# CLUSTER-BASED JOB MATCHING SYSTEM

**PHYO PYAE SONE**

**M.C.Sc. JANUARY, 2020**

# CLUSTER-BASED JOB MATCHING SYSTEM

**By**

**PHYO PYAE SONE**

**B.C.Sc.**

**A dissertation submitted in partial fulfilment of the requirements for the degree of**

**Master of Computer Science**

**(M.C.Sc.)**

**UNIVERSITYOF COMPUTERSTUDIES, YANGON**

**JANUARY, 2020**

# ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere thanks to those who helped me with various aspects of conducting research and writing this thesis. To complete this thesis, many things are needed like my hard work as well as the supporting of many people.

First and foremost, I would like to express my deepest gratitude and my thanks to **Dr. Mie Mie Thet Thwin**, Rector, the University of Computer Studies, Yangon, for her kind permission to submit this thesis.

I would like to express my appreciation to **Dr. Thi Thi Soe Nyunt**, Professor, Head of Faculty of Computer Science, the University of Computer Studies, Yangon, for her superior suggestion, administrative supports and encouragement during the academic study.

My thanks and regards go to my supervisor, **Dr. Khine Moe Nwe**, Professor, Faculty of Computer Science, the University of Computer Studies, Yangon, for her support, guidance, supervision, patience and encouragement during the period of study towards completion of this thesis.

I also wish to express my deepest gratitude to **Daw Aye Chan Nyein**, Lecturer, the Department of Language, the University of Computer Studies, Yangon, for her editing this thesis from the language point of view.

Moreover, I would like to extend my thanks to all my teachers who taught me throughout the master’s degree course and my friends for their cooperation.

I especially thank to my parents, all of my colleagues, and friends for their encouragement and help during my thesis.

# STATEMENT OF ORIGINALITY

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

--------------------------------            ----------------------------------

Date            Phyo Pyae Sone

# ABSTRACT

A recruitment system comprises the processes, routines and elements essential to speedy and effective hiring for an organization or for a company. A good recruitment system includes all necessary features to aid management in hiring the best candidates for open positions. An "e-Recruitment" system is also available for recruiters to advertise jobs online where the applicants fill a form online and send or post their profiles. It is also a strong and effective recruitment system. However, there are many difficulties such as time-consuming and the lack of relevancy of job-matching. Online job recruitment platform is one of the most prominent channels for both job seekers and recruiters to hunt jobs and find suitable employees respectively. In the traditional job matching process, manually scanning the resume or profile of a job seeker and matching the resume of job seekers and requirements of job recruiters takes time-consuming and makes difficulties for both seekers and recruiters. Thus, nowadays, many job recommendation systems and cluster-based job matching systems appear. The studies applied k-means clustering for providing the similar clusters of data but gives less relevant data. This system has implemented a job matching system using k-means and word2vec that is to output the clusters with semantically similar words. As a result, using k-means clustering and word2vec model, recruiters can get the most relevant job seekers that fit employers' needs than k-means clustering only. And then, for the relevancy of job matching between job seekers and job recruiters, the classification accuracy method for the relevancy has been used in this system.

# TABLE OF CONENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF EQUATIONS

# CHAPTER 1
# INTRODUCTION

The development of information technology has extraordinarily improved employment enlistment to automate work enrollment framework advanced from the since quite a while ago existed prototypical framework. This work has structured and executed an automated activity enrollment framework. This gives easy to use intuitive programming condition joined with esteem included administrations like precise outcome handling framework, the organized framework examination and structure strategy that was received in planning the mechanized enlistment framework.

Because of the constrained of ability pool and the experience of enrollment specialists, programmed looking through frameworks that can locate the correct competitors become essential. This framework proposes a vocation enlistment framework dependent on k-implies grouping and word2vec model that give an increasingly exact resume and occupation coordinating calculation.

Bunching implies the demonstration of parceling, for example, an unlabeled dataset into gatherings of comparative articles. Each gathering, called a 'bunch', comprises of articles that are comparative among themselves and not at all like objects of different gatherings. In the previous scarcely any decades, bunch investigation has assumed a focal job in an assortment of fields extending from designing (AI, man-made consciousness, design acknowledgment, mechanical building, electrical building), PC sciences (web mining, spatial database examination, printed record assortment, picture division).

It offers a robotized enrolling process that decreases the requirement for manual procedures and administrative work. Right now, searcher dataset loads and embeds into database as preparing information. At the point when work spotter posts new position opening, the characteristics of employment searchers and occupation enrollment specialists that need to change vectors are recovered and changed into vectors utilizing word2vec model. Semantic vector esteems that word2vec assesses are bunched utilizing k-implies grouping. Besides, the framework suggests the rundown of the most reasonable competitors positioning for spotters once they post their enlistments [1].

## 1.1. Related Works

In the Job Recommendation System [5], the significant test of bunching is proficiently important gatherings that succinctly explained. Understudies' information bunching is the programmed association of understudies into bunch or gatherings so understudies inside a group have high likeness in contrast with each other, yet are not at all like understudies in different groups. Right now, examination and k-implies calculation use in the field of instruction. Bunching the understudy's information are as indicated by their test marks. The framework can investigate the understudy's information and yield the examination results. The proposed framework accommodates any administration innovation secondary school, which has the connection between understudies' selection test result and their prosperity.

In [6], a powerful methodology, Cluster based Ranking Index is applied for Enhancing Recruitment Process utilizing Text Mining and Machine Learning, January 2017, is applied for separating pertinent words from the resumes utilizing Term Document Matrix. A grouping philosophy has been utilized to locate the comparable resumes. The significance of each word has been determined by the group which makes this framework remarkable. The term report framework portrays the recurrence consider of words as a real part all things considered. In the term record lattice, each line speaks to one resume and every segment speaks to a word and every passage speaks to the recurrence include of a specific word in that specific resume. This framework can create vector structure dependent on recurrence check yet semantic worth can't be considered by Term Document Matrix.

Distinguishing Varying Patterns for Job Search Using Divisive Correlation Clustering Algorithm utilizes Divisive Correlation Clustering Algorithm (DCCA) for making comparative employment designs. This framework powerfully made bunches for client. DCCA can acquire grouping arrangement from work articulation datasets. DCCA can have the option to distinguish bunches containing occupations with comparable variety in example of articulation, without taking the normal number of groups as an information. DCCA is more huge than k-implies calculation from the exhibition examination results. The DCCA calculation keeps grouping until all bunches are containing just emphatically associated sets of employments [9].

## 1.2. Objectives of the Thesis

The main objectives of the thesis are as follow:

- To understand the nature of K-means algorithm and their applications, andto solvevarious optimization problems
- To exploit the use of K-means algorithm in clustering problem to discover appropriate clusters
- To study word2vector model that can evaluate semantic values of words.
- To apply the K-means clustering for a job matching system and to provide more relevant matching between job seekers and recruiters

## 1.3. Overview of the Thesis

Inefficiencies in the job recruitment such as friction in matching members to jobs, and the existence of skill gaps in various sectors of the economy are considered to be major problems facing economies today [2]. The central premise is that increasing the productivity of a member of workforce crucially depends on identifying and recommending skills whose acquisition will yield the highest utility gains for that member. In this end, this system develops a job matching models which can match job profile of recruiter to the most relevant job seekers. The matching step is followed by a skill recommendation step which makes demand-based skill recommendations to members. The extensive quantitative evaluation using adataset comprised of job seekers' profiles and job postings from recruiter suggests that skill recommendations made by the algorithm are highly correlated with skills demanded in held out future jobs. In this thesis, the system is presented to support a cluster-based job matching system for job seekers and job recruiters. The attributes of job seekers and job recruiter (job title, degree, skill) are transformed into semantic vectors using word2vec model. And then, the semantic vectors of job seekers are clustered using k-means clustering into the similar groups of job seekers. Moreover, the job profile of recruiter is also transformed into semantic vectors using word2vec model and matches with the clustered job seeker data by calculating the nearest score between recruiter's job profile and job seekers' profiles.

## 1.4. Organization of the Thesis

In Chapter 1, introduction, objectives of the thesis, overview of the system and organization of the thesis are described.In Chapter 2, the background theory of the proposed system such as clustering, validity measure and clustering are presented.And then, Chapter 3 describesJob Classification using genetic algorithm.Chapter 4 presents design and implementation of the system.In Chapter 5, conclusion, the conclusion, advantages, limitation and further extension of the system are presented.

# CHAPTER 2
# THEORICAL BACKGROUND

## 2.1 Clustering

Grouping targets speaking to huge datasets by a less number of models or bunches. It acquires straightforwardness displaying information and hence assumes a focal job during the time spent information revelation and information mining. In the field of grouping, k-implies calculation is the most prominently utilized calculation to discover a segment that limits mean square blunder (MSE) measure. In spite of the fact that k-implies is a broadly valuable bunching calculation, it experiences a few disadvantages. The target capacity of the k-implies isn't arched and it might contain nearby minima. Subsequently, while limiting the goal work, there is probability of stalling out at nearby minima. The exhibition of the k-implies calculation relies upon the underlying decision of the bunch focuses. Information mining errands, in nowadays, require quick and exact apportioning of tremendous datasets, which may accompany an assortment of characteristics of highlights. Thusly, this forces high computational prerequisites on the applicable grouping strategies. A group of bio-enlivened calculations, understood as Evolutionary Computing has as of late rose that meets these prerequisites and has effectively been applied to various certifiable bunching issues. Average group models include: Connectivity models: for instance, various leveled bunching assembles models dependent on separation network. Bunch examination can be a ground-breaking information digging device for any association that necessities to distinguish discrete gatherings of clients, deals exchanges, or different kinds of practices and things [8].

## 2.2 Clustering Methods

The primary explanation behind having many bunching techniques is the way that the idea of "group" isn't absolutely characterized. Since many bunching strategies have been built up, every one of grouping techniques utilizes an alternate acceptance standard. Bunching techniques can be separated into five gatherings: apportioning grouping and various leveled grouping, thickness based grouping strategies, model-based bunching strategies and framework based grouping techniques.

### 2.2.1 Partitioning Clustering Methods

Partitioning clustering methods relocate instances by moving them from one cluster to another, starting from an initial partitioning. Partitioning methods typically require the number of clusters to be predefined. The most well-known and commonly used partitioning methods are k-means and k-medoids.

**K-Means Method:** The k-means method takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intra cluster similarity is high but the inter cluster similarity is low. Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster’s centroid or center of gravity. The k-means procedure is summarized in Figure 2.1.

```
Algorithm: k-means.

Input:

    k: the number of clusters,
    D: a data set containing n objects.
Output: a set of k clusters.

Method:

    Arbitrary choose k objects from D as the initial cluster centers;
    Repeat
        (Re)assign each object to the cluster to which the object is the
        most
        similar, based on the mean value of the objects in the cluster;
    Update the cluster means, i.e., calculate the mean value of the
            objects for each cluster;

    Until no change;
```

**Figure 2.1 K-means Algorithm**

**K-MedoidsMethod:** Anotherpartitioning method, which attempts to minimize the sum of the square error for all objects in the dataset, is the k-medoids also called Partitioning AroundMedoids (PAM). This method is very similar to the k-means method. It differs from the K-Means Clustering mainly in its representation of the different clusters. Each cluster is represented by the most centric object in the cluster, rather than by the implicit mean that may not belong to the cluster. In general, the algorithm iterates until, eventually, each representative object is actually the medoid, or most centrally located object, of its cluster. This is the basis of the k-medoids method for grouping n objects into k clusters.

```
Algorithm: k-medoids.
Input:
    k: the number of clusters,
    D: a data set containing n objects.
Output: a set of k clusters.

Method:
    Arbitrarily choose k objects in D as the initial representative objects or
    seeds;
    Repeat
    Assign each remaining object to the cluster with the nearest
            representative object;
    Randomly select a no representative object, $o_{random}$;
  Compute the total cost, $s$, of swapping representative object, $o_j$,
            with $o_{random}$;
    If $s$< 0 thenswap $o_j$ with $o_{random}$ to form the new set of k
            representative objects;
    Until no change;
```

**Figure 2.2 K-medoids Algorithm**

## 2.2.2 Hierarchical Clustering Methods

These techniques build the bunches by recursively parceling the examples in either a top-down or base up design. These strategies can be sub-isolated as agglomerative various leveled grouping and troublesome progressive bunching.

