candidate 4)

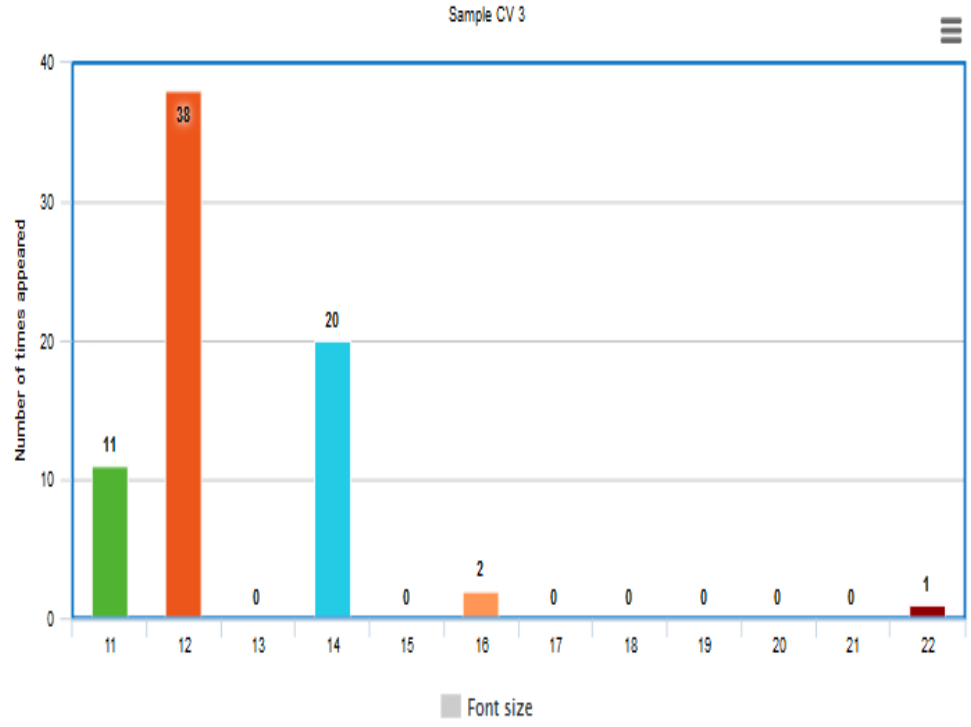

Fig. 4.2.5 Histogram from CV of Sample Candidate 4

CV link: (CV of Sample Candidate 5)

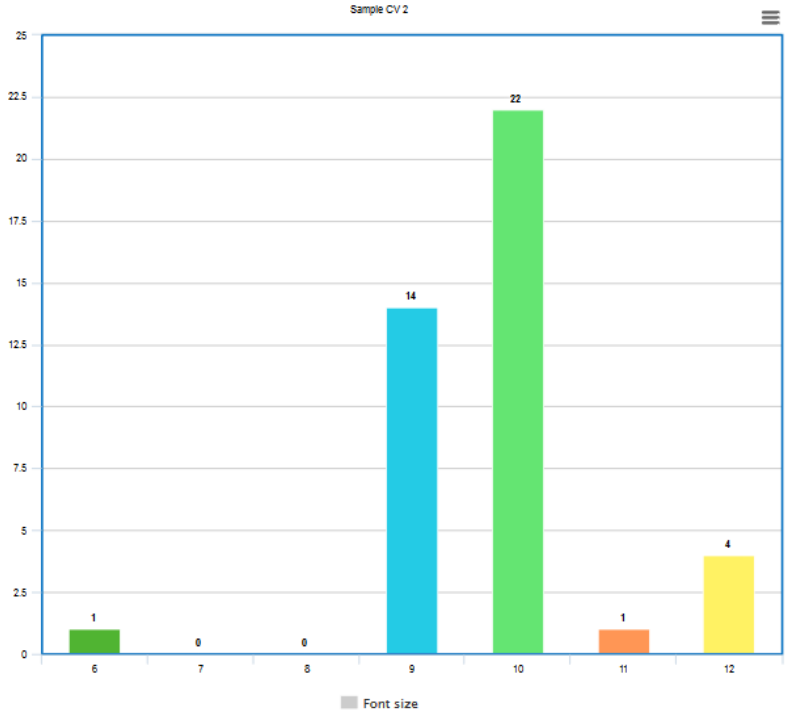

Fig. 4.2.6 Histogram from CV of Sample Candidate 5

While looking at each of the bar graphs, it can be easily noticed that, there is usually one or two specific fonts that appears a lot more that the other font sizes. This is rightfully so, because in general cases only the headings or the start of a segment in a piece of CV/Resume contains bigger fonts. Therefore, those fonts appears less amount of times. Other than that, all the other writings are written in some other fonts and those fonts appears much often.

