**JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY**



**Artificial Intelligence Lab**

# PROJECT REPORT

**TOPIC:** Resume Parser using NLTK and Python

Faculty **- Dr. Ankita Verma**

# INTRODUCTION

In this project, we have implemented a resume parser where important keywords are fetched in a resume using python for ease to the HR interviewer.

* The script uses the natural language processing library NLTK to extract the information. It also uses the PDFminer library to extract text from a PDF file.
* The extract\_text\_from\_pdf function is used to extract text from a PDF file, and the extract\_names function extracts names from the text using named entity recognition.
* It also uses NLTK's stop words and bigram/trigram generation algorithms to extract skills from the resume.
* Overall, the code uses a combination of regular expressions and NLP techniques to extract relevant information from a resume.

# PROBLEM STATEMENT

With the rapid growth in online-based recruiting systems, candidates apply for jobs on web portals by uploading their resumes. Due to internet-based recruiting systems, candidates participate in large volumes; hence, it becomes a challenge for the recruiter to filter candidates for the required role. The resumes uploaded by the candidate are varied in format such as font, color, font size, etc. and it is difficult for the recruiters to find the best match for a job role. Natural Language Processing (NLP) helps to deal with such problems and helps recruiters to extract detailed information of the candidates required to carry forward their candidature. In this work, we propose to use *named entity recognition* of the Stanford CoreNLP system to extract information relevant for the recruiting process. Moreover, based on skill set of candidates, the resume of the candidate is assigned a genre such as Computer Science, Statistics, Business Development etc. In this paper we propose to design an intelligent resume parser system capable of converting the unstructured data into a structured format which enables the recruiter to filter the right candidates for the desired job role.

# OBJECTIVE AND SCOPE OF PROJECT

The objective of a Resume Parser is to replace slow and expensive human processing of resumes with extremely fast and cost-effective software. A Resume Parser allows businesses to eliminate the slow and error-prone process of having humans hand-enter resume data into recruitment systems. A Resume Parser classifies the resume data and outputs it into a format that can then be stored easily and automatically in a database or ATS or CRM.

By using a Resume Parser, a resume can be stored into the recruitment database in real-time, within seconds of when the candidate submitted the resume.

# METHODOLOGY

The functionality of the code includes:

Converting resume into plain text which includes extracting text from PDF files

* . Extracting fields from resumes
  + Extracting names from resumes
  + Extracting phone numbers from resumes
  + Extracting email addresses from resumes
  + Extracting skills from the resumes
  + Extracting education and schools from resumes.
* We’ll use [Python 3](https://www.python.org/) for its wide range of libraries that is already available and for its general acceptance in the data sciences area.
* We’ll also be using [nltk](https://www.nltk.org/" \t "_blank) for NLP (natural language processing) tasks such as stop word filtering and tokenization, [docx2txt](https://github.com/ankushshah89/python-docx2txt) and [pdfminer.six](https://github.com/pdfminer/pdfminer.six" \t "_blank) for extracting text from MS Word and PDF formats.
* We assume you already have Python3, pip3 on your system and possibly using the marvels of virtualenv. We’ll not get into details for installing those. We also assume you are running on a Posix based system such as Linux (Debian based) or macOS.

# TECHNOLOGY USED

* **Python**
* **NLTK(Natural Language Toolkit)**
* **PDFminer**

# NLTK

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interface, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

# PDFMINER

PDFMiner is a text extraction tool for PDF documents.

**CONCLUSION**

Resume parsing is tricky. There are hundreds of ways of doing it. We’ve just covered one easy way of doing it and unfortunately do not expect miracles. It may work for some layouts and otherwise for some

# REFERENCES

**https://www.geeksforgeeks.org/tokenize-text-using-nltk-python/**

**https://www.geeksforgeeks.org/introduction-to-nltk-tokenization-stemming-lemmatization-pos-tagging/**