Main problem

A client company is looking for employees. Employees are looking for jobs. Employees send in a resume and the company sends in a job description. Extract exact which resume matches the job description the best.

Main topics for the resume

* Skillset
* Qualifications
* Work Experience
* Career objectives

Main topics for the Jary (free flow text from company)

* Skill set
* Experience
* Location
* Responsibilities
* Role

Flow plan

1. Create a dataset of skill sets and similar skills to each using dice
   1. <https://www.dice.com/skills>
   2. Use firebase database
   3. Use web scraping → robotic process automation web scraping
2. Connect the 50 resumes to the firebase database
   1. Be able to make database dynamic with easier to add
3. Using NLP analysis resume getting basic understand
   1. Create new database where we can save this preliminary data
4. Build on the main topics of the resume and connect them to the generated topics
   1. This will be the basics of understanding
5. Connect the website to the python code
   1. With this we must also connect the database
6. Develop website to upload documents to database
7. Connect main topics of the Jary
8. Generate a SVD for understanding the topics (unstructured data)
9. Use and Bayes Theorem and Naive Bayes to generate a basic outline of both documents
10. Using transfer learning build a running base model (Spacy)
11. Optimize model for speed and accuracy

Emails:

* [siddhantmane@gmail.com](mailto:siddhantmane@gmail.com)
* amank26@outlook.com

What we need to learn:

* Uploading word documents to html website
* Running python on the website

Important Links

* Github
  + <https://github.com/Kunal2341/ClientResumeMatching>

Let's do the following tasks → TIMELINE IS STILL TBD

**TASK 1**

* Build a base website → Sidd
* Build a form on the website that allows you to upload some information (Maybe add more) → Probably using JavaScript → Sidd
  + Name
  + Phone Number
  + Email
  + Document
    - Any sort of word document
    - <https://aashni.me/blog/using-firebase-to-upload-files/#:~:text=Connect%20to%20Firebase&text=Once%20you're%20on%20the,closing%20tag.>
* Create database and store data created → Aman and Sidd
  + [Connecting Firebase to a Contact Form](https://www.youtube.com/watch?v=PP4Tr0l08NE)
* Run python code on HTML
  + create an example list → Aman
  + Save file and call code on website→ Aman
  + Transfer the data created in py file to HTML and JS → Sidd

**TASK 2**

* Create Database to store all the skills→ Aman
* Web scrape all skills from dice.com → Kunal
* Web-scrape connected skills also and add to database→ Kunal
  + For each of the 7102 skills
    - there is around 5 close related skills
    - there is around 25 distant related skills
* Store all data to database → Kunal and Aman
* Be able to scan through database using python → Kunal
* Connect this database to the website (maybe display) → → Kunal and Aman and Sidd

**TASK 3**

* Scan through Resumes and extract
  + Name (Basic Py)→ Kunal and Aman
  + Email (Basic Py)→ Kunal and Aman
  + Phone Number (Basic Py)→ Kunal and Aman
  + Skillset → Kunal
  + Qualifications→ Kunal
  + Work Experience→ Kunal
  + Career objectives → Kunal

NLP Details

* Sentences
  + Split into tokens
  + Using the defining linguistic features to detect specific words
  + Get needed words from dataset and using spacy scan each token
  + Scan if the defining features match
    - Text: The original word text.
    - Lemma: The base form of the word.
    - POS: The simple UPOS part-of-speech tag.
    - Tag: The detailed part-of-speech tag.
    - Dep: Syntactic dependency, i.e. the relation between tokens.
    - Shape: The word shape – capitalization, punctuation, digits.
    - is alpha: Is the token an alpha character?
    - is stop: Is the token part of a stop list, i.e. the most common words of the language?
  + Detect if skill is detect and if correct
  + Problems faced
    - Misses a lot of the things
    - Write code to create a benchmark to check if it works and understand the many different occasions this can happen and re add it into the code
  + Start editing own model
    - Use different previous examples as data