Agglomerative Hierarchical Clustering Method: In agglomerative various leveled bunching, each item at first speaks to its very own group. At that point, bunches are effectively converged until the ideal group structure is gotten.

Troublesome Hierarchical Clustering Method: In disruptive various leveled grouping, all articles at first have a place with one bunch. At that point, the group is separated into sub-bunches, which are progressively isolated into their own sub-bunches. This procedure proceeds until the ideal group structure is acquired.

The consequence of the progressive bunching techniques is a dendrogram, speaking to the settled gathering of articles and similitude levels at which groupings change. A grouping of the information objects is acquired by cutting the dendrogram at the ideal comparability level. The blending or division of bunches is performed by some comparability measure, picked in order to advance some basis.

### 2.2.3 Density-based Clustering Methods

Density-based methods assume that the points belong to each clusterthatis drawn from a specific probability distribution. The overall distribution of the data is assumed to be a mixture of several distributions. Some density-based methods are Density-Based Spatial Clustering of Applications with Noise(DBSCAN) and DENsity-based CLUstEring (DENCLUE) methods.

Density-Based Spatial Clustering of Applications with Noise(DBSCAN) Method: The algorithm grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise. It defines a cluster as a maximal set of density-connected points.

DENsity-basedCLUstEring(DENCLUE) Method is a clustering method based on a set of density distribution functions. The method is built on the following ideas:  The influence of each data point can be formally modeled using a mathematical function, is called an influence functionwhich describes the impact of a data point within its neighborhood.The overall density of the data space can be modeled analytically as the sum of the influence function applied to all data points; and clusterscan then be determined mathematically by identifying density attractors, where density attractors are local maxima of the overall density function.

### 2.2.4 Model-based Clustering Methods

Model-based grouping techniques endeavor to streamline the fit between the given information and some scientific models. In contrast to regular bunching, it distinguishes gatherings of items; model-based bunching strategies additionally discover trademark portrayals for each gathering, where each gathering speaks to an idea or class. The most every now and again utilized model-based bunching techniques are choice trees and neural systems strategies.

Choice Trees Method: In choice trees strategy, the information is spoken to by a progressive tree, where each leaf alludes to an idea and contains a probabilistic portrayal of that idea. A few calculations produce order trees for speaking to the unlabeled information.

Neural Networks Method: Neural Networks technique speaks to each group by a neuron or "model". The info information is likewise spoken to by neurons, which are associated with the model neurons. Each such association has a weight, which is found out adaptively during learning. A well-known neural calculation for grouping is oneself arranging map (SOM).

### 2.2.5 Grid-based Clustering Methods

A network based grouping calculation incorporates five essential advances: parceling the information space into a limited number of cells (or making lattice structure), evaluating the cell thickness for every cell, arranging the cells as indicated by their densities, distinguishing bunch focuses, and traversal of neighbor cells. A significant preferred position of lattice based bunching is that it essentially lessens the computational intricacy. Some framework based grouping strategies are Statistical Information Grid (STING) and Wave Cluster.

Measurable Information Grid is a framework based multi goals grouping system in which the spatial zone is separated into rectangular cells. There are typically a few degrees of such rectangular cells comparing to various degrees of goals, and these phones structure a progressive structure. Every cell at a significant level is apportioned to frame various cells at the following lower level.

In Wave Cluster strategy, information focuses are first doled out to a lot of cells of a framework separating the first element space consistently. The size of the lattice shifts comparing to various sizes of change. A discrete wavelet change is then utilized on these cells to delineate information into the new element space, where the groups, spoke to as the associated parts in the space, are identified. Various goals of wavelet change lead to various arrangements of groups.

## 2.3 Distance Measure

An important component of a clustering algorithm is the distance measure between data points. The distance measure determines how the similarity of two elements is calculated. This influence the shape of the clusters, as some elements may be close to one another according to one distance and further away according toanother. Formally, the distance d(x, y) between x and y is considered to be a two argument function satisfying the following conditions:

$d(x, y) \geq 0$ for every x and y

$d(x, x) = 0$ for every x

$d(x, y) = d(y, x)$

The evenness is additionally an undeniable prerequisite. The uniqueness accomplishes a worldwide least when managing two indistinguishable examples that is d (x, x) = 0. Regular separation capacities can be viewed as the Euclidean separation, the Manhattan separation, the Maximum separation, the Minkowski separation, the Mahalanobis separation, the Average separation and different separations. Euclidean separation is presumably the most well-known separation for numerical information. For two information focuses x and y in d-dimensional space, the Euclidean separation between them is characterized to be as appeared in Equation 2.1.

$$D_{euc}(x, y) = \left[\sum_{j=1}^{d}\left(x_j - y_j\right)^2\right]^{\frac{1}{2}} = ||x - y|| \qquad (2.1)$$

Where $x_j$and$y_j$are the values of the j^th attribute of x and y, respectively.

## 2.4Clustering Applications

Grouping has been applied in a wide assortment of fields, for example, building, PC sciences, life and restorative sciences, cosmology and earth sciences, sociologies and financial matters.

Application in Engineering: Typical uses of grouping in building can be gone from biometric acknowledgment and discourse acknowledgment, to radar signal investigation, data pressure, and commotion expulsion.

Application in Computer sciences: Applications of grouping can be framed in web mining, spatial database examination, data recovery, printed report assortment, and picture division.

Application in Life and therapeutic sciences: These zones comprise of the significant uses of grouping in its beginning period and will keep on being one of the fundamental playing fields for bunching calculations. Significant applications incorporate scientific classification definition, quality and protein work ID, ailment analysis and treatment, etc.

Application in Astronomy and earth sciences: Clustering can be utilized to arrange stars and planets, research land developments, parcel areas and urban communities, and study waterway and mountain frameworks.

Application in Social sciences: Interesting applications can be found in personal conduct standard investigation, connection recognizable proof among various societies, development of transformative history of dialects, examination of informal organizations, archeological finding and ancient rarity arrangement, and the investigation of criminal brain science.

Application in Economics: Applications in client attributes and obtaining design acknowledgment, gathering of firms, and stock pattern investigation all profit by the utilization of group examination.

## 2.5 Cluster Validity Measures

Cluster validity is measuring goodness of a clustering relative to others created by other clustering algorithms, or by the same algorithms using different parameter values. Cluster validation is a very important issue in clustering analysis because the result of clustering needs to be validated in most applications. In most clustering algorithms, the number of clusters is set as user parameter. There are a lot of approaches to find the best number of clusters. Intra distance can be computed as shown in Equation 2.2[13].

$$\mathrm{M_{intra}} = \frac{1}{N}\sum_{\mathrm{i}=1}^{\mathrm{k}} \sum_{\mathrm{x}\in \mathrm{C}_i} ||\mathrm{x} - \mathrm{z_i}||^2 \tag{2.2}$$

Where $\mathrm{M_{intra}}$ is the minimum intra distance, N is the number of datasets, k is the number of clusters, the dataset into kclusters$C_1$, $C_2$, . . ., *Ck* $_{and}$z$_i$be the cluster center for cluster $C_i$.

## 2.6 Web Page Clustering

Site page bunching is a sort of archives grouping. The vast majority of the current strategies for record grouping depend on either probabilistic techniques, or separation and similitude measures. Separation based strategy, for example, k-mean examination, various leveled bunching utilized a chose set of words showing up in various archive as highlights. Each record is exhibited by an element vector, and can be seen as a point in multidimensional space.

There are various issues with grouping in a multidimensional space utilizing conventional probabilistic base strategy or separation technique. First it isn't minor to characterize separation measure right now. Highlight vector must be scale to keep away from the slanting the outcome by various report lengths or conceivably by how a typical word is across numerous archives. Second, the quantity of various words in the reports can be exceptionally enormous. High dimensionality of the site page should be considered for bunching issue; unimportant element can upset the arrangement of the grouping calculation. Thusly, successful strategy for diminishing dimensionality of the web information should be proceeded as preprocessing step preceding grouping.

HTML labels, CSS and JavaScript should be expelled before grouping calculation starts. Other preprocessing steps, for example, expelling stop words and stemming should be acted so as to lessen the dimensionality space of the information. Each website page grouping framework need include choice which chooses the best element so as to diminish the dimensionality of information. A few methods have been as of late proposed to naturally produce Web wrappers, i.e., programs that concentrate information from HTML pages, and change them into an increasingly organized configuration, regularly in XML.

A few powerful element choice techniques are given, including two managed strategies, data gain (IG) and □2 Statistic (CHI), and unaided strategy, term quality (TS). Every one of these strategies appoint a score to every individual component and afterward select highlights which are more prominent than a pre-characterized limit.

### 2.6.1 Information Gain (IG)

Information gain of a term measures the number of bits of information obtained for category prediction by the presence or absence of the term in a document [12]. The information gain of a term t is defined as

$$\begin{aligned} \mathrm{IG}(t) = &-\sum \mathrm{P}(c_i)\log(\mathrm{P}(c_i)) + \mathrm{P}(t) \sum \mathrm{P}(c_i|t)\log(\mathrm{P}(c_i|t)) \\ &+ \mathrm{P}(\neg t)\sum \mathrm{P}(c_i|\neg t)\log(\mathrm{P}(c_i|\neg t)) \end{aligned} \tag{2.3}$$

Where $\mathrm{P}(c_i)$ is probability document in $c_i$.$\mathrm{P}(t)$ is probability t appears in document, $\mathrm{P}(c_i|t)$ is probability document in $c_i$given$t$ appears, $\mathrm{P}(c_i|\neg t)$ is document in $c_i$given$t$ does not appear.

#### 2.6.1.1 χ2 Statistic (CHI)

The $\chi^2$ statistic measures the association between the term and the category. $\chi^2$ statistic of a term in categories $\chi^2(\mathrm{t,c})$is defined to be

$$\chi^2(t,c) = \frac{N * \left(p(t,c) * p(\overline{t,c}) - p(t,\bar{c}) * p(\bar{t},c)\right)^2}{p(t) * p(\bar{t}) * p(c) * p(\bar{c})} \tag{2.4}$$

$$\chi^2(\mathrm{t}) = \underset{\mathrm{i}=1}{\overset{\mathrm{m}}{\mathrm{avg}}} \{\chi^2(\mathrm{t,c_i})\} \tag{2.5}$$

Where N be the number of documents in the dataset, c is the category, P(t) is probability t appears in document, m is the number of categories,P(c) is probability document in c andP(t,c) is probability document in c given t appears.

#### 2.6.1.2 Term Strength (TS)

Term strength is originally proposed and evaluated for vocabulary reduction in text retrieval as shown in Equation 2.6. It is computed based on the conditional probability that a term occurs in the second half of a pair of related documents given that it occurs in the first half:

$$TS(t) = p\left(t \in d_j | t \in d_i\right), d_i, d_j \in D \cap sim\left(d_i, d_j\right) > \beta \tag{2.6}$$

Where $\beta\ is$ the parameter to determine the related pairs, the similarity is can be calculated for each document pair, the time complexity of TS is quadratic to the number of documents. Because the class label information is not required, this method is also suitable for term reduction in text clustering.

## 2.7 Development Areas of Clustering

Grouping is utilized in practically every one of the fields.

- Clustering assists advertisers with improving their client base and work on the objective regions. It enables gathering to individuals (as indicated by various criteria's, for example, readiness, buying power and so forth.) in view of their comparability from numerous points of view identified with the item viable.
- Clustering helps in recognizable proof of gatherings of houses based on their worth, type and geological areas.
- Clustering is utilized to contemplate earth-shake. In light of the zones hit by a quake in an area, bunching can help examine the following plausible area where tremor can happen.

### 2.7.1 K-means Clustering

A pizza tie needs to open its conveyance focuses over a city. The potential difficulties are as the accompanying.

- They need to break down the zones from where the pizza is being requested much of the time.
- They need to comprehend concerning what number of pizza stores must be opened to cover conveyance in the region.
- They need to make sense of the areas for the pizza stores inside every one of these regions so as to keep the separation between the store and conveyance focuses least.

