ISBN: 978-1-948012-17-1

Ethics and Information Technology (ETIT)

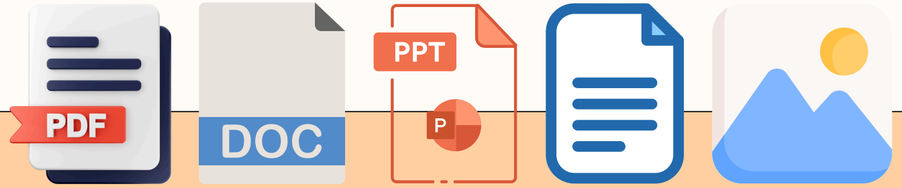
# AN AUTOMATED RESUME SCREENING SYSTEM USING NATURAL LANGUAGE PROCESSING AND SIMILARITY

Automated Resume Screening

ISBN: 978-1-948012-17-1

# Introduction

- increase in internet connectivity, recruitment process change
  - E-recruitment: provided convenience
  - dilemma: large companies/ agencies receive thousands of unformatted resumes daily (no standard)



ISBN: 978-1-948012-17-1

# Introduction

- only around **5**% gets accepted
  - **pointless** to scan all
  - time consuming to do so
- temporary suggestion:
  - job portals make job seekers fill out their resume **manually** in an **online structured form**
    - tedious for the applicants
    - not domain specific

ISBN: 978-1-948012-17-1

# Introduction

keyword-based search: ×

- for shortlisting candidates
  - insufficient in matching candidates with job desc.
- relies on existence of certain keywords
  - give irrelevant results
    - deserving candidates miss out

ISBN: 978-1-948012-17-1

# Introduction

Proposed solution:

- Relate applicants' profile features with defined job descriptions
- two main phases:
  - ending: ranked list of applicants

## 1 First Phase

- all **relevant** candidate **information** is **extracted** from the **unstructured** text in the resumes.
- **NLP** to parse then summarize

## 2 Second Phase

- **resume ranking**: content vs job description
- documents as **vectors (VSM)**
- best fitting: **cosine similarity**

ISBN: 978-1-948012-17-1

# RRL

The **recruitment process** in today's world has witnessed a **major change** with the evolution of technologies like the **Internet**. Contains literary work performed in this domain of **E-recruitment** systems.

ISBN: 978-1-948012-17-1

# RRL

## 1 EXPERT (Kumaran, V.S. and Sankar, A., 2013)

- proposed the use of ontology mapping for screening candidates for the given job description.
  - three phases of operation:
    - creation of candidate ontology
    - construction of job criteria ontology document
    - mapping of both of these to evaluate which candidates are eligible for the job

ISBN: 978-1-948012-17-1

# RRL

***ontology**: formal, explicit, specification of a **shared** conceptualization (Tom Gruber)*

**Ontology mapping**:

- process of establishing **relationships** between concepts in different ontologies to facilitate data **integration** and **interoperability**
- enables systems to **understand** and reconcile **differences** in **semantics**, allowing for more effective data **sharing** across various domains and applications.
- links **related** entities and concepts from **diverse** sources.

ISBN: 978-1-948012-17-1

# RRL

## 2 Automated Job Screening System (Faliagka, Ramantas, Tsakalidis, & Tzimas, 2012)

- discusses different **machine learning** algorithms
  - uses **Support Vector Regression** to create a **list** of ranked **candidates** for the given job.

## 3 Social Media Presence (Weathington and Bechtel, 2012)

- described how **social media** (e.g. LinkedIn, Facebook, etc.) information of the applicants can be used for **recruitment decisions**.

ISBN: 978-1-948012-17-1

# RRL

Proposed solution:

- **different** approach
- focuses mainly on the **content** of the **resumes**
  - extraction of **skills** and related parameters to **match** candidates with the **job descriptions**

ISBN: 978-1-948012-17-1


# Methodology

2 phases:

- Phase 1
  - 5 parts
- Phase 2
  - 3 parts

**Phase Out**

ISBN: 978-1-948012-17-1

# Phase 1

## *Information extraction*

ISBN: 978-1-948012-17-1

# Phase 1 *Information extraction*

- involves **information extraction** using NLP
  - not present in a structured format:
    - noises, inconsistencies, and irrelevant bits of data
- **objective**: derive relevant **keywords** from the unstructured textual data in the resume without human intervention
  - uses Tokenization, Stemming, POS Tagging, Chunking, and Named Entity Recognition
- output - summary **JSON** format: important job-related content (skills, experience, education, etc.)