One important factor to notice here that, the attempt of segmenting purely based on font size information is not completely reliable. This is because there are CVs/ Resumes where the font size of the headings are the same as the general text. In this case, the abovementioned procedure will absolutely fail. In addition to that, in some CVs/ Resumes there are semi-headings under the main heading where the font size of the semi-headings are almost similar to the main heading and this creates confusing results. As a result, to distinguish the headings and to get a more precise result, syntax analysis is performed on the text and this analysis is performed in an order from big font size to small font size.

4.3 Syntax Analysis

Before performing syntax analysis on the text, it is needed to do some pre-analysis such as how many segments will be looking for in general cases and what information are entailed in those segments. In general case, following segments in a CV/ Resume will be looked for,

1. Personal information
2. Academic qualifications
3. Working experience
4. Skills
5. Extra-curricular activities
6. Awards & achievements
7. Field dependent achievements (e.g. projects for engineering students, voluntary or event participations for business students, exclusive degrees for doctors)

As the field dependent achievements are different for different fields, this greatly increases the complexity and therefore, the scope of this paper will be kept among the engineering side CVs/ Resume only, where mainly things like projects, competition results are showcased.

In order to perform syntax analysis [18], it is needed to collect some sample headings from which an initial syntax tree is created and the syntax analysis will be performed based on that syntax tree.

Here are some examples of Syntax Trees of all the potential headings of a CV / Resume,