Settling these difficulties incorporates a ton of examination and science. It tends to be found out about how grouping can give an important and simple technique for sifting through such genuine difficulties.

### 2.7.2 K-means Clustering Method

If k is given, the K-means algorithm can be executed in the following steps:

- Partition of objects into k non-empty subsets
- Identifying the cluster centroids (mean point) of the current partition.
- Assigning each point to a specific cluster
- Compute the distances from each point and allot points to the cluster where the distance from the centroid is minimum.
- After re-allotting the points, find the centroid of the new cluster formed.

In Figure 2.3, there are five steps to describe detail processes of K-means. For k=2, firstly k objects are initialized with cluster centers. And then, each point is assigned to similar centers. In the next step, the cluster centers are identified using Euclidean distance method. And then, the points (based on minimum distance) are reassigned. After then, the new cluster centroids are identified and reassign the points according to the new cluster centroids.

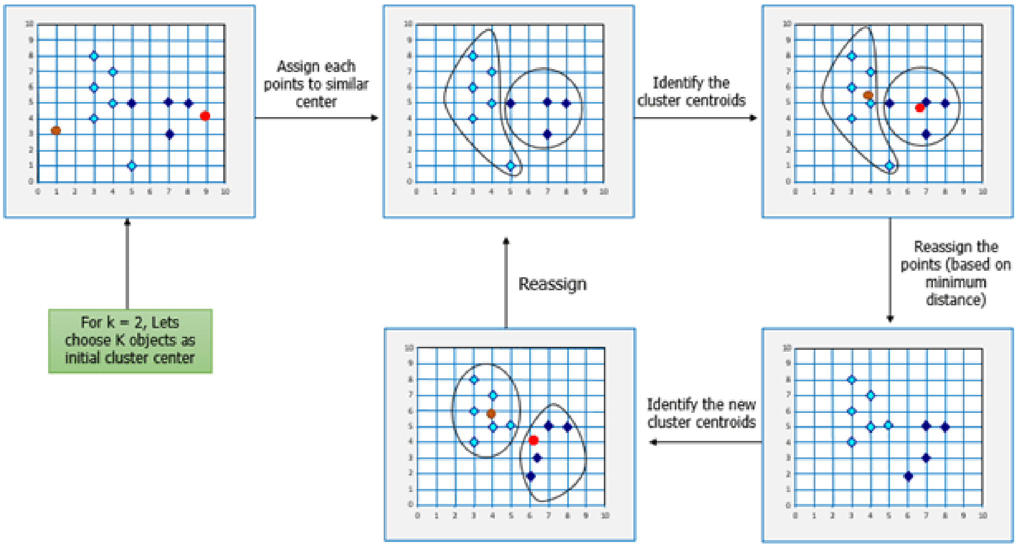

Assign each points to similar center
Identify the cluster centroids
Reassign the points (based on minimum distance)
Reassign
For k = 2, Lets choose K objects as initial cluster center
Identify the new cluster centroids

**Figure 2.3 Detail Processes of K-Means**

The pizza chain with focuses dependent on K-implies calculation is depicted. Essentially, for opening Hospital Care Wards: K-implies Clustering will amass these areas of most extreme inclined zones into groups and characterize a bunch place for each bunch, which will be where the Emergency Units will open. These Clusters focuses are the centroids of each bunch and are at any rate good ways from every one of the purposes of a specific group, consequently, the Emergency Units will be at least good ways from all the clumsy territories inside a group.

K-Means is one of the most significant calculations with regards to Machine learning Certification Training. Right now, K-Means bunching calculation will be comprehended with the assistance of models.

A Hospital Care affix needs to open a progression of Emergency-Care Center inside a locale. It tends to be accepted that the emergency clinic knows the area of all the most extreme clumsy regions in the district. They need to choose the quantity of the Emergency Units to be opened and the area of these Emergency Units, with the goal that all the clumsy zones is shrouded in the region of these Emergency Units. The test is to choose the area of these Emergency Units with the goal that the entire district is secured.

A bunch alludes to a little gathering of items. Bunching is gathering those articles into groups. So as to pick up bunching, it is imperative to comprehend the situations that lead to group various articles. It very well may be distinguished as the accompanying.

Bunching: Clustering is separating information focuses into homogeneous classes or groups: Points in a similar gathering are as comparable as could reasonably be expected and Points in various gathering are as divergent as would be prudent. At the point when an assortment of articles is given, items can be assembled into bunch dependent on comparability.

K-implies bunching is a sort of unaided realizing, which is utilized when you have unlabeled information (i.e., information without characterized classes or gatherings). The objective of this calculation is to discover bunches in the information, with the quantity of gatherings spoke to by the variable K. K-implies is a grouping calculation that attempts to parcel a lot of focuses into K sets (bunches) to such an extent that the focuses in each group will in general be close with one another. It is unaided on the grounds that the focuses have no outside characterization. K-implies grouping is one of the most straightforward and mainstream unaided AI calculations. At the end of the day, the K-implies calculation distinguishes k number of centroids, and afterward assigns each datum point to the closest bunch, while keeping the centroids as little as conceivable [8].

# CHAPTER 3
# WORD2VEC MODEL AND CLUSTERING

## 3.1. Word Embedding

In the simplistic terms, Word Embedding are the texts converted into numbers and there may be different numerical representations of the same text. Among them, the details of Word Embedding can be divided into various categories.

Many Machine Learning algorithms and almost all Deep Learning Architectures are incapable of processing *strings* or *plain text* in their raw form. They require numbers as inputs to perform any sort of job, such as classification, regression etc. And with the huge amount of data that present in the text format, it is imperative to extract knowledge out of it and build applications. Some real-world applications of text applications are – sentiment analysis of reviews by Amazon, document or news classification or clustering by Google.

A Word Embedding format generally tries to map a word using a dictionary to a vector. This sentencecanbebrokendown into finer details to have a clear view. For example, sentence=”Word Embedding are WordsConverted into numbers”. A *word* in this sentence may be “Embedding” or “numbers” etc.

A *dictionary* may be the list of all unique words in the sentence**.** Therefore, a dictionary may look like – [‘Word’,’Embedding’,’are’,’Converted’,’into’,’numbers’]

A ve*ctor* representation of a word may be a one-hot encoded vector where 1 stands for the position where the word exists and 0 everywhere else. The vector representation of “numbers” in this format according to the above dictionary is [0,0,0,0,0,1] and of converted is[0,0,0,1,0,0]. This is just a very simple method to represent a word in the vector form[3].

## 3.2. Different types of Word Embedding

The different types of word embedding can be broadly classified into two categories: Frequency based Embedding and Prediction based Embedding.

### 3.2.1. Frequency based Embedding

There are generally three types of vectors that we encounter under this category: count vector, TF-IDF vector and Co-Occurrence vector.

### 3.2.1.1 Count Vector

There is a corpus C of D documents {d1,d2…..dD} and N unique tokens extracted out of the corpus C. The N tokens form the dictionary and the size of the Count Vector matrix M is given by D x N. Each row in the matrix M contains the frequency of tokens in document D(i). For Example, D1: He is a lazy boy. She is also lazy.D2: Neeraj is a lazy person.The dictionary created may be a list of unique tokens (words) in the corpus ,[‘He’,’She’,’lazy’,’boy’,’Neeraj’,’person’].

D=2, N=6

The count matrix M of size 2 X 6 is represented as shown in Figure 3.1.

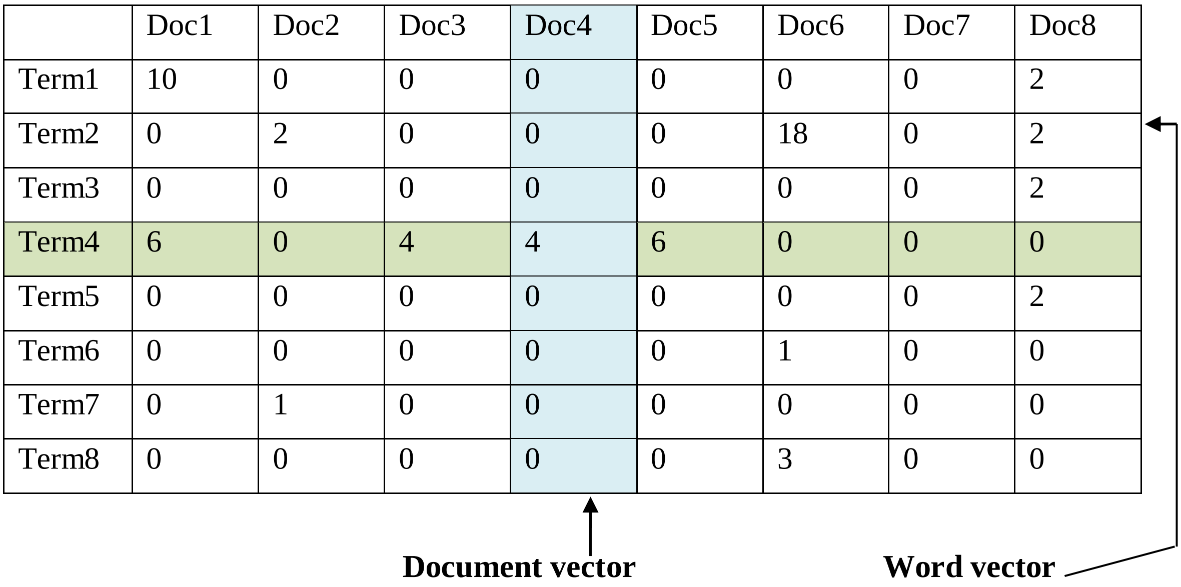
| | He | She | lazy | boy | Neeraj | Person |
|---|---|---|---|---|---|---|
| D1 | 1 | 1 | 2 | 1 | 0 | 0 |
| D2 | 0 | 0 | 1 | 0 | 1 | 1 |

**Figure3.1The Count Matrix M**

A column can also be understood as word vector for the corresponding word in the matrix M. For example, the word vector for ‘lazy’ in the above matrix is [2,1] and so on. Here the *rows* correspond to the *documents* in the corpus and the *columns* correspond to the *tokens* in the dictionary. The second row in the above matrix may be read as – D2 contains ‘lazy’: once, ‘Neeraj’: once and ‘person’ once.

There may be quite a few variations while preparing the above matrix M. The variations will be generally in the way dictionary is prepared because in real world applications in which there is a corpus which contains millions of documents. And with millions of documents, hundreds of millions of unique words can be extracted. Therefore, basically, the matrix that is prepared like above is a very sparse one and inefficient for any computation. Therefore, an alternative to using every unique word as a dictionary element would be to pick say top 10,000 words based on frequency and then prepare a dictionary.The way count is taken for each word. It can be taken that the frequency (number of times a word has appeared in the document) or the presenceto be the entry in the count matrix M. But generally, frequency method is preferred over the latter.

Figure 3.2is a representational image of the matrix M for easy understanding.



|  | Doc1 | Doc2 | Doc3 | Doc4 | Doc5 | Doc6 | Doc7 | Doc8 |
|---|---|---|---|---|---|---|---|---|
| Term1 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Term2 | 0 | 2 | 0 | 0 | 0 | 18 | 0 | 2 |
| Term3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Term4 | 6 | 0 | 4 | 4 | 6 | 0 | 0 | 0 |
| Term5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Term6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Term7 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Term8 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 |

Document vector
Word vector

**Figure3.2 Representational Image of the Matrix M**

## 3.2.1.2 TF-IDF Vectorization

This is another technique which depends on the recurrence strategy yet it is distinctive to the include vectorization as in it considers not simply the event of a word in a solitary archive yet in the whole corpus.

Regular words like 'is', 'the', 'an' and so forth will in general show up every now and again in contrast with the words which are imperative to an archive. For instance, a record A contains more events of "Messi" in contrast with different archives. In any case, basic words like "the" and so on are additionally going to be available in higher recurrence in pretty much every archive. Preferably, it is to down weight the normal words happening in practically all records and give more significance to words that show up in a subset of documents.TF-IDF works by punishing these basic words by allocating them lower loads while offering significance to words like Messi in a specific report. In this way, how precisely does TF-IDF work?