ISBN: 978-1-948012-17-1

# Phase 1.1 *Tokenization*

- **after** text **conversion** of resumes
- identifying **terms** or **words** that form up a character **sequence**
  - words -> **original** meaning
- dividing **big** chunks of text into **smaller** parts (i.e. tokens)
  - characters like **whitespaces** and **punctuations** are **removed** or isolated.
- Tokens are **sentences** initially and then are further split into individual **words**

ISBN: 978-1-948012-17-1

# Phase 1.1 *Tokenization*

- **Derive** information like the **number** of words in a text, **frequency** of a particular **word** in the text
- Tokenization is a **mandatory step** for further **text processing** such as removal of **stop words**, **stemming** and **lemmatization**.

ISBN: 978-1-948012-17-1

# Phase 1.2 *Stemming and Lemmatization*

- Example -implement, implemented and implementing
  - are just different tenses of the same verb
- For reduction of derived forms to central base and those with similar meanings are not considered different
- Stemming and lemmatization: same objective, different approach

ISBN: 978-1-948012-17-1

# Phase 1.2 *Stemming and Lemmatization*

“Stemming is the **mechanism** of reducing inflected or derived words to their word root, or **stem**. It is a crude heuristic process that involves chopping off the ends of words to achieve this objective, and often includes the removal of derivational affixes” (Jivani, A.G., 2011).

ISBN: 978-1-948012-17-1

# Phase 1.2 *Stemming and Lemmatization*

**Stemming**:

- rule-based
- word **tested** on a range of **conditions** and then based on a **list** of known suffixes, decides how to **cut** it down
- the root derived after stemming may **not** be identical to the morphological root of the word.
- *issues: under-stemming, over-stemming*

ISBN: 978-1-948012-17-1

# Phase 1.2 *Stemming and Lemmatization*

**Lemmatization** :

- process of utilizing a **language dictionary** to perform an **accurate** reduction to root words
- uses language **vocabulary** and **morphological** analysis of words to give **linguistically** correct lemmas.
- utilizes the knowledge of **context**
  - can differentiate between words that have **different** meanings based on **parts** of **speech**.



For the English language, our system uses the **WordNet Lemmatizer** (based on WordNew Database) provided by the NLTK python package.

ISBN: 978-1-948012-17-1

# Phase 1.3 *Parts of speech (POS) tagging*

- process of **assigning** grammatical **information** to a **word** based on its **context** and its **relationship** with other words in the **sentence** (Gelbukh, 2014).
- The **part-of-speech tag** specifies whether the word is a noun, pronoun, verb, adjective according to its usage
- tag -> sentence meaning -> knowledge graphs
  - a word may have a different POS based on different contexts in which it is used

ISBN: 978-1-948012-17-1

# Phase 1.3 *Parts of speech (POS) tagging*

- For example:
  - “I am building a software”
    - building is a Verb
  - “I work in the tallest building of that street”,
    - building is a Noun
- also called grammatical tagging or word-category disambiguation
  - a supervised learning solution that analyses the features to label the words after tokenization.



Rule-Based POS tagging, Stochastic POS tagging, and Transformation based tagging are mostly used (Hasan, 2006).

ISBN: 978-1-948012-17-1

# Phase 1.4 *Chunking*

- process that aims to **add** more **structure** to sentences by grouping **short phrases** with POS **tags**.
- chunking **combines** POS tags with regular expressions to give a result as a **set** of **chunk tags** like Noun Phrase (NP), Verb Phrase (VP), etc.
- **Shallow Parsing**: parse tree **construction** that can have a **max 1 level** of information from roots to leaves
  - ensures **more** information than just POS of the word **without** needing to create a full parse tree.



Chunking segments and labels multi-token sequences (Bird, Klein and Loper, 2009), mostly making groups of “noun phrases” that are used for finding named entities.

ISBN: 978-1-948012-17-1

# Phase 1.5 *Named entity recognition*

- information **extraction** technique which **extracts** relevant information by **classifying** chunks of **unorganized** text into **predefined** categories like **names** of persons, companies, contact info, educational credentials, and skills.
- spaCy module: various pre-trained models that can recognize a number of default entities from the content of the documents.