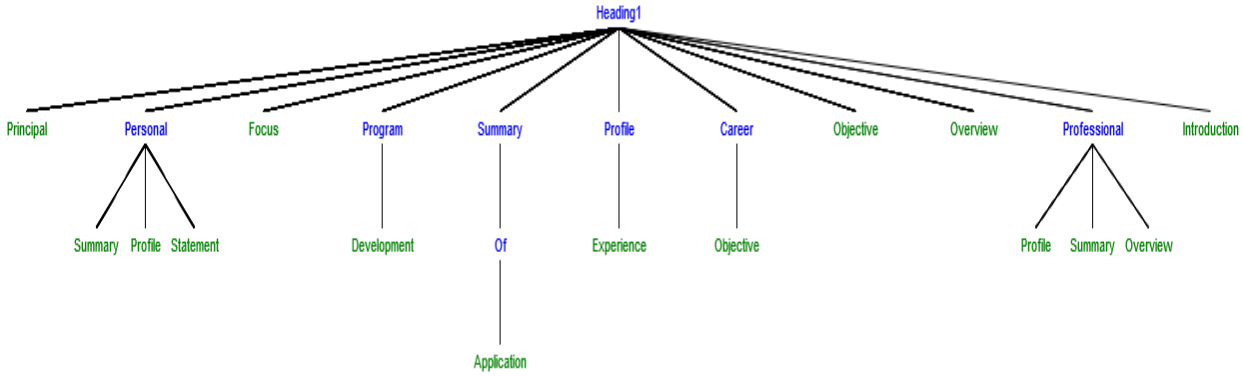

Fig. 4.3.1 Syntax Tree of Personal Information Heading

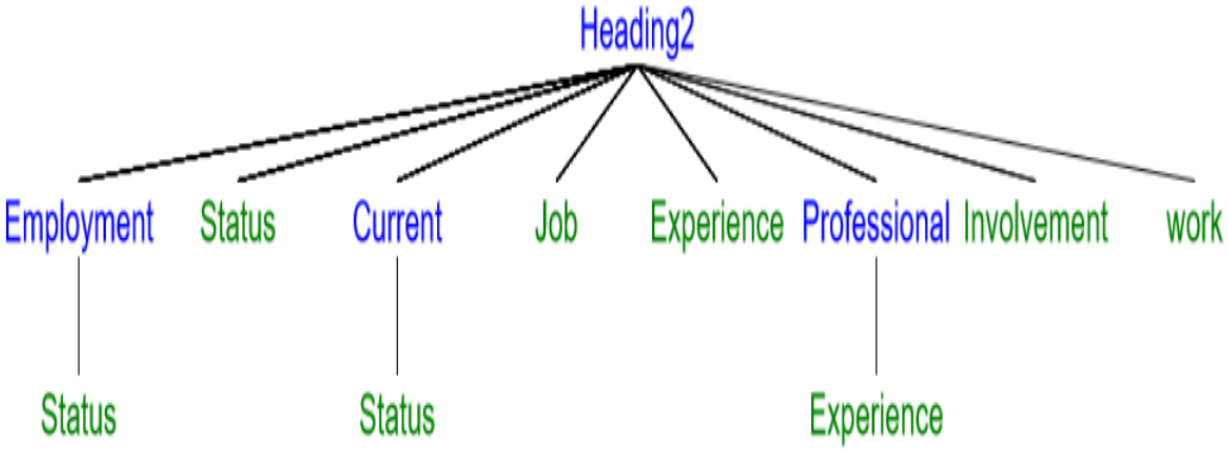

Fig. 4.3.2 Syntax Tree of Working Experience Heading

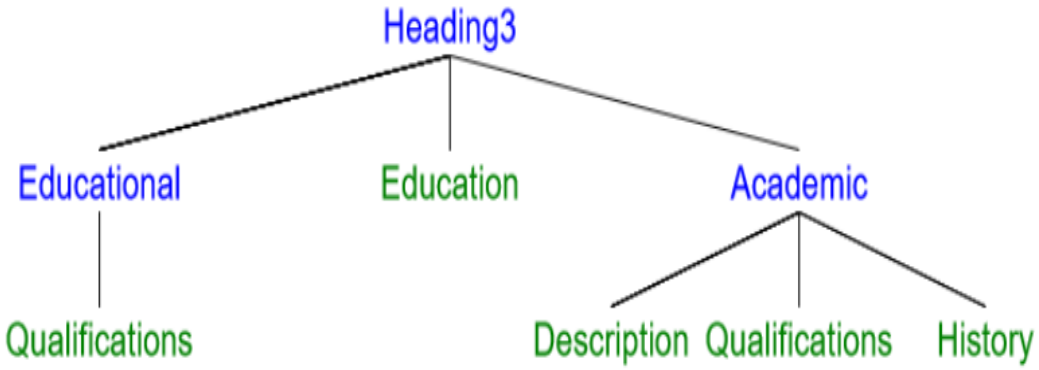

Fig. 4.3.3 Syntax Tree of Educational Qualification Heading

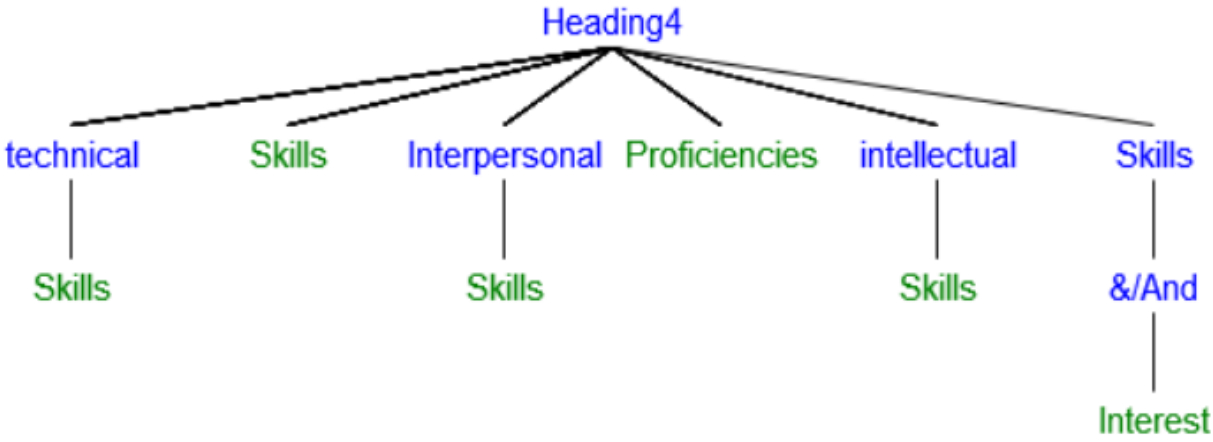

Fig. 4.3.4 Syntax Tree of Skills Heading

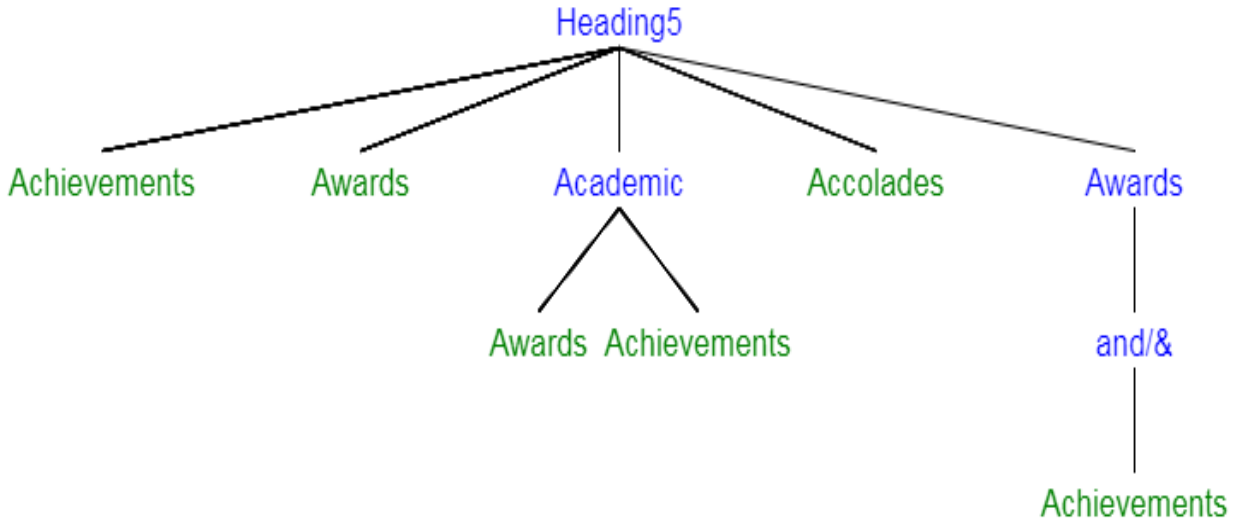

Fig. 4.3.5 Syntax Tree of Achievements Heading

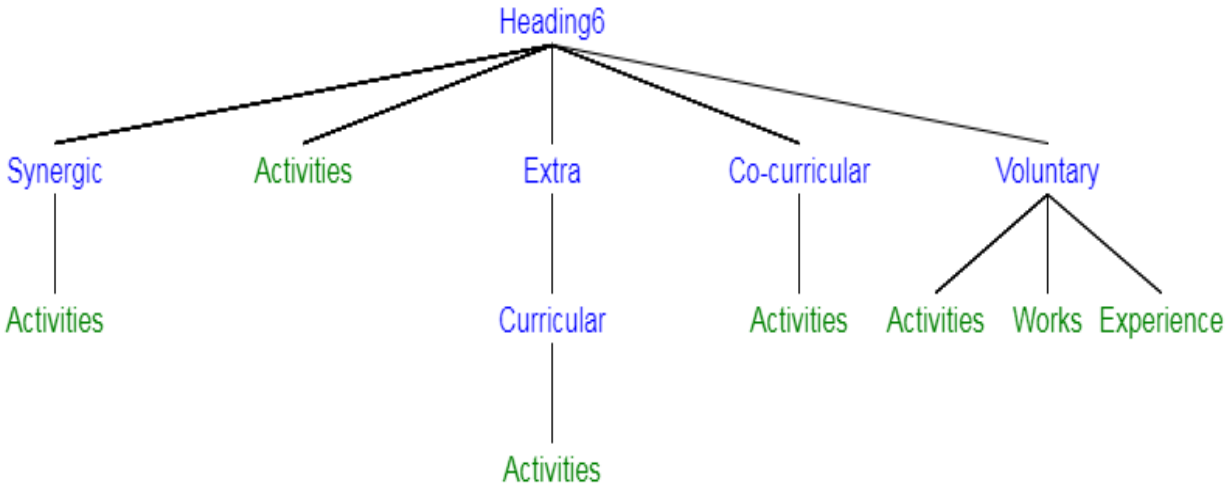

Fig. 4.3.6 Syntax Tree of Extra-curricular Activities Heading

The syntax trees given above are sample trees and a representative data sample to identify section headings in the CVs/ Resumes. In order to identify headings more accurately, there should be more data and the syntax trees will be much bigger. The more data there is, the better.

Now that syntax trees are done with each and every segment, syntax analysis on each of the lines will be performed. The analysis is done on the lines in an order from higher font to lower font, with a hope that most of the headings exist in the lines with bigger font so it's possible to get all the headings by performing syntax analysis on only a little portion of all the lines.

After performing analysis on around 50 CVs / Resumes, a common pattern was discovered. In a large number of CVs/ Resumes, the line with the largest font represents the name of the CV/ Resume owner. There is a font smaller than the largest one that usually represents the font of the headings. However, this is not entirely consistent as there are also the font size of the sub headings that