As appeared in test, it gives the check of terms (tokens/words) in two reports. There are two kinds of classes: term and tally. Terms are one of a kind words.

**Table 3.1 TF-IDF Document1**

| Term | Count |
|---|---|
| This | 1 |
| Is | 1 |
| About | 2 |
| Messi | 4 |

**Table 3.2 TF-IDF Document2**

| Term | Count |
|---|---|
| This | 1 |
| Is | 1 |
| About | 2 |
| Messi | 4 |

As shown in Table 3.1 and Table 3.2 , two documents are defined.

TF = (Number of times term t appears in a document)/ (Number of terms in the document).Therefore, TF (This, Document1) = 1/8 ,TF (This, Document2) =1/5. It denotes the contribution of the word to the document i.e. words relevant to the document should be frequent. For example, a document about Messi should contain the word ‘Messi’ in large number.

$$IDF = \log(N/n) \qquad (3.1)$$

Where, N is the number of documents ,

n is the number of documents a term t has appeared in.Therefore, IDF (This) = log (2/2) = 0.

Therefore, the reasoning behind IDF,if a word has appeared in the entire document, then probably that word is not relevant to a particular document. But if it has appeared in a subset of documents then probably the word is of some relevance to the documents as the following.Compute IDF for the word ‘Messi’.IDF (Messi) = log (2/1) = 0.301.

Compare the TF-IDF for a common word ‘This’ and a word ‘Messi’ which seems to be of relevance to Document 1.TF-IDF (This, Document1) = (1/8) * (0) = 0, TF-IDF (This, Document2) = (1/5) * (0) = 0, TF-IDF (Messi, Document1) = (4/8)*0.301 = 0.15

For Document1, TF-IDF method heavily penalizes the word ‘This’ but assigns greater weight to ‘Messi’. Therefore, this may be understood as ‘Messi’ is an important word for Document1 from the context of the entire corpus.

### 3.2.1.3 Co-OccurrenceMatrix with a Fixed Context Window

Similar words tend to occur together and will have similar context for example – Apple is a fruit. Mango is a fruit. Apple and mango tend to have a similar context i.e. fruit.

Before the details of how a co-occurrence matrix is constructed, there are two concepts that need to be clarified – Co-Occurrence and Context Window.Co-occurrence – For a given corpus, the co-occurrence of a pair of words say w1 and w2 is the number of times they have appeared together in a Context Window.

Context Window – Context window is specified by a number and the direction. Therefore, what does a context window of 2 (around) means?

An example,

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Quick | Brown | Fox | Jump | Over | The | Lazy | Dog |

The green words are a 2 (around) context window for the word 'Fox' and for calculating the co-occurrence only these words will be counted. Context window for the word is 'Over'.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Quick | Brown | Fox | Jump | Over | The | Lazy | Dog |

An example corpus to calculate a co-occurrence matrix.

Corpus = He is not lazy. He is intelligent. He is smart.

**Table 3.3 Co-occurrence matrix**

| | He | is | not | lazy | intelligent | smart |
|---|---|---|---|---|---|---|
| He | 0 | 4 | 2 | 1 | 2 | 1 |
| is | 4 | 0 | 1 | 2 | 2 | 1 |
| not | 2 | 1 | 0 | 1 | 0 | 0 |
| lazy | 1 | 2 | 1 | 0 | 0 | 0 |
| intelligent | 2 | 2 | 0 | 0 | 0 | 0 |
| smart | 1 | 1 | 0 | 0 | 0 | 0 |

As shown in Table 3.3, this co-occurrence matrix shows how to appear words. The box of the first rowis the number of times 'He' and 'is' have appeared in the context window 2 and it can be seen that the count turns out to be 4. As shown in Table 3.4, the count can be visualized according to the number of times the word appears.

**Table 3.4 Co-occurrence matrix2**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| He | is | not | lazy | He | is | intelligent | He | is | smart |
| He | is | not | lazy | He | is | intelligent | He | is | smart |
| He | is | not | lazy | He | is | intelligent | He | is | smart |
| He | is | not | lazy | He | is | intelligent | He | is | smart |

The word 'lazy' has never appeared with 'intelligent' in the context window and therefore has been assigned 0 in the blue box.

**Variations of Co-occurrence Matrix:** there are V unique words in the corpus. Vocabulary size = V. The columns of the Co-occurrence matrix form the *context word*s. The different variations of Co-Occurrence Matrix are:

- A co-occurrence matrix of size V X V. Now, for even a decent corpus V gets very large and difficult to handle. So generally, this architecture is never preferred in practice.
- A co-occurrence matrix of size V X N where N is a subset of V and can be obtained by removing irrelevant words like stop words etc. for example. This is still very large and presents computational difficulties.

This co-occurrence matrix is not the word vector representation that is generally used. Instead, this Co-occurrence matrix is decomposed using techniques like PCA, SVD etc. into factors and combination of these factors forms the word vector representation.

For example, PCA can be performed on the above matrix of size VXV. The V principal components are obtained and k components out of these V components. Therefore, the new matrix will be of the form V X k.

And, a single word, instead of being represented in V dimensions will be represented in k dimensions while still capturing almost the same semantic meaning. k is generally of the order of hundreds.

## Advantages of Co-occurrence Matrix

Co-occurrence matrix preserves the semantic relationship between words. For example, man and woman tend to be closer than man and apple.It uses SVD at its core, which produces more accurate word vector representations than existing methods. Moreover, it uses factorization which is a well-defined problem and can be efficiently solved. And then, it has to be computed once and can be used anytime once computed. In this sense, it is faster in comparison to others.

## Disadvantages of Co-Occurrence Matrix

It requires huge memory to store the co-occurrence matrix. The length of the vocabulary is large, resulting in a large vector length of the word. The co-occurrence matrix is also a sparse matrix (can be used *SVD,PCA* Such algorithms perform dimensionality reduction, but the amount of calculation is large).

### 3.2.2 Prediction Based Vector

Pre-requisite is that a working knowledge of how a neural network works and the mechanisms by which weights in an NN are updated. As a beginner for Neural Network, the following article will help to gain a very good understanding of how NN works.

There are deterministic methods to determine word vectors. But these methods proved to be limited in their word representations until Mikolov etc. el introduced word2vec to the NLP community. These methods were prediction based in the sense that they provided probabilities to the words and proved to be state of the art for tasks like word analogies and word similarities. They were also able to achieve tasks like King -man +woman = Queen, which was considered a result almost magical. Therefore, word2vec model is used to generate semantic vectors.

Word2vec is not a single algorithm but a combination of two techniques – CBOW (Continuous bag of words) and Skip-gram model. Both of these are shallow neural networks which map word(s) to the target variable which is also a word(s). Both of these techniques learn weights which act as word vector representations. Let us discuss both these methods separately and gain intuition into their working.

Given a text corpus, the word2vec tool learns a vector for every word in the vocabulary using the Continuous Bag-of-Words or the Skip-Gram neural network architectures [4].

The user should specify the following:

- Desired vector dimensionality
- The size of the context window for either the Skip-Gram or the Continuous Bag-of-Words model
- Training algorithm: hierarchical softmax and / or negative sampling
- Threshold for downsampling the frequent words
- Number of threads to use
- The format of the output word vector file (text or binary)

The script demo-word.sh downloads a small text corpus from the web, and trains a small word vector model. After the training is finished, the vector values are evaluated.

### 3.2.2.1 CBOW (Continuous Bag of words)

The way CBOW work is that it tends to predict the probability of a word given a context. A context may be a single word or a group of words. But for simplicity, single context word is taken and tries to predict a single target word.

For Example, a corpus C "Hey, this is sample corpus using only one context word." and defined a context window of 1. This corpus may be converted into a training set for a CBOW model. The input is shown in Table 3.5,the matrix on the right contains the one-hot encoded from of the input on the left.

**Table 3.5 One-hot vector**

| Input | Output | Datapoint | Hey | This | is | sample | corpus | using | only | one | context | word |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hey | this | Datapoint1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| this | hey | Datapoint2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| is | this | Datapoint3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| is | sample | Datapoint4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| sample | is | Datapoint5 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| sample | corpus | Datapoint6 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| corpus | sample | Datapoint7 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| corpus | using | Datapoint8 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| using | corpus | Datapoint9 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| using | only | Datapoint10 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| only | using | Datapoint11 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| only | one | Datapoint12 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| one | only | Datapoint13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| one | context | Datapoint14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| context | one | Datapoint15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| context | word | Datapoint16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| word | context | Datapoint17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

The target for a single data point is Data point 4 [0010000000] as shown in Table 3.6. If firstly 1 appears as one-hot vector twice in the same column, the second row is the single datapoint.

**Table 3.6 Single Data Point**

| Hey | this | is | sample | corpus | using | only | one | context | word |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

This matrix is sent into a shallow neural network with three layers: an input layer, a hidden layer and an output layer. The output layer is a softmax layer which is used to sum the probabilities obtained in the output layer to 1. The forward propagation will work to calculate the hidden layer activation.

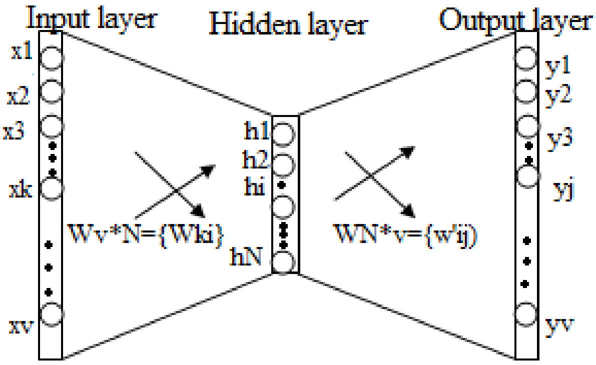


**Figure3.3 Diagrammatic Representation of CBOW Model**

The work flow of CBOW model as shown in Figure 3.3 is as follows:

- The input layer and the target, both are one- hot encoded of size [1 X V]. Here V=10 in the above example.
- There are two sets of weights. One is between the input and the hidden layer and second between input and output layer. Input-Hidden layer matrix size =[V X N], hidden-Output layer matrix size= [N X V]: Where N is the number of dimensions to represent each word. It is arbitrary and a hyper-parameter for a Neural Network. Also, N is the number of neurons in the hidden layer. Here, N=4.
- There is a no activation function between any layers. (More specifically, I am referring to linear activation)
- The input is multiplied by the input-hidden weights and called hidden activation. It is simply the corresponding row in the input-hidden matrix copied.
- The hidden input gets multiplied by hidden- output weights and output is calculated.
- Error between output and target is calculated and propagated back to re-adjust the weights.
- The weight between the hidden layer and the output layer is taken as the word vector representation of the word.

## Advantages of CBOW

Being probabilistic is nature, it is supposed to perform superior to deterministic methods (generally).It is low on memory. It does not need to have huge RAM requirements like that of co-occurrence matrix where it needs to store three huge matrices.

## Disadvantages of CBOW

CBOW takes the average of the context of a word (as seen above in calculation of hidden activation). For example, Apple can be both a fruit and a company but CBOW takes an average of both the contexts and places it in between a cluster for fruits and companies. Training a CBOW from scratch can take forever if not properly optimized.

### 3.2.2.2 Skip – Gram model

Skip-gram follows the same topology as of CBOW. It just flips CBOW's architecture on its head. The aim of skip-gram is to predict the context given a word. The same corpus that is similar to CBOW model on. C= "Hey, this is sample corpus using only one context word." The training data can be constructed as shown in Table 3.7. Input column is tokenized words of a sentence. The output context window is 2. Therefore, the output context words are defined next two placements from left to right depending on the target word.

**Table 3.7 Training Data Table**

| Input | Output(Context1) | Output(Context2) |
|---|---|---|
| Hey | this | <padding> |
| this | Hey | is |
| is | this | sample |
| sample | Is | corpus |
| corpus | sample | corpus |
| using | corpus | only |
| only | using | one |

| | | |
|---|---|---|
| one | only | context |
| context | one | word |
| word | context | <padding> |

The input vector for skip-gram is going to be similar to a 1-context CBOW model. Also, the calculations up to hidden layer activations are going to be the same. The difference will be in the target variable. Since it can be defined a context window of 1 on both the sides, there will be “two” one hot encoded target variables and “two” corresponding outputs as can be seen by the blue section in the image.