ISBN: 978-1-948012-17-1

# Phase 1.5 *Named entity recognition*

- **trained** the model on a **large** annotated **set** of resume samples for better **accuracy** in the entity recognition.
  - could detect entities like name, phone number, email, educational institute, organization

ISBN: 978-1-948012-17-1

# Phase 2

## *Content based candidate recommendation*

ISBN: 978-1-948012-17-1

# Phase 2 *Content based candidate recommendation*

- utilizes the **extracted** entities from **Phase 1** to **recommend** the most **appropriate** resumes for the given job description.
- employs concepts like **Vectorization**, importance or **weight** assigning techniques (**TF-IDF**) and similarity measures like **cosine** distance for calculating the **similarity** among the contents of the documents.

ISBN: 978-1-948012-17-1

# Phase 2.1 *Vectorization*

- an **algebraic model** for representing text information for Information Retrieval, NLP and Text Mining.
- process of turning a **document** into a **numerical vector**
- most machine learning models require the input to be **numerical vectors** rather than strings.
- common way of vectorizing text: **map** every possible **word** to a specific **integer**. If we have a large array then every word fits into a **unique** slot in the array

ISBN: 978-1-948012-17-1

# Phase 2.2 *TF-IDF*

- "**Term Frequency – Inverse Document Frequency**"
- invented for information **retrieval** and document search
  - used in text mining techniques
- **weight**: numerical measure to determine **term importance** to a document in a **collection** or corpus.
- high importance, high frequency of a word within the document but is **offset** by the number of documents that contain the word.



So, terms that are frequently used in every document, such as this, and, what, whom, is, the, if, etc. rank low even though they may appear many times since they don't mean much to that document in particular (Stecanella, 2020)

ISBN: 978-1-948012-17-1

# Phase 2.2 *TF-IDF*

- The **TF-IDF** value for a term in a document is calculated by multiplying two different metrics (Stecanella, 2020)

$$TF - IDF\ (t,d) = TF\ (t,d) * IDF\ (t,d)$$

**Term Frequency:**

$$TF\ (t,d) = \frac{freq\ (t,d)}{\sum_i^n freq\ (t_i,d)}$$

**Inverse Document Frequency:** word importance/ rareness

$$IDF(t) = log\left(\frac{N}{count(t)}\right)$$

ISBN: 978-1-948012-17-1

# Phase 2.3 *Cosine similarity*

- metric that **determines** how **much** the two objects are **alike**.
- Cosine similarity: find how **similar** the two documents are **regardless** of their **size**.
  - It represents the orientation of the documents
  - a symmetrical algorithm,
  - results from computing (**equal**):
    - similarity of item X:Y
    - similarity of item Y:X

$$cos(\theta) = \frac{\vec{a}.\vec{b}}{\left\|\vec{a}\right\|\left\|\vec{b}\right\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}$$