gets included here if exists. The font size below it represents the general text of CVs/ Resumes and this font size is presented in the largest number. Performing syntax analysis on these CVs/ Resumes are relatively simple and usually all the desired headings are found after performing syntax analysis under only 10 times. In exceptional cases where no clear font pattern is found, the syntax analysis has to be performed more intensively and more number of times.

In the CV/ Resume converted into HTML, all the desired headings were found after performing syntax analysis for only 8 times. When all the headings are found, segmenting is a very easy job in a general CV/ Resume. All it is needed is to gather texts from one heading to the next heading and a segment is found.

After the segmentation part is done, the data now looks like the following,

---

Segment 1 (Name and details)

Md. Sakib Zaman Flat-A3, 127 West Kafrul, Agargaon, Taltola, Dhaka 1207 Mobile: +8801912397694 E-mail: sakib2033@gmail.com

Segment 2 (Working experience)

Employment Status Currently working as a Student Tutor/Teaching Assistant at Department of Computer Science & Engineering, BRAC University from January 2017 Currently working as a Student Trainer at Competitive Programming Training Session organized by Department of Computer Science & Engineering, BRAC University and BRAC University ACM Students Chapter from August 2016 Currently working as a Student Mentor at First Year Advising Team,

BRAC University Former Intern Software Engineer at Projukti Next from 2 nd May 2017 to 31st May 2017

Segment 3 (Educational qualifications)