Two separate errors are calculated with respect to the two target variables and the two error vectors obtained are added element-wise to obtain a final error vector which is propagated back to update the weights.

The weights between the input and the hidden layer are taken as the word vector representation after training. The loss function or the objective is of the same type as of the CBOW model.The skip-gram architecture is shown in Figure 3.4 [7].

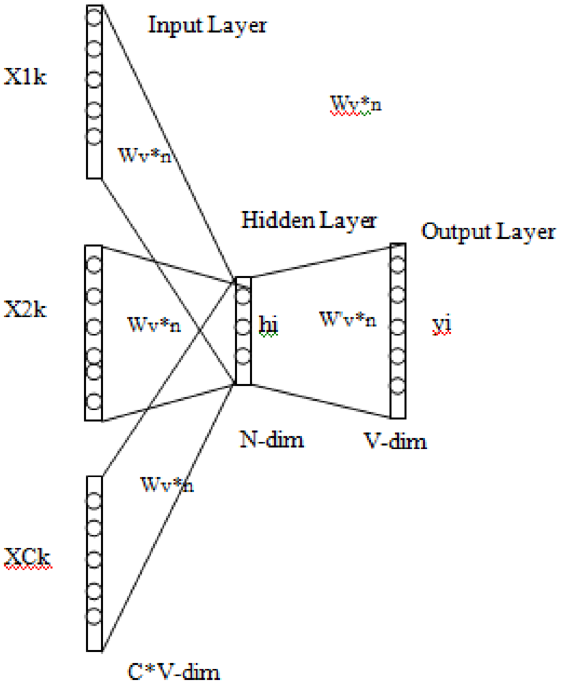


**Figure 3.4 Skip-gram Architecture**

## Advantages of Skip-Gram Model

- Skip-gram model captures two semantics for a single word. For example, it will have two vector representations of Apple. One for the company and other for the fruit.
- Skip-gram with negative sub-sampling performs every other method generally.

## 3.3. Word2Vec and K-means Clustering

The case studies for word2vec calculation and k-means clustering calculation are described step by step. Word2vec model calculates semantic vector outputs that are used k-means clustering as the input to cluster the similar groups of job seekers data.

### Word2Vec Calculation

**Table 3.8 Job Seekers' Job Titles Table**

| JSID | Job Title |
|---:|---|
| 1 | Automation Test Engineer |
| 2 | Information Security-Engineer |
| 3 | Network Engineer |
| 4 | Sr. Web Application Developer |
| 5 | Front End Developer |
| 6 | OpenStack Engineer |
| 7 | Data Security Administrator |
| 8 | Software Engineer Manager |
| 9 | Sales Engineer |
| 10 | Project Manager |

As shown in Table 3.8, there are sample job titles of job seekers. The first step of word2vec is to convert unique words into one-hot vectors. Input and output of skip-gram model are one-hot vectors as shown in Table 3.9.

Two steps of word2vec model are forward propagation and backward propagation. Forward propagation is simple neural network architecture with one hidden layer. Backward propagation is updating input weight matrix and out weight matrix of vector values according to the context words and target words. After backward propagation step, semantic vector values of words are evaluated as shown Table 3.10.

**Table 3.9 Converting Job Titles into One-Hot Vectors Table**

| WordID | Word | One-Hot Vector |
|---|---|---|
| 1 | Automation | 10000 00000 00000 0000 |
| 2 | Test | 01000 00000 00000 0000 |
| 3 | Engineer | 00100 00000 00000 0000 |
| 4 | Information | 00010 00000 00000 0000 |
| 5 | Security | 00001 00000 00000 0000 |
| 6 | Network | 00000 10000 00000 0000 |
| 7 | Sr | 00000 01000 00000 0000 |
| 8 | Web | 00000 00100 00000 0000 |
| 9 | Application | 00000 00010 00000 0000 |
| 10 | Developer | 00000 00001 00000 0000 |
| 11 | Front | 00000 00000 10000 0000 |
| 12 | End | 00000 00000 01000 0000 |
| 13 | OpenStack | 00000 00000 00100 0000 |
| 14 | Data | 00000 00000 00010 0000 |
| 15 | Administrator | 00000 00000 00001 0000 |
| 16 | Software | 00000 00000 00000 1000 |
| 17 | Manager | 00000 00000 00000 0100 |
| 18 | Sales | 00000 00000 00000 0010 |
| 19 | Project | 00000 00000 00000 0001 |

**Table 3.10 Converting Semantic Word2Vec**

| Change Vectors using Word2Vec | | | |
|---|---|---|---|
| JSID | Degree | Job Title | Skill-Set |
| 1 | 2.598455 | 1.200178 | 2.3939535 |
| 2 | 3.412667 | 1.312127 | 3.969146 |
| 3 | 3.256848 | 0.341778 | 3.089183 |
| 4 | 2.598455 | 2.310381 | 3.946951 |
| 5 | 2.754274 | 1.554208 | 3.31721 |
| 6 | 2.598455 | 0.803198 | 0.581001 |
| 7 | 2.754274 | 1.413141 | 2.538502 |
| 8 | 2.754274 | 0.717775 | 1.027695 |
| 9 | 3.412667 | 0.498919 | 1.359059 |
| 10 | 2.598455 | 0.182651 | 1.372904 |

## K-Means Clustering Calculation

In this case study, K-means clustering calculation is described with two tables. The input dataset is used with 10 records of job seekers and the number of cluster is 2. The nearest distance between each job seeker and the recruiter is calculated and the cluster centroids are updated step by step.

Firstly two records of job seekers are randomly initialized as shown in Table 3.11.

As shown in Table 3.12, first iteration of K-means clustering is described. Firstly, number of clusters is defined and two records are randomly initialized. And then, cluster centroids are calculated. Then, the difference of each cluster and each item is calculated. And then, cluster centroids are updated. Until the cluster centroids are constant, these steps are repeated.K-means produce tighter clusters than hierarchicalclustering, especially if the clusters are globular.

**Table 3.11 Initialized Two Clusters of Job Seekers Data**

| K | JSID | Degree | Job Title | Skill-Set |
|---|---|---|---|---|
| K1 | 4 | 2.598455 | 2.310381 | 3.946951 |
| K2 | 8 | 2.754274 | 0.717775 | 1.027695 |

**Table 3.12 First Iteration of K-means Clustering**

| | | | | |
|---|---|---|---|---|
| P(jsid1,k1) | | | | Sum |
| 1 | 2.598455 | 1.200178 | 2.3939535 | |
| K1 | 2.598455 | 2.310381 | 3.946951 | |
| | 0 | 1.110203 | 1.5529975 | 2.6632005 |
| P(jsid1,k2) | | | | Sum |
| 1 | 2.598455 | 1.200178 | 2.3939535 | |
| K2 | 2.754274 | 0.717775 | 1.027695 | |
| | 0.155819 | 0.482403 | 1.3662585 | 2.0044805 |
| | | | | |
| P(jsid2,k1) | | | | Sum |
| 2 | 3.412667 | 1.312127 | 3.969146 | |
| K1 | 2.598455 | 2.310381 | 3.946951 | |
| | 0.814212 | 0.998254 | 0.022195 | 1.834661 |
| P(jsid2,k2) | | | | Sum |
| 2 | 3.412667 | 1.312127 | 3.969146 | |
| K2 | 2.754274 | 0.717775 | 1.027695 | |
| | 0.658393 | 0.594352 | 2.941451 | 4.194196 |
| | | | | |
| P(jsid3,k1) | | | | Sum |
| 3 | 3.256848 | 0.341778 | 3.089183 | |
| K1 | 2.598455 | 2.310381 | 3.946951 | |
| | 0.658393 | 1.968603 | 0.857768 | 0 |
| P(jsid3,k2) | | | | Sum |
| 3 | 3.256848 | 0.341778 | 3.089183 | |
| K2 | 2.754274 | 0.717775 | 1.027695 | |

| | | | | |
|---|---|---|---|---|
| | 0.502574 | 0.375997 | 2.061488 | 1.934911 |

| | | | | |
|---|---|---|---|---|
| P(jsid4,k1) | | | | Sum |
| 4 | 2.598455 | 2.310381 | 3.946951 | |
| K1 | 2.598455 | 2.310381 | 3.946951 | |
| | 0 | 0 | 0 | 0 |
| P(jsid4,k2) | | | | Sum |
| 4 | 2.598455 | 2.310381 | 3.946951 | |
| K2 | 2.754274 | 0.717775 | 1.027695 | |
| | 0.155819 | 1.592606 | 2.919256 | 4.667681 |
| | | | | |
| P(jsid5,k1) | | | | Sum |
| 5 | 2.754274 | 1.554208 | 3.31721 | |
| K1 | 2.598455 | 2.310381 | 3.946951 | |
| | 0.155819 | 0.756173 | 0.629741 | 1.541733 |
| P(jsid5,k2) | | | | Sum |
| 5 | 2.754274 | 1.554208 | 3.31721 | |
| K2 | 2.754274 | 0.717775 | 1.027695 | |
| | 0 | 0.836433 | 2.289515 | 3.125948 |
| | | | | |
| P(jsid6,k1) | | | | Sum |
| 6 | 2.598455 | 0.803198 | 0.581001 | |
| K1 | 2.598455 | 2.310381 | 3.946951 | |
| | 0 | 1.507183 | 3.36595 | 4.873133 |
| P(jsid6,k2) | | | | Sum |
| 6 | 2.598455 | 0.803198 | 0.581001 | |
| K2 | 2.754274 | 0.717775 | 1.027695 | |
| | 0.155819 | 0.085423 | 0.446694 | 0.687936 |
| | | | | |
| P(jsid7,k1) | | | | Sum |
| 7 | 2.754274 | 1.413141 | 2.538502 | |
| K1 | 2.598455 | 2.310381 | 3.946951 | |
| | 0.155819 | 0.89724 | 1.408449 | 2.461508 |
| P(jsid7,k2) | | | | Sum |
| 7 | 2.754274 | 1.413141 | 2.538502 | |
| K2 | 2.754274 | 0.717775 | 1.027695 | |
| | 0 | 0.695366 | 1.510807 | 2.206173 |
| | | | | |
| P(jsid8,k1) | | | | Sum |
| 8 | 2.754274 | 0.717775 | 1.027695 | |
| K1 | 2.598455 | 2.310381 | 3.946951 | |
| | 0.155819 | 1.592606 | 2.919256 | 4.667681 |
| P(jsid8,k2) | | | | Sum |
| 8 | 2.754274 | 0.717775 | 1.027695 | |
| K2 | 2.754274 | 0.717775 | 1.027695 | |

| | | | | |
|---|---|---|---|---|
| | 0 | 0 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| P(jsid9,k1) | | | | Sum |
| 9 | 3.412667 | 0.498919 | 1.359059 | |
| K1 | 2.598455 | 2.310381 | 3.946951 | |
| | 0.814212 | 1.811462 | 2.587892 | 5.213566 |
| P(jsid9,k2) | | | | Sum |
| 9 | 3.412667 | 0.498919 | 1.359059 | |
| K2 | 2.754274 | 0.717775 | 1.027695 | |
| | 0.658393 | 0.218856 | 0.331364 | 1.208613 |
| | | | | |
| P(jsid10,k1) | | | | Sum |
| 10 | 2.598455 | 0.182651 | 1.372904 | |
| K1 | 2.598455 | 2.310381 | 3.946951 | |
| | 0 | 2.12773 | 2.574047 | 4.701777 |
| P(jsid9,k2) | | | | Sum |
| 10 | 2.598455 | 0.182651 | 1.372904 | |
| K2 | 2.754274 | 0.717775 | 1.027695 | |
| | 0.155819 | 0.535124 | 0.345209 | 1.036152 |

**Table 3.13Resulted Two Clusters of Job Seekers Data**

| JSID | Degree | Job Title | Skill-Set | |
|---|---|---|---|---|
| 2 | 3.412667 | 1.312127 | 3.969146 | |
| 4 | 2.598455 | 2.310381 | 3.946951 | C1 |
| 5 | 2.754274 | 1.554208 | 3.31721 | |
| 7 | 2.754274 | 1.413141 | 2.538502 | |
| 1 | 2.598455 | 1.200178 | 2.3939535 | |
| 3 | 3.256848 | 0.341778 | 3.089183 | |
| 6 | 2.598455 | 0.803198 | 0.581001 | |
| 8 | 2.754274 | 0.717775 | 1.027695 | C2 |
| 9 | 3.412667 | 0.498919 | 1.359059 | |
| 10 | 2.598455 | 0.182651 | 1.372904 | |

Using word2vec model and k-means clustering, final result is evaluated, C1 (cluster 1) and C2 (cluster 2)as shown in Table 3.13. Records in each cluster are nearly similar vectors to each other. And then, in the matching step, total score of each job seeker and recruiter's job profile are calculated. After then, the nearest total score between each cluster and job profile is calculated. According to the nearest score, final cluster is evaluated.