ISBN: 978-1-948012-17-1

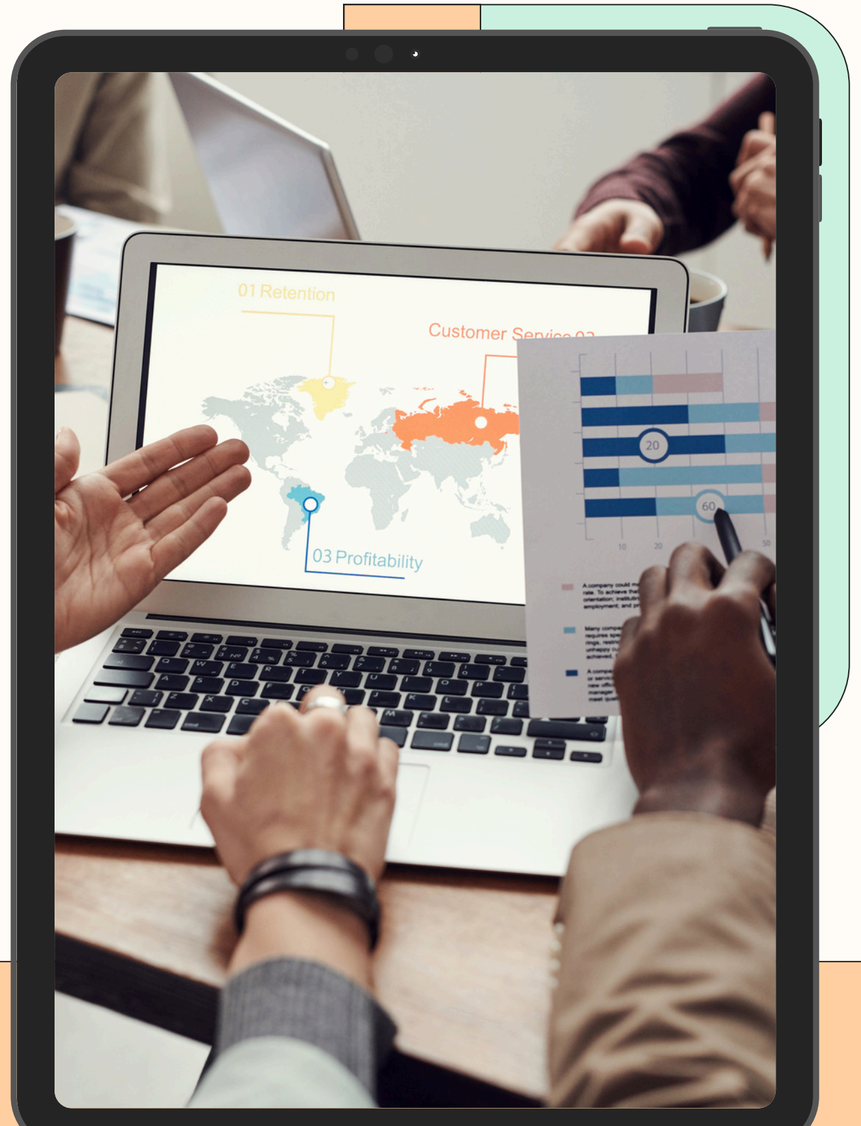
# Phase 2.3 *Cosine similarity*

- **rank** the resume documents with respect to a given **vector** of query words.
- cosine similarity focuses on features that are related to the **text's words only** and will give **less** accurate results.
- The efficiency of similarity measures can be improved by the **inclusion** of **semantic information**.
    - This will constitute the future scope of the automated resume screening system

ISBN: 978-1-948012-17-1

# SYSTEM ARCHITECTURE

**solution**: build a **content-based job** recommendation system that uses the (**VSM**) in calculating the **similarity** between the **content** of the candidate **resumes** and the **job** requirements to recommend the best fitting candidates to the **employer**.