Educational Qualification Final year student of Computer Science and Engineering, BRAC University, Dhaka CGPA: 3.74 in scale of 4.0 (till April, 2017) H.S.C. (2013) from Notre Dame College, Dhaka GPA: 5.0 in scale of 5.0 S.S.C (2011) from Sher-E-Bangla Nagar Govt. Boys High School, Dhaka GPA: 5.0 in scale of 5.0

Segment 4 (Technical skills) – Field dependent

Technical Skills Programming Languages: Java, C, C++, C# Operating Systems: Windows, Linux Database System: MySQL Web: HTML5, CSS3

Segment 5 (Awards & achievements)

Achievements in Competitions and Programming 1st Runner-up in BRAC University Intra University Programming Contest, Spring 2016 Participated in DIU ACM ICPC World Finals Warm-Up Contest 2016 and ranked 19th Participated in ACM ICPC Dhaka Regional On Site 2015 and ranked 100th Participated in EATL - Prothom Alo Apps Contest 2016 and was in top 30 Participated in BRACATHON 2015 & BRACATHON II 2017 More than 200 solved problems in UVA Online Judge, UVA user name: Sakib2033 Participated in various other Online and Onsite Programming Contests

Segment 6 (Projects)-Field dependent:

Projects Currently working on an Online File Server System for Educational Institutions Library Management System for Data Structure course Hospital Management System for Data Structure course Cineplex Management System for Database course Dhaka City Management System for Software Engineering course

Segment 5 (Awards & achievements continued)

Academic Awards Awarded Performance Based Scholarship to study at BRAC University Awarded Lifetime Membership Award and Youth Leadership Award from Notre Dame English Club, Notre Dame College, Dhaka in 2014 Awarded 100% Attendance Certificate from Notre Dame College, Dhaka in 2013

4.4 Extracting information

After the part of proper segmentation, extracting information from the segments becomes much easier. If it is already known that, what segments are actually searching in, an idea can be developed about what should be searched for in each segments. For example, it will be looked for contact number, emails in the personal profile section, CGPA and degrees along with the institution name etc. in the academic career section and so on.

Extracting some of the information is relatively easy and can be done by simple linear pattern matching. To extract the name of the CV / Resume owner the named entity recognition of the NLTK library of python can be used. In this case, it will be started using it from the bigger font sizes to the smaller font sizes because as said earlier, most of the time the biggest font in the CV consists of the name of the CV owner. In addition to that, if there are any contact number or email address in the personal profile section, it can be easily extracted from these by performing simple pattern matching techniques.

Email matcher pattern:

```
[A-Za-z0-9]@[gmail | yahoo | hotmail].com
```

Mobile number matcher pattern:

```
01[5|6|7|8][0-9]{8}
```

Additionally, extracting the age, marital status, physical properties (if needed) of the CV owner is also a simplistic job. Most of the time these values will be around some specific words. It can be expected that, the word ‘age’ or the words ‘years old’ around the numerical value of the attribute.

The more difficult thing is however, extracting the information that is not very clear to extract or may take some ambiguous form. For example, extracting the name of the institutions and for what purpose the candidate got admitted in the institution is a difficult task. In addition to that, extracting the information related to the project works and other accomplishment can also prove to be a very difficult task.

In order to extract these information, syntax analysis was performed and it was looked for some specific sentence structure. For example, if it is assumed that the institution name will be in some sentence structured as ‘studied at <university name>’ or ‘completed <program-name> from <university name>’. From this structure, it can be inferred that:

<Degree name> in <program name> from <Institution name>

This is how data can be taken in some structured form, after taking into structured form, the data can be evaluated more easily. After looking for sentence patterns and searching for keywords, some useful information will be received and that information is taken into JSON format [19]. As there are existing libraries to parse data from JSON format easily, this makes the task faster.

The result in JSON format looks like the following,

{'personal': {'first_name': u'Sakib',
'last_name': u'Zaman',
'current_designation': u‘127 West Kafrul, Agargaon, Taltola, Dhaka’,
'email': u’sakib2033@gmail.com’
‘Mobile’: u‘ +8801912397694’},
'education': [{'School': u'Sher-E-Bangla Nagar Govt. Boys High School’,
'Degree': u'SSC, 2011,
'Duration': None,
'Grade': 5.00},
{'College': u'Notre Dame College’,
'Degree': u'HSC, 2013,
'Duration': None,
'Grade': 5.00},
{'College': u'BRAC University’,
'Degree': u"Bachelor's degree, Computer Science & Engineering, 2017",