# CHAPTER 4

# DESIGN AND IMPLEMENTATION OFSYSTEM

## 4.1 The Overall System Architecture

Inefficiencies in the recruitment such as friction in matching job seekers to job posts and the existence of skill gaps in various job vacancies of the economy are considered to be major problems facing economies today. The central problem is that recruiters cannot get the most relevant job seekers that fit with job vacancies' requirements in a short time. This system implements a cluster-based job matching system that can match job vacancies with the most relevant job seekers and job seekers can also get many opportunities that are related with job posts as shown in Figure 4.1.

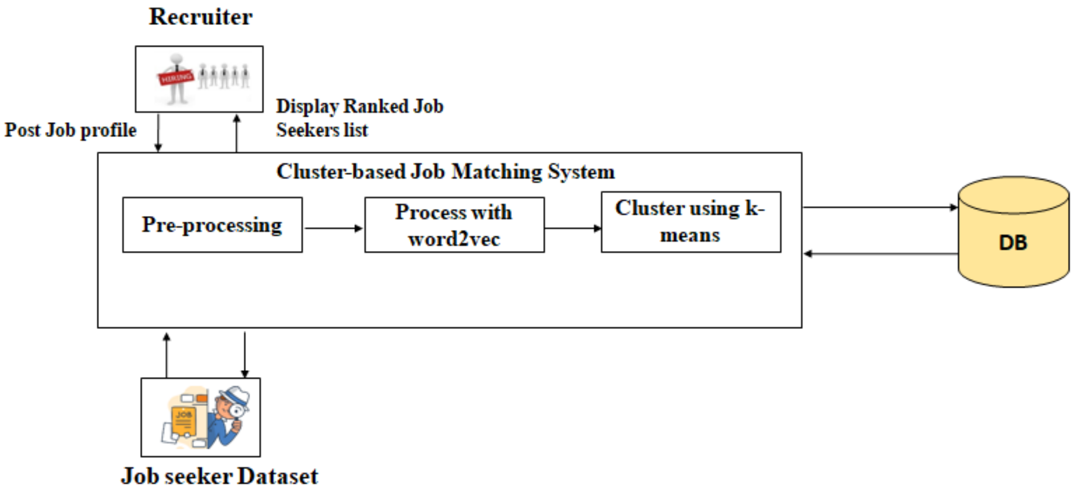


**Figure 4.1The Overall System Architecture**

The comprehensive quantitative assessment using a dataset consisting of job seekers' profiles and recruiter job posts suggests that skill recommendations made by k-means clusteringand then they are highly correlated with skills required in future jobs. This indicates that either members of the work force do not have skills demanded by jobs or do not have enough information about which are the best skills to signal for competing in the labor market.

## 4.2 The Design of The System

In this system, there are three users: Job Seeker, Job Recruiter, and Administrator. Firstly, job seekers' data are loaded and saves job seeker data into database. And then, the required attributes such as job title, degree, and skill set are retrieved. Then, tokenizing and removing stop-word processes are done as a processing step. After the preprocessing step,the words of job title, degree and skill set are changed into vectorsusing word2vec model. After these semantic words have been trained with word2vec, these are clustered using k-means clustering as shown in Figure 4.2.

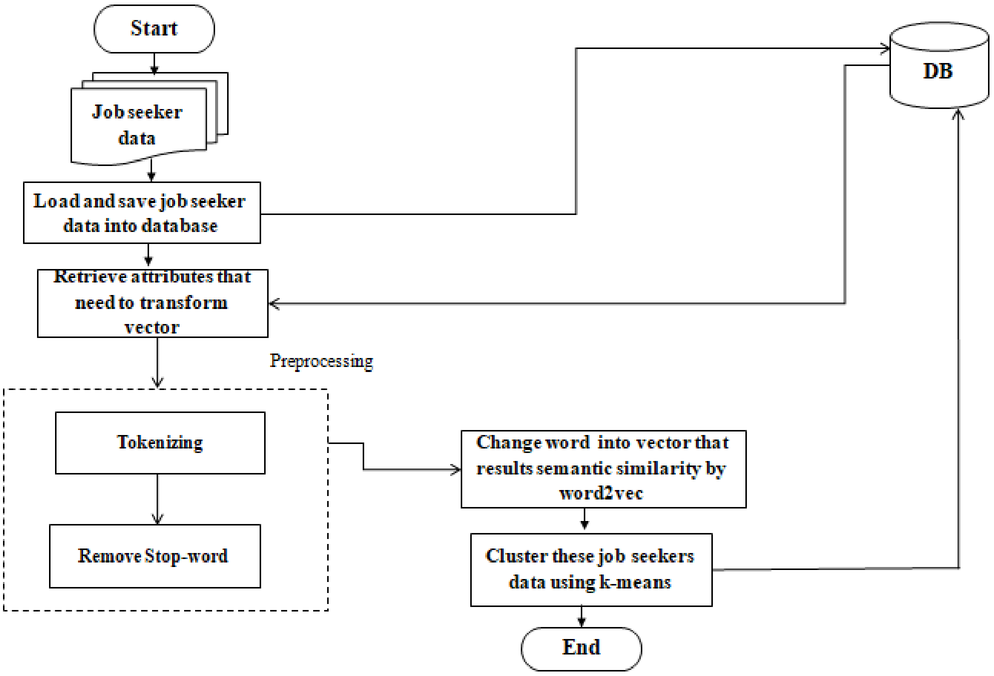

Start
Job seeker data
Load and save job seeker data into database
Retrieve attributes that need to transform vector
Preprocessing
Tokenizing
Remove Stop-word
Change word into vector that results semantic similarity by word2vec
Cluster these job seekers data using k-means
End
DB

**Figure4.2 Flow Diagram for Job Seeker**

As shown in Figure 4.3, once the job recruiter's job profile has been posted, the system saves into database as a new job profile. And then, administrator retrieves attributes that need to transform vectors such as job title, degree and skill set of recruiter's job vacancy profile. And then, the preprocessing steps such as tokenizing and removing stop-words of job title, degree and skill set. Those preprocessed unique words of job recruiter's data are converted into semantic vectors using word2vec model.

And then, the resulted semantic vectorsare clustered using k-means clustering method to evaluate the most relevant job seekers that meet with recruiter posts job profile. Mainly, administrator uses an effective job matching process in which once recruiter posts a new job profile, administrator not only converts attributes of that job profile (job title, degree, skill set) and but also converts attributes of all job seekers data that have been saved in database. And then, clusters are evaluated as number of clusters that the administrator submits. The nearest score is calculated between total scores of each job seekers and recruiter's job profile. According to the nearest score, the most relevant cluster of job seekers is evaluated.

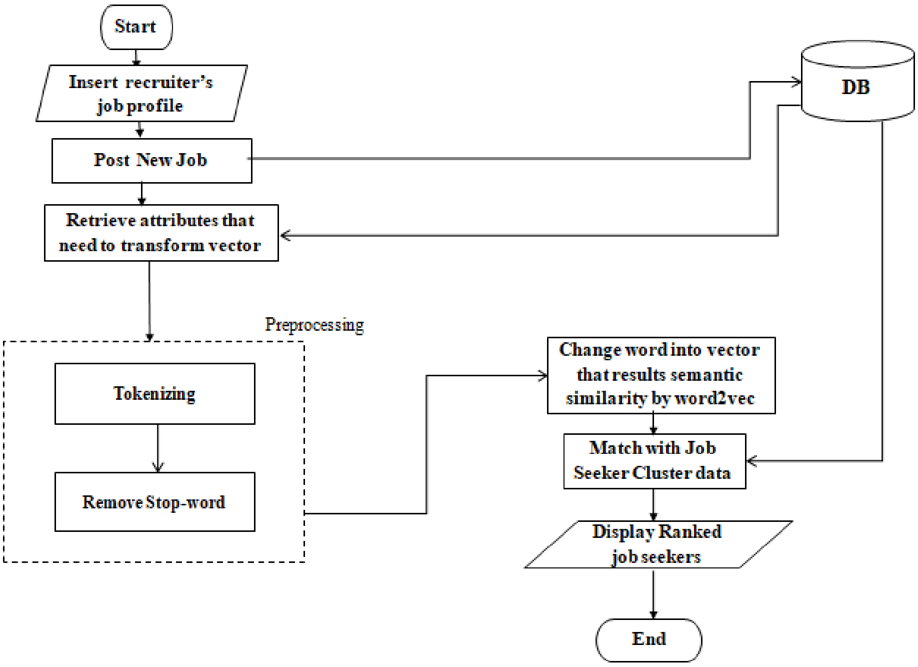

Start
Insert recruiter's job profile
Post New Job
Retrieve attributes that need to transform vector
Preprocessing
Tokenizing
Remove Stop-word
DB
Change word into vector that results semantic similarity by word2vec
Match with Job Seeker Cluster data
Display Ranked job seekers
End

**Figure4.3 Flow Diagram for Job Recruiter**

The database design of the system is described as shown in Figure 4.4. User table is used to login for recruiters. JS (Job Seeker) table is linked to the nearest score table and the sum of score table. And then, JS (job seeker) table is linked to TokenizedWordTable, RemoveWordCleanedTable and Word2Vec table. Moreover, JP (job post) table is linked to TokenizedWordTableJP, RemoveWordCleanedTableJP and Word2VecJP.4

