**Job Description vs Resume Matching**

Table of Contents

[1 Steps 2](#_Toc56002057)

[2 Points and Discussions 3](#_Toc56002058)

[3 Matching Score Database 4](#_Toc56002059)

# Steps

1. Fetch resume and job data from database
2. Preprocess resume details and job description(remove stopwords, punctuation and special symbols)

Output: Preprocessed resume details and job description in the form of list of words (resume\_details\_tokens, job\_description\_tokens)

1. Extract skills in resume details and job description

Initialize skills\_in\_resume = [], skills\_in\_job\_description = []

for word in resume\_details\_tokens:

if word in skills.csv:

add word in skills\_in\_resume

for word in job\_description\_tokens:

if word in skills.csv:

add word in skills\_in\_job\_description

1. Count matched skills in resume with job and find percentage of job matched

Initialize matched\_skill\_count = 0

for skill in skills\_in\_resume:

if skill in skills\_in\_job\_description:

incerement matched\_skill\_count

skill\_percentage = (matched\_skill\_count / length(skills\_in\_job\_description))\*100

1. Find the skill appear maximum number of times in resume detail and match it with job title to find similarity score
2. Find the average of skill\_percentage\_matched in step 4 and similarity score in step 5

Overall\_skill\_score = (skill\_percentage\_matched + similarity\_score)/2

1. Find similarity score between resume title and job title (title\_matched\_score)
2. Find distance between location (loc\_distance)
3. Save overall\_skill\_score, title\_matched\_score, loc\_distance with resume\_id and job\_id
4. Repeat from step 3 to 7 for every resume with every job
5. Sort first on the basis of title\_matched\_score then overall\_skill\_score and then loc\_distance

# Points and Discussions

For Skills,   
there is a direct match i.e. word to word. Hence +1 for each match.   
  
For locations  
We need to build logic for the same i.e. exact same location we will have more score, as distance increases (or city change), the score will decrease, and stops when there is country change.   
  
For job Titles  
There would be an approximate match, closer approximate will have more score and less approximate match.   
  
Skill vs location vs job title  
Which should be given priority and under what conditions, this is still open. Thinking about it. Please suggest if possible.

For MVP, i think going with skill count is OK, but later we need to do deeper skill score calculation. For skills, there are 2 things - firstly, does that skill exist in resume and/or job description. The other is the weightage of skill i.e if the count of same skill is multiple time, it has more weightage for e.g. if the python skill/keyword appears more than 5 times as compared to perl skill/keyword that appears only 1 time - it would mean the candidate is more proficient in python and it should match with job description proficiency. Hope this makes sense.  
  
With regards to priority, the below will take precedence:

1. Job Title (this has to be a good match)
2. Skills (this too has to be a fair match)
3. Location (this could be loose match as some candidates are OK to relocate - but again within a country)

# Matching Score Database

there is a table called as "job\_person\_matches" table that you can use to populate the details. You can create additional fields as necessary. here's what needs to be populated:  
  
type → should be defaulted to "Job-Resume-Match"  
source → "Backend"  
name → Job Title  
status → "Active"  
job\_id → Job ID  
resume\_id → Resume ID  
person\_id → Person ID (from resume table)  
tenant\_id → Tenant\_ID from jobs table (if available)  
  
You will need to create additional columns for matching results.