'Duration': None,

'Grade': u'3.74cgpa'}],

'experience': [{'Company': u'BRAC University’,

'Duration': None,

'Role': u'Student Tutor'}

{'Company': u'BRAC University’,

'Duration': None,

'Role': u'Student Trainer'},

{'Company': u'BRAC University’,

'Duration': None,

'Role': u'Student Mentor'},

{'Company': u'Projukti Next,

'Duration': u’1 month’,

'Role': u'Intern software engineer}],

'project': [{'Name': u’Online File Server System for Educational Institutions’},

{'Name': u’Library management system’}

{'Name': u’Dhaka city management system’}],

'skills': {

u'Java',

u'C',

u'C++'

u'Windows'

u’Linux'

u'Database'

u'Web'

u'HTML5'

u'CSS3},

'achievements': [{'Event': u'BRAC University Intra University Programming Contest’,

‘Achievement’:u’1’},

{'Event': u'DIU ACM ICPC World Finals Warm-Up Contest’,

'‘Achievement’:u’19’},

{'Event': u'ACM ICPC Dhaka Regional On Site’,
'‘Achievement’:u’100’},
{'Event': u'EATL - Prothom Alo Apps Contest 2016’,
'‘Achievement’:u’top 30’},
{'Event': u'BRACATHON 2015 &amp’,
'‘Achievement’:u’top 30’},
'Event': u'UVA Online Judge’,
'‘Achievement’:u’200’}],
'project': [{'Name': u’Online File Server System for Educational Institutions’},
{'Name': u’Library management system’}
{'Name': u’Dhaka city management system’}],
'summary': None}

4.5 Evaluating Data

When the data extraction in structured format is done, now it is time to justify the data to actually evaluate the CV/ Resume. Decision tree learning algorithm ID3 will be used to justify a CV/ Resume. The ID3 algorithm [10] looks like the following,

```
ID3 (Examples, Target_Attribute, Attributes)
    Create a root node for the tree
    If all examples are positive, Return the single-node tree Root, with label
= +.
    If all examples are negative, Return the single-node tree Root, with label
= -.
    If number of predicting attributes is empty, then Return the single node
tree Root,
    with label = most common value of the target attribute in the examples.
    Otherwise Begin
        A ← The Attribute that best classifies examples.
        Decision Tree attribute for Root = A.
        For each possible value, v_i, of A,
            Add a new tree branch below Root, corresponding to the test A = v_i.
            Let Examples(v_i) be the subset of examples that have the value v_i
for A
            If Examples(v_i) is empty
                Then below this new branch add a leaf node with label = most
common target value in the examples
            Else below this new branch add the subtree ID3 (Examples(v_i),
Target_Attribute, Attributes – {A})
    End
    Return Root
```

Before using ID3 algorithm, the algorithm is trained with dataset from multiple CVs/ Resumes. From the dataset a sample decision table was made. Here's what a sample CV/ Resume evaluation table looks like the following,

TABLE I: Decision Tree Table

CGPA*Varsity Weight	Project weight	Achievement weight	Skills weight	Acceptance
>3.7	Medium	High	Medium	Yes
>3.3 &<3.7	Low	Low	Low	No
<3.3	High	High	High	Yes
>3.7	Low	Low	Low	Yes
<3.3	High	Medium	Medium	No
>3.3 &<3.7	High	High	High	Yes
>3.7	Low	High	Medium	Yes
<3.3	Low	Low	Medium	No
<3.3	Low	Medium	Low	No
>3.3 &<3.7	Medium	Medium	High	No
>3.3 &<3.7	Low	Medium	Medium	No
<3.3	Low	Low	Low	No
>3.7	High	High	High	Yes
>3.3 &<3.7	High	Medium	Low	Yes
>3.3 &<3.7	High	High	Low	Yes

One problem with the ID3 algorithm is, it doesn't work very well with continuous values. Therefore, if there are any continues values, some ranges has to be fixed and the each ranges will become discrete values. The range may vary from company to company.

After table construction, the entropy is calculated and the maximum information gain is also calculated from it. If there are n number of class values, the formula for calculating the entropy [20, 21] is:

$$I(P) = -(p_1 * \log_2(p_1) + p_2 * \log_2(p_2) + .. + p_n * \log_2(p_n))$$

After calculating entropy, the value of information gain is calculated, the formula for calculating information gain is,

```
IG = 1 - I(P)
```

For example, if CGPA vs Skills is taken and I(P) of it is calculated for the class values of yes and no, the following equation is formed,

$I(P_{CGPA}) = (3/9)(0)+(3/9)(-(1/3)\log_2(1/3)-(2/3)\log_2(2/3))+(3/9)(-(1/3)\log_2(1/3)-(2/3)\log_2(2/3))$

$= .605$

So, information gain is

$IG_{CGPA} = 1 - .605 = .395$

$I(P_{Skills}) = (4/9)(-(1/2)\log_2(1/2)-(1/2)\log_2(1/2))+(3/9)(-(1/3)\log_2(1/3)-(2/3)\log_2(2/3))+(2/9)(-(1/2)\log_2(1/2)-(1/2)\log_2(1/2))$

$= .962$

The information gain,

$IG_{Skills} = 1 - .918 = .038$

As it can be evaluated that, the information gain from the CGPA attribute is .605 which is much higher than the information gain from skills attribute which is .038. From this result, it can be inferred that the CGPA attribute should get a higher priority in our tree than the skills attribute.