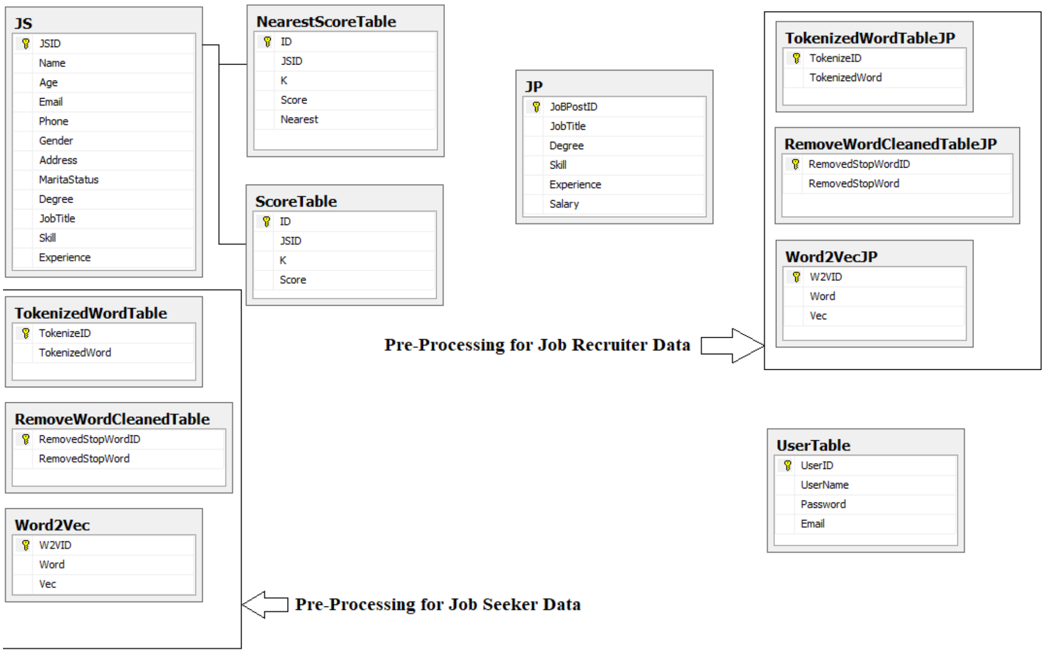


Figure 4.4 Database Design of the System

## 4.3Detail Steps of Proposed System

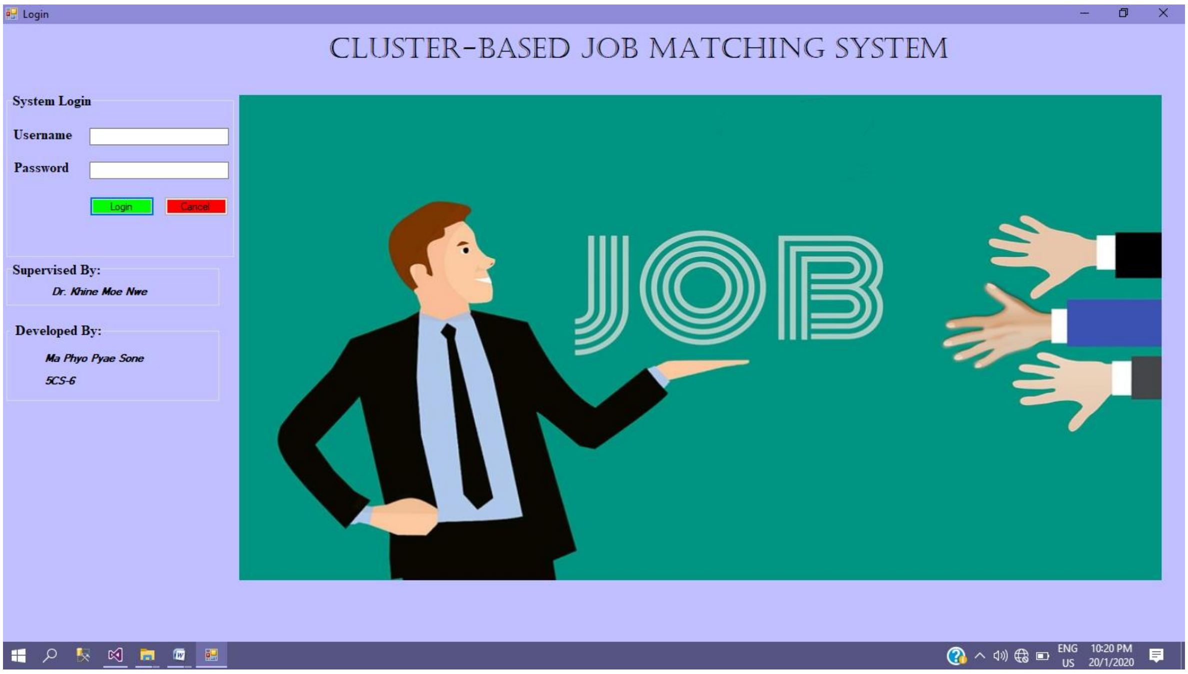


Figure4.5The Login Page

This implemented job matching system is aimed to serve as a third-party software service. Therefore, the job seekers can post their personal data and job-related data items in the system's job seekers' dataset. The job recruiters can also post their job post to hire the suitable employee by the aid of this system. Both of the job recruiters and job seekers must pass the system authentication as shown in Figure 4.5and then only the authenticated user can do their related processes in this job matching system.

## 4.4 Loading New Job Seeker Dataset

After user logins successfully, home page can be viewed as five main menus: “LoadJobSeekerData” menu which is used to load new job seekers’ dataset form external resources as shown in Figure 4.6, “Jobseekers” menu is to load the existing job seekers’ dataset from the system database, “Job Recruiter” menu is to load or submit new job post by the job recruiter, “Calculate” menu is main process of this system in which the job matching process will make by k-means clustering algorithm and the last menu is “Exit” which is the termination of this program from execution.

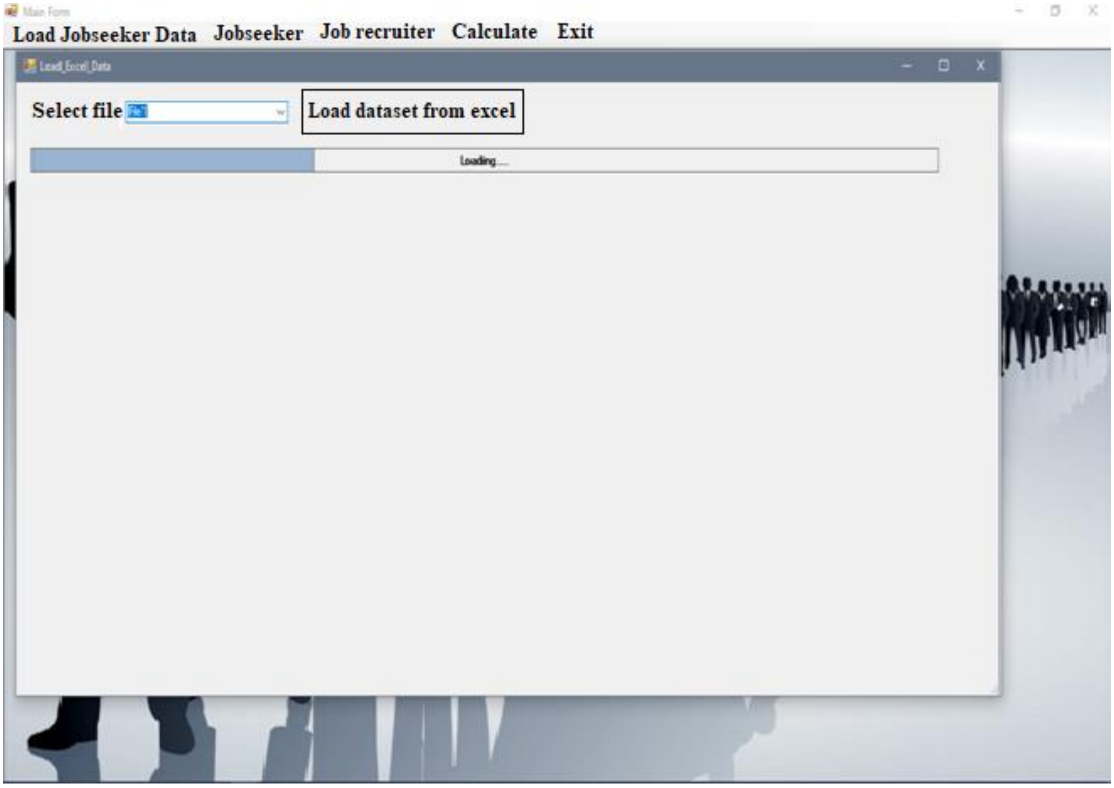


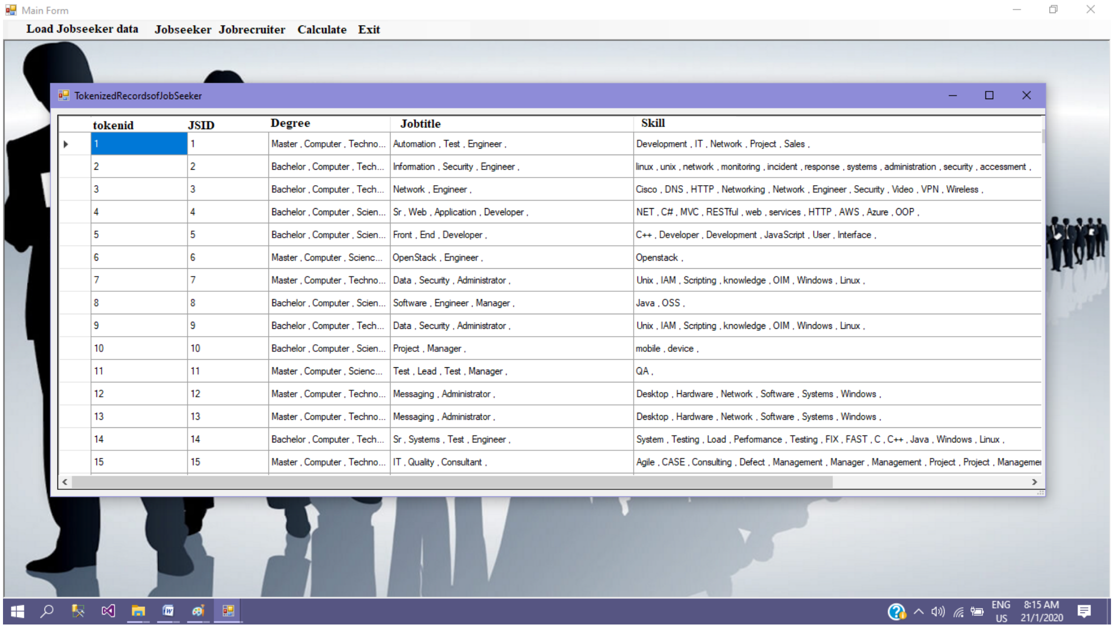
**Figure 4.6 Loading New Job Seeker Data**

## 4.5Pre-processing Steps of Job Matching System

After loading job seekers’ data, the loaded data must be processed the tokenizing and removing stop words as reprocessing phase.

### Tokenization

Tokenization is the process of dividing text into a set of meaningful pieces. These pieces are called tokens. For example, it can be divided a chunk of text into words, or it can be divided into sentences. Depending on the task at hand, it can be definedthat own conditions to divide the input text into meaningful tokens. The Job seeker data tokenization example is as shown in Figure 4.7.