ISBN: 978-1-948012-17-1

# SYSTEM ARCHITECTURE

## Vector Space Model

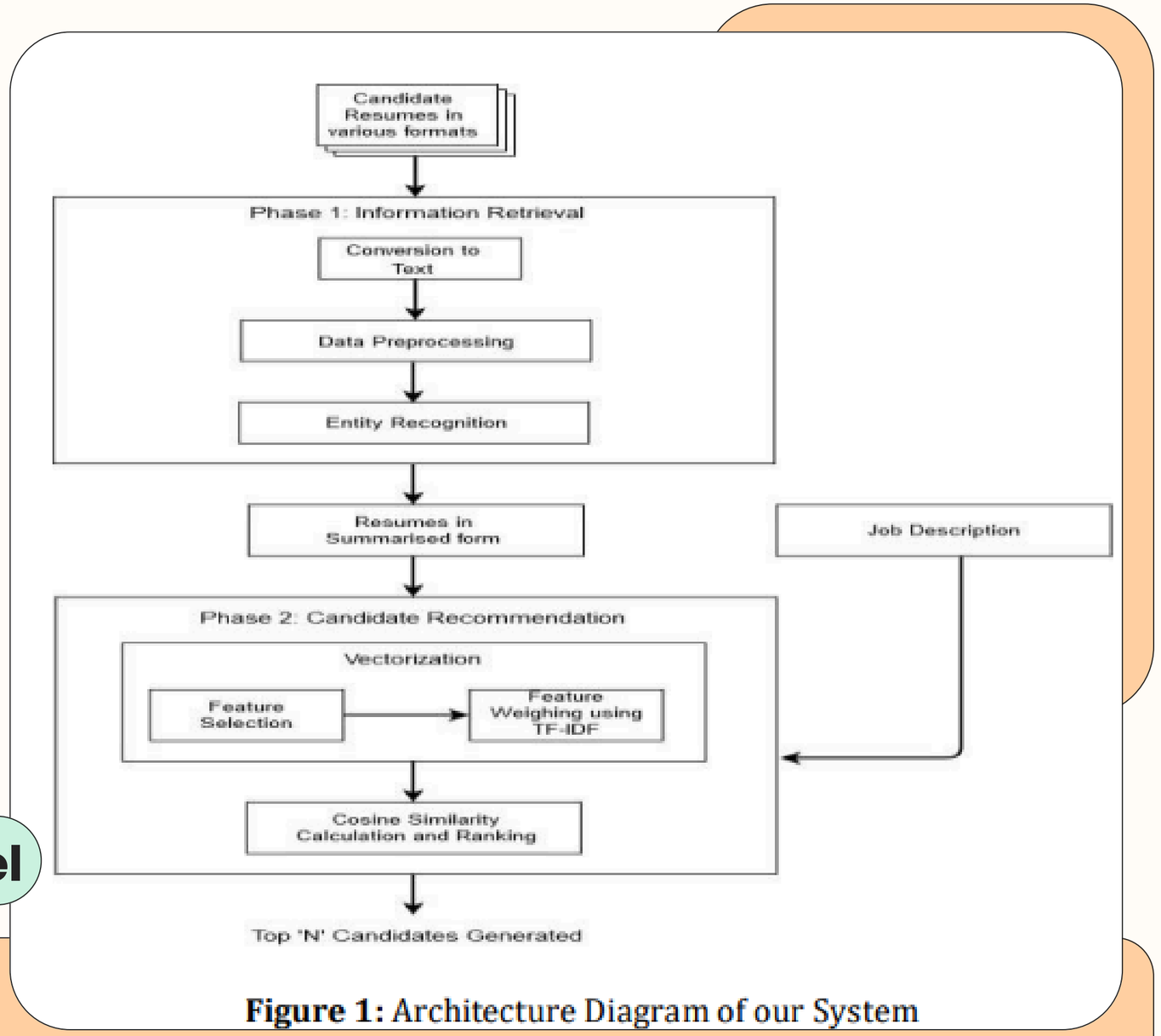



**Figure 1:** Architecture Diagram of our System

ISBN: 978-1-948012-17-1

# Results Phase 1

For testing the system

- **Amazon.com Inc.** job description for a Software Developer Engineer at its Bengaluru office.
- relevant **resume samples** from the Internet

**RESULTS ARE OUT NOW!**


```
{
 "name": "CHIRAG DARYANI",
 "email": "chiragdaryani28@gmail.com",
 "mobile_number": "9977777777",
 "skills": [
  "Debugging", "C", "Database", "Advertising", "Content", "Api",
  "Sql", "Html5", "Certification", "Js", "Javascript", "Java",
  "Android", "Technical skills", "C++", "Nltk", "Pandas",
  "Matplotlib", "System", "Programming", "Numpy", "Algorithms",
  "Analysis", "Mysql", "Spring", "JSF", "JDBC", "Writing", "Html",
  "Opencv", "Python", "R", "Textblob", "Css", "Testing", "Technical"
 ],
 "college_name": [
  "Medi-Caps University, Indore"
 ],
 "degree": [
  "Bachelor of Engineering – Computer Science"
 ],
 "designation": [
  "Associate Professional Product Developer "
 ],
 "experience": [
  "09/19 – Present",
  "Indore, MP, India",
  "COMPUTER SCIENCES CORPORATION",
  "Associate Professional Product Developer – Insurance domain (Projects: USA – Wilton
RE, Americo)",
  "Involved in coding, testing phases of software development life cycle (using Spring, JSF
and JDBC",
  "templates), as well implemented new functionalities based on requirements gathered.",
  "Collaborated with technical team members to integrate back-end and front-end
elements.",
  "Fixing bugs reported by users and took care of enhancements suggested by customers"
 ],
 "company_names": [
  "COMPUTER SCIENCES CORPORATION"
 ],
 "no_of_pages": 1,
 "total_experience": 0
```


Figure 2: This image presents the JSON output generated by our system after the completion of the information extraction process (phase 1) on a sample resume

ISBN: 978-1-948012-17-1

# Results Phase 1

For testing the system

- For feature selection **parameters**:
  - Educational Degree, University, Total Experience, Designation with the Organization in which the candidate has worked in the past, and skills for the job.