Now, after running this on all the attributes that has been used to evaluate a CV and then the following tree can be represented as a result,

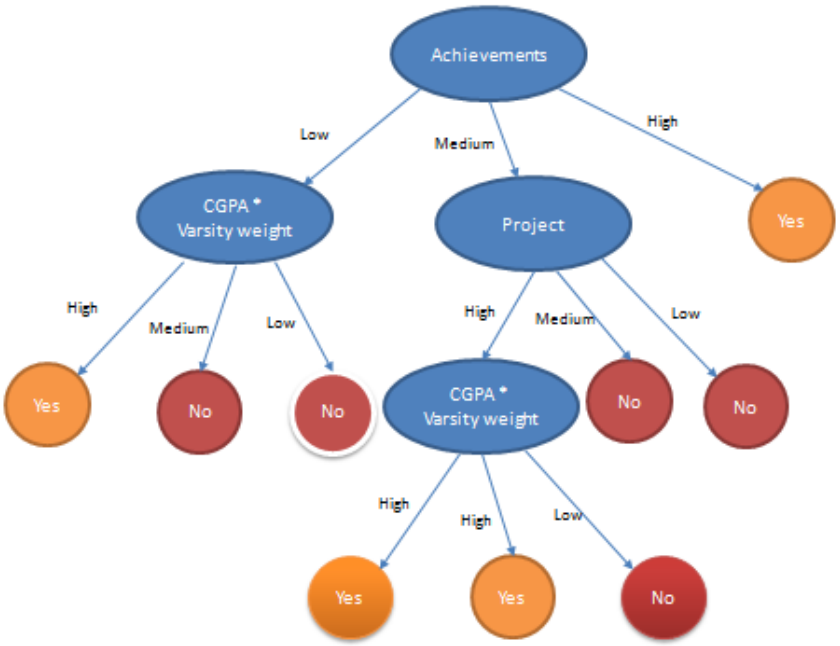

Fig. 4.5.1 ID3 Induced Decision Tree

As it can be seen that, the tree is much smaller than it would be in the original case. From the tree it can be inferred that the skill attribute is not even needed in order to evaluate according to the given criteria. The smaller decision tree helps the ID3 algorithm as the algorithm is good for smaller sized trees. The NumPy[22] and the SciPy[23] open source libraries were used to implement the decision tree algorithm

From the leaf nodes of the tree, evaluation values are received. Now, if a close look is given at sample CV/ Resume, the result that can be seen is,

CGPA > 3.7

Achievement weight = High

Skill weight = High

Project weight = Medium

Finally, a positive result is calculated as final evaluation. This result proves that, this candidate can attend for an interview or can perform the next step such as written examination.

4.6 Final Evaluation

After deciding about the algorithm and after completing the Natural Language Processing step, Machine Learning techniques were used to train our system so that it gives better performance. To train and test our system around 50 CVs / Resumes have been used. After using them and training our system an accuracy of 80% - 85% was received. However, if the system could have used more CVs / Resumes then the accuracy would have been much better. To measure the accuracy and performance the used equations are explained in the following lines,

Accuracy [24] is used to measure the number of CVs / Resumes that was accepted against the number of CVs / Resumes that should have been accepted in manual checking.

$$\text{Accuracy} = \frac{Total\ number\ of\ CVs\ /\ Resumes\ that\ was\ accepted}{Total\ number\ of\ CVs\ /\ Resumes\ that\ should\ have\ been\ accepted} * 100$$

Since ID3 has some problem such as not being able to do well with continues value and may get stuck into local optima, another classifier algorithm logistic regression is used to classify the CVs and to compare the results. To classify the CVs, multivariate logistic regression is used. The formula for multivariate logistic regression is,

$$\pi(x) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p}}$$

With low number of CVs, multivariate logistic regression performs worse than decision tree algorithm. However, when number of sample CVs is increased, logistic regression starts to perform better.

This leads to a conclusion that, if there are low amount of sample data, ID 3 gives better result. However, with bigger amount of sample data, logistic regression performs better.

CHAPTER 05

Conclusion and Future Scope

5.1 Conclusion

The proposed model in this paper extracts necessary data from a CV/Resume and segments them based on their values. However the ranking and positive weight given for a CV/Resume might change based on different company or employers preference. As described, the whole process was segmented and each segmented was designed separately to perform its task. The segment that deals with Natural Language Processing actually worked with only the Natural Language Processing task and similarly the segments that deal with Machine Learning, completely deals with Machine Learning techniques. A different way of evaluating and analyzing the data in a CV/Resume is proposed in this paper and that was converting data into HTML code to understand different values. Finally, the model gives ranks to CVs/Resumes based on the necessary data and employers needs taking previous requirements in consideration.

