**Resume Parser in AI using NLP**

[**Abstract**](https://psychology.ucsd.edu/undergraduate-program/undergraduate-resources/academic-writing-resources/writing-research-papers/research-paper-structure.html)**:**

A resume parser is a deep learning/AI framework that identifies complete information from resumes, analyses, store, organize, and enriches it through its taxonomies. Resume parsing software makes the hiring process quick and more productive.

In many major companies, recruiter doesn’t have time to view or read full resume to select the person for a specific domain or skill. So, for that we designed a system where user upload resume docx file of person and required skills or education and in returns system returns the how much skills and education matches with required skills and education.

Resume parsers are already standard in most mid- to large-sized companies and this trend will continue as the parsers become even more affordable.

A qualified candidate's resume can be ignored if it is not formatted the proper way or doesn't contain specific keywords or phrases. As Machine Learning and Natural Language Processing get better, so will the accuracy of resume parsers.

One of the areas resume parsing software is working on expanding into is performing contextual analysis on the information in the resume rather than purely extracting it. One employee at a parsing company said “a parser needs to classify data, enrich it with knowledge from other sources, normalize data so it can be used for analysis and allow for better searching.”

This entire system is based on The **Natural Language Toolkit** (NLTK) which is an open-source Python library for Natural Language Processing.

In order to achieve the desired goal, the entire process divided in 3 basic segments. The first segment consists function which extracts the text from given document file. And in second segment there is a function which extracts the skills and education from the extracted words from the document. And finally compares with given keywords of skills and education to founded keywords. And returns how much that candidate matches with recruiter’s eligibility.

Doc2txt

doc2text extracts higher quality text by fixing common scan errors Developing text corpora can be hard for everyone. Much of the text data we are interested in as scientists are locked away in pdfs that are poorly scanned. These scans can be off kilter, poor resolution, have a hand in them… and if you OCR these scans without fixing these errors, the OCR doesn’t turn out so well. doc2text was created to help researchers fix these errors and extract the highest quality text from their pdfs as possible.

NLTK

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, 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.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.

Features used from NLTK Library

n-grams: N-grams analyses are often used to see which words often show up together. An n-gram is a contiguous sequence of n items from a given sample of text or speech. In the text analysis, it is often a good practice to filter out some stopwords, which are the most common words but do not have significant contextual meaning in a sentence (e.g., “a”, “the”, “and”, “but”, and so on). NLTK provides us a list of such stopwords. We can also add customized stopwords to the list. For example, here we added the word “though”