**RESULTS ARE OUT NOW!**


```
{
 "name": "CHIRAG DARYANI",
 "email": "chiragdaryani28@gmail.com",
 "mobile_number": "9977777777",
 "skills": [
  "Debugging", "C", "Database", "Advertising", "Content", "Api",
  "Sql", "Html5", "Certification", "Js", "Javascript", "Java",
  "Android", "Technical skills", "C++", "Nltk", "Pandas",
  "Matplotlib", "System", "Programming", "Numpy", "Algorithms",
  "Analysis", "Mysql", "Spring", "JSF", "JDBC", "Writing", "Html",
  "Opencv", "Python", "R", "Textblob", "Css", "Testing", "Technical"
 ],
 "college_name": [
  "Medi-Caps University, Indore"
 ],
 "degree": [
  "Bachelor of Engineering – Computer Science"
 ],
 "designation": [
  "Associate Professional Product Developer "
 ],
 "experience": [
  "09/19 – Present",
  "Indore, MP, India",
  "COMPUTER SCIENCES CORPORATION",
  "Associate Professional Product Developer – Insurance domain (Projects: USA – Wilton
RE, Americo)",
  "Involved in coding, testing phases of software development life cycle (using Spring, JSF
and JDBC",
  "templates), as well implemented new functionalities based on requirements gathered.",
  "Collaborated with technical team members to integrate back-end and front-end
elements.",
  "Fixing bugs reported by users and took care of enhancements suggested by customers"
 ],
 "company_names": [
  "COMPUTER SCIENCES CORPORATION"
 ],
 "no_of_pages": 1,
 "total_experience": 0
```


Figure 2: This image presents the JSON output generated by our system after the completion of the information extraction process (phase 1) on a sample resume

ISBN: 978-1-948012-17-1

The table below presents the ranked list of candidates according to the calculated cosine similarity values.

**Table 1:** Resultant ranked list of candidates prioritized by similarity score

| Candidate Number (Resumes) | Cosine Similarity Score | Rank for the Job |
|---|---|---|
| Candidate 2 | 0.6802823482591744 | 1st |
| Candidate 4 | 0.6514716047844277 | 2nd |
| Candidate 3 | 0.49850131321205904 | 3rd |
| Candidate 1 | 0.4907052756267933 | 4th |

Based on the results,

- **candidate 2** best fits the job posting
- followed by candidate 4
- 3 and 1 are the least appropriate candidates in this sample

# Results Phase 2

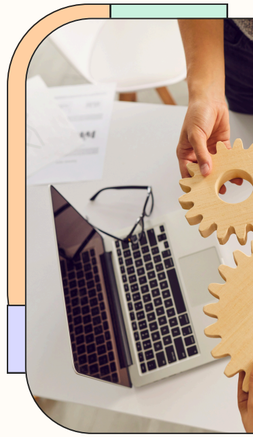
job query is as follows:

cossimilarity(resume1, jobquery, similarity_matrix)= 0.4907052756267933 cossimilarity(resume2, jobquery, similarity_matrix)= 0.6802823482591744 cossimilarity(resume3, jobquery, similarity_matrix)= 0.49850131321205904 cossimilarity(resume4, jobquery, similarity_matrix)= 0.6514716047844277

ISBN: 978-1-948012-17-1

# CONCLUSION

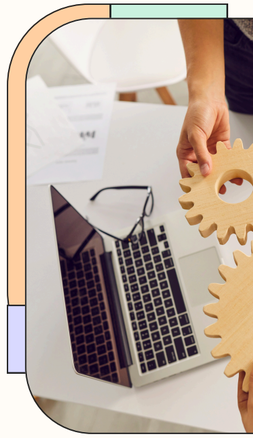
- presented an **automated resume screening** system that simplifies the E-recruitment process by **eliminating** the recruiters' **problems**
- uses **NLP** to extract relevant **information** from the resumes.
- It creates a **summarized** version of each resume which has only the entities that are pertinent to the selection process.
- **Simplified** screening task = **better** analyze each resume with better efficiency



ISBN: 978-1-948012-17-1

# CONCLUSION

- system provides the **provision** of ranking the applicants by using a **content-based** recommendation
  - uses the **Vector Space Model** and similarity
  - matches the **extracted** resume **features** with the **requirements** in the job description
  - calculates the similarity score value for each resume
  - creates a **ranked list** of top-N **recommended** candidates that **best fit** the particular job



ISBN: 978-1-948012-17-1

# Future Works

- mining **social** networking **data**
  - utilizing in combination with **resume content** to for improved recommendations
- using a **collaborative filtering** based approach
  - **match** the current **applicant** with a job according to how well **other** similar candidates are rated for it
- use of Latent Semantic Analysis

ISBN: 978-1-948012-17-1

# Thank You

THANK YOU AGAIN

Thanking you once more, but in a paragraph form. thank you so much for listening!