5.2 Future Scope

Even though in the research one of the most feasible way to evaluate a CV/ Resume was detailed, the domain was kept restricted to the CVs/ Resumes of only engineering students and the amount of sample data versus the amount of test data was relatively small. In addition to that, CVs/ Resumes with some varied layout design is out of the scope of this paper. For the future scope of this research, the methodologies can be used for the data from CVs/ Resumes of other job departments or the whole research can be done in a much larger scope. Additionally, algorithms such as naive Bayes, logistic regression or c4.5 analysis [25] can be performed to see if it improves the result. Therefore, the future scope is very broad.

References

[1] Westermann, F., Wei, J. S., Ringner, M., Saal, L. H., Berthold, F., Schwab, M., Khan, J. (2002). Classification and diagnostic prediction of pediatric cancers using gene expression profiling and artificial neural networks. GBM Annual Fall meeting Halle 2002,2002(Fall).

[2] Ryland, E. K., & Rosen, B. (1987). Personnel Professionals Reactions to Chronological and Functional Résumé Formats [Abstract]. The Career Development Quarterly, 35(3), 228-238.

[3] Malamitsa, K., Kokkotas, P., amp; Kasoctas, M. (december 2008). Graph/Chart Interpretation and Reading Comprehension as Critical Thinking Skills. Science Education International, 19(4)

[4] “Language-Check 0.8: Python Package Index,” Pypi.python.org. N.p., 2016. Web. 17 Apr. 2016.

[5] “Machine Learning: What It Is and Why It Matters,” Sas.com. N.p., 2016. Web. 17 Apr. 2016.

[6] Kleinbaum, D. G., & Klein, M. (2010). Analysis of matched data using logistic regression. In *Logistic regression* (pp. 389-428). Springer New York.

[7] McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, pp. 41-48).

[8] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81-106.

[9] Breslow, N. E., & Day, N. E. (1998). Statistical methods in cancer research. Lyon: International Agency for Research on Cancer.

[10] Jin, C., De-Lin, L., & Fen-Xiang, M. (2009, July). An improved ID3 decision tree algorithm. In *Computer Science & Education, 2009. ICCSE'09. 4th International Conference on* (pp. 127-130). IEEE.

[11] Korting, T. S. (2006). C4. 5 algorithm and multivariate decision trees. *Image Processing Division, National Institute for Space Research–INPE Sao Jose dos Campos–SP, Brazil*.

[12] “An Overview of Current Research on Automated Essay Grading”, S. Valenti, F. Neri and A. CucchiarelliBlood, Ian. "Automated Essay Scoring: A Literature Review". Working Papers in TESOL and AppliedLinguistics 11.(2011) (2016): n. pag. Web. 16 Apr. 2016.

[13] Automated Essay Grading" P. Reddy and G. Jambagi, University of Illinois, Chicago, USA, 2003"Machine Learning: What It Is and Why It Matters," Sas.com. N.p., 2016. Web. 17 Apr. 2016.

[14] Ramakrishnan, C., Patnia, A., Hovy, E., & Burns, G. A. (2012). Layout-aware text extraction from full-text PDF of scientific articles. Source code for biology and medicine, 7(1), 7.

[15] Richardson, L. (n.d.). Beautiful Soup Documentation.http://www.crummy.com/software/BeautifulSoup/bs4/doc/, accessed on 04/11/2017

[16] Helmut Berger and Dieter Merkl, A Comparison of Text-CategorizationMethods Applied to N-gram Frequency Statistics, InAustralian JointConference on Artificial Intelligence, 2004

[17] Cornelißen,T,(2008).The Interaction of Job Satisfaction, Job Search, and Job Changes. An Empirical Investigation with German Panel Data, Published online: 18 March 2008Springer Science+Business Media B.V. 2008.

[18] Albus, J. E., Anderson, R. H., Brayer, J. M., DeMori, R., Feng, H. Y., Horowitz, S. L., ... & Vamos, T. (2012). Syntactic pattern recognition, applications (Vol. 14). Springer Science & Business Media.

[19] Bray, T. (2014). The javascript object notation (json) data interchange format.

[20] Cios, K. J., & Sztandera, L. M. (1992, March). Continuous ID3 algorithm with fuzzy entropy measures. In Fuzzy Systems, 1992., IEEE International Conference on (pp. 469-476). IEEE.

[21] Ichihashi, H., Shirai, T., Nagasaka, K., & Miyoshi, T. (1996). Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning. Fuzzy sets and systems, 81(1), 157-167.

[22] Walt, S. V. D., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22-30.

[23] Jones, E., Oliphant, T., & Peterson, P. (2014). {SciPy}: open source scientific tools for {Python}.

[24] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427-437.

[25] Ruggieri, S. (2002). Efficient C4. 5 [classification algorithm]. *IEEE transactions on knowledge and data engineering*, *14*(2), 438-444.