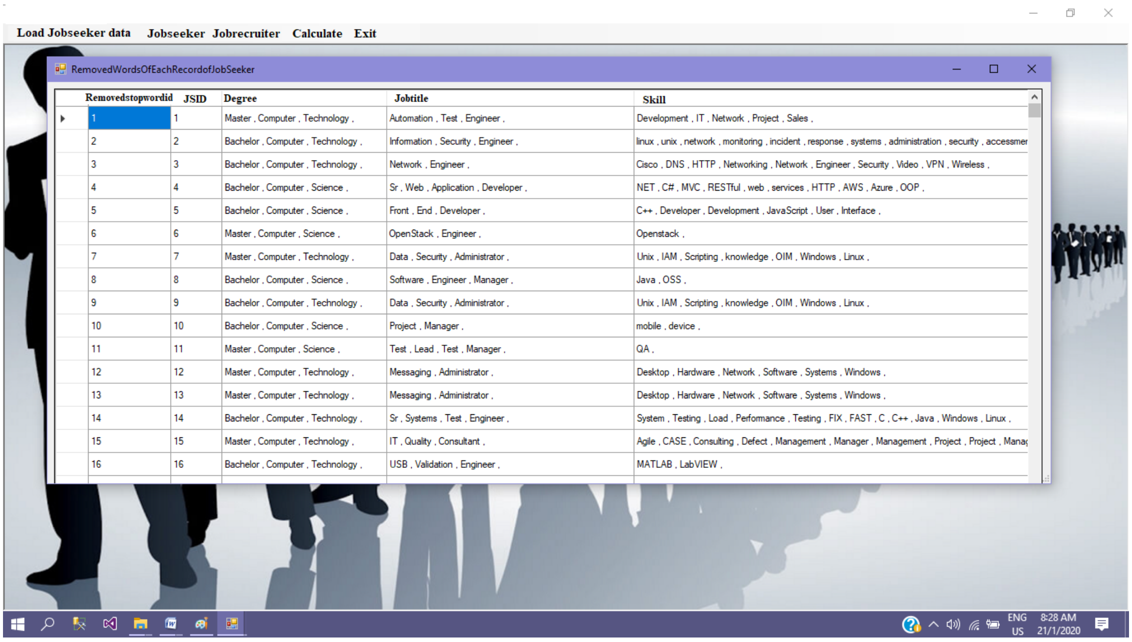


| | tokenid | JSID | Degree | Jobtitle | Skill |
|---|---|---|---|---|---|
| ▸ | 1 | 1 | Master , Computer , Techno... | Automation , Test , Engineer , | Development , IT , Network , Project , Sales , |
| | 2 | 2 | Bachelor , Computer , Tech... | Information , Security , Engineer , | linux , unix , network , monitoring , incident , response , systems , administration , security , accessment , |
| | 3 | 3 | Bachelor , Computer , Tech... | Network , Engineer , | Cisco , DNS , HTTP , Networking , Network , Engineer , Security , Video , VPN , Wireless , |
| | 4 | 4 | Bachelor , Computer , Scien... | Sr , Web , Application , Developer , | NET , C# , MVC , RESTful , web , services , HTTP , AWS , Azure , OOP , |
| | 5 | 5 | Bachelor , Computer , Scien... | Front , End , Developer , | C++ , Developer , Development , JavaScript , User , Interface , |
| | 6 | 6 | Master , Computer , Scienc... | OpenStack , Engineer , | Openstack , |
| | 7 | 7 | Master , Computer , Techno... | Data , Security , Administrator , | Unix , IAM , Scripting , knowledge , OIM , Windows , Linux , |
| | 8 | 8 | Bachelor , Computer , Scien... | Software , Engineer , Manager , | Java , OSS , |
| | 9 | 9 | Bachelor , Computer , Tech... | Data , Security , Administrator , | Unix , IAM , Scripting , knowledge , OIM , Windows , Linux , |
| | 10 | 10 | Bachelor , Computer , Scien... | Project , Manager , | mobile , device , |
| | 11 | 11 | Master , Computer , Scienc... | Test , Lead , Test , Manager , | QA , |
| | 12 | 12 | Master , Computer , Techno... | Messaging , Administrator , | Desktop , Hardware , Network , Software , Systems , Windows , |
| | 13 | 13 | Master , Computer , Techno... | Messaging , Administrator , | Desktop , Hardware , Network , Software , Systems , Windows , |
| | 14 | 14 | Bachelor , Computer , Tech... | Sr , Systems , Test , Engineer , | System , Testing , Load , Performance , Testing , FIX , FAST , C , C++ , Java , Windows , Linux , |
| | 15 | 15 | Master , Computer , Techno... | IT , Quality , Consultant , | Agile , CASE , Consulting , Defect , Management , Manager , Management , Project , Project , Manageme |

**Figure4.7 Tokenizing Job Seeker Data**

### RemovingStopwords

Words that carry no particular meaning such as “a”, “an”, “the” and some other common words should be eliminated. List of stop-words are maintained in the system database. The system removes the stopwords after tokenizing. Figure 4.8 shows the stopwords removed data list. To convert word into vector using word2vec model, pre-processing step is firstly used. After removing stopwords, the system converts unique words into semantic vector values.



| Removedstopwordid | JSID | Degree | Jobtitle | Skill |
|---|---|---|---|---|
| 1 | 1 | Master , Computer , Technology , | Automation , Test , Engineer , | Development , IT , Network , Project , Sales , |
| 2 | 2 | Bachelor , Computer , Technology , | Information , Security , Engineer , | linux , unix , network , monitoring , incident , response , systems , administration , security , accessmer |
| 3 | 3 | Bachelor , Computer , Technology , | Network , Engineer , | Cisco , DNS , HTTP , Networking , Network , Engineer , Security , Video , VPN , Wireless , |
| 4 | 4 | Bachelor , Computer , Science , | Sr , Web , Application , Developer , | NET , C# , MVC , RESTful , web , services , HTTP , AWS , Azure , OOP , |
| 5 | 5 | Bachelor , Computer , Science , | Front , End , Developer , | C++ , Developer , Development , JavaScript , User , Interface , |
| 6 | 6 | Master , Computer , Science , | OpenStack , Engineer , | Openstack , |
| 7 | 7 | Master , Computer , Technology , | Data , Security , Administrator , | Unix , IAM , Scripting , knowledge , OIM , Windows , Linux , |
| 8 | 8 | Bachelor , Computer , Science , | Software , Engineer , Manager , | Java , OSS , |
| 9 | 9 | Bachelor , Computer , Technology , | Data , Security , Administrator , | Unix , IAM , Scripting , knowledge , OIM , Windows , Linux , |
| 10 | 10 | Bachelor , Computer , Science , | Project , Manager , | mobile , device , |
| 11 | 11 | Master , Computer , Science , | Test , Lead , Test , Manager , | QA , |
| 12 | 12 | Master , Computer , Technology , | Messaging , Administrator , | Desktop , Hardware , Network , Software , Systems , Windows , |
| 13 | 13 | Master , Computer , Technology , | Messaging , Administrator , | Desktop , Hardware , Network , Software , Systems , Windows , |
| 14 | 14 | Bachelor , Computer , Technology , | Sr , Systems , Test , Engineer , | System , Testing , Load , Performance , Testing , FIX , FAST , C , C++ , Java , Windows , Linux , |
| 15 | 15 | Master , Computer , Technology , | IT , Quality , Consultant , | Agile , CASE , Consulting , Defect , Management , Manager , Management , Project , Project , Manag |
| 16 | 16 | Bachelor , Computer , Technology , | USB , Validation , Engineer , | MATLAB , LabVIEW , |

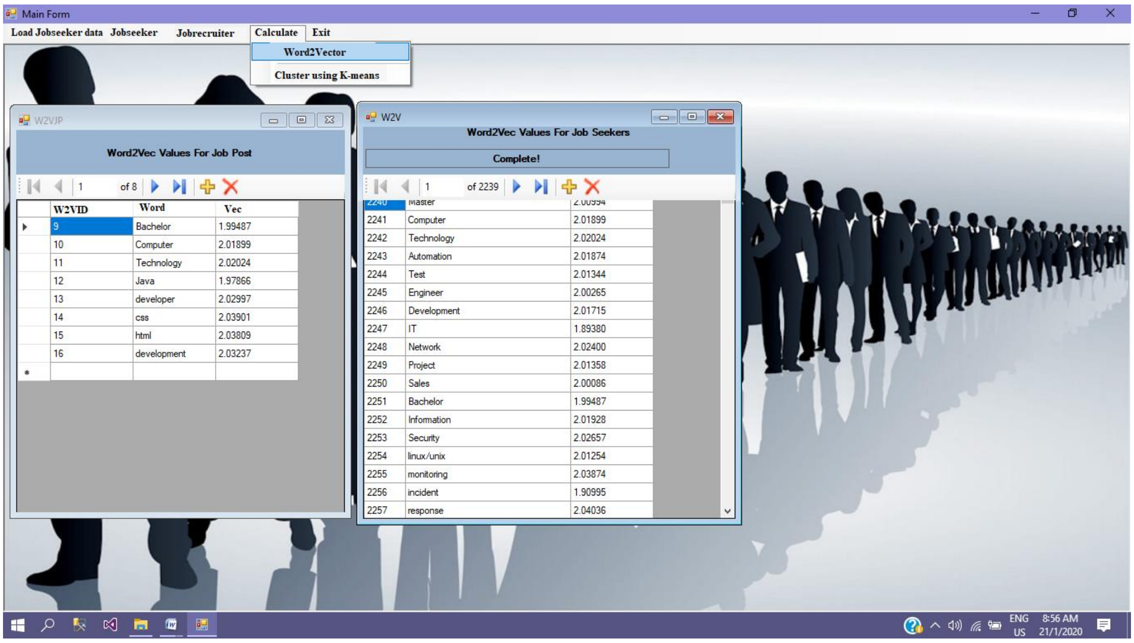
**Figure 4.8 Removing Stopwords of Job Seeker Data**

After preprocessing the job seekers' data, the job recruiter data are also to be loaded/ posted and processing must be the same processing steps as job seekers' data processing.

## 4.6 Calculating Word2vec and Clustering using K-means

After both data of job seeker and job recruiter data are tokenized and removed stop-words, user can calculate word2vec values as shown in Figure 4.9. This system used the Skip-gram Word2vector tool to transform the preprocessed job seekers' data and job recruiters' data.

Word2vec is a class of models that represents a word in a large text corpus as a vector in n-dimensional space or n-dimensional feature space bringing similar words closer to each other. One such model is the Skip-Gram model. **Skip-gram** model is one of the most important concepts in Natural Language Processing. The word2vector transformed job post and job seekers data are as shown in Figure 4.9.



Word2Vec Values For Job Post

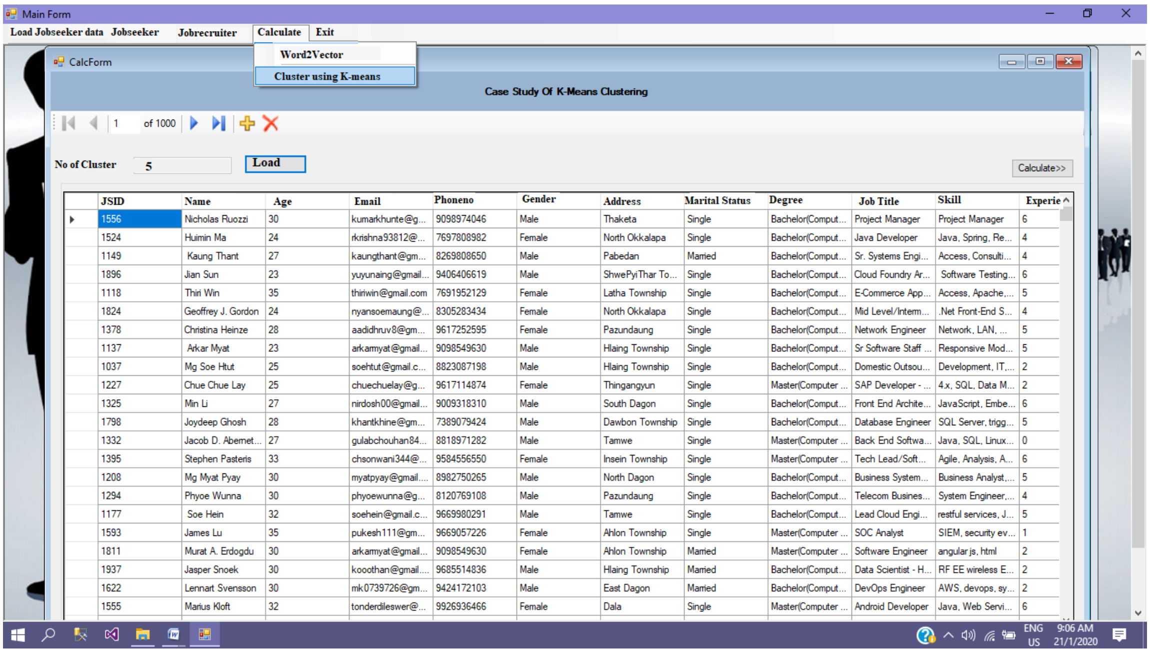
| W2VID | Word | Vec |
|---|---|---|
| 9 | Bachelor | 1.99487 |
| 10 | Computer | 2.01899 |
| 11 | Technology | 2.02024 |
| 12 | Java | 1.97866 |
| 13 | developer | 2.02997 |
| 14 | css | 2.03901 |
| 15 | html | 2.03809 |
| 16 | development | 2.03237 |

Word2Vec Values For Job Seekers

| | | |
|---|---|---|
| 2240 | Master | 2.00994 |
| 2241 | Computer | 2.01899 |
| 2242 | Technology | 2.02024 |
| 2243 | Automation | 2.01874 |
| 2244 | Test | 2.01344 |
| 2245 | Engineer | 2.00265 |
| 2246 | Development | 2.01715 |
| 2247 | IT | 1.89380 |
| 2248 | Network | 2.02400 |
| 2249 | Project | 2.01358 |
| 2250 | Sales | 2.00086 |
| 2251 | Bachelor | 1.99487 |
| 2252 | Information | 2.01928 |
| 2253 | Security | 2.02657 |
| 2254 | linux/unix | 2.01254 |
| 2255 | monitoring | 2.03874 |
| 2256 | incident | 1.90995 |
| 2257 | response | 2.04036 |

**Figure 4.9 Changing Word2Vec Values for Job Post and Job Seekers Data**

And then, the system user can load job seeker dataset to calculate as shown in Figure 4.10. In this system, five numbers of clusters are used as default to evaluate the job matching by using K-means.



Case Study Of K-Means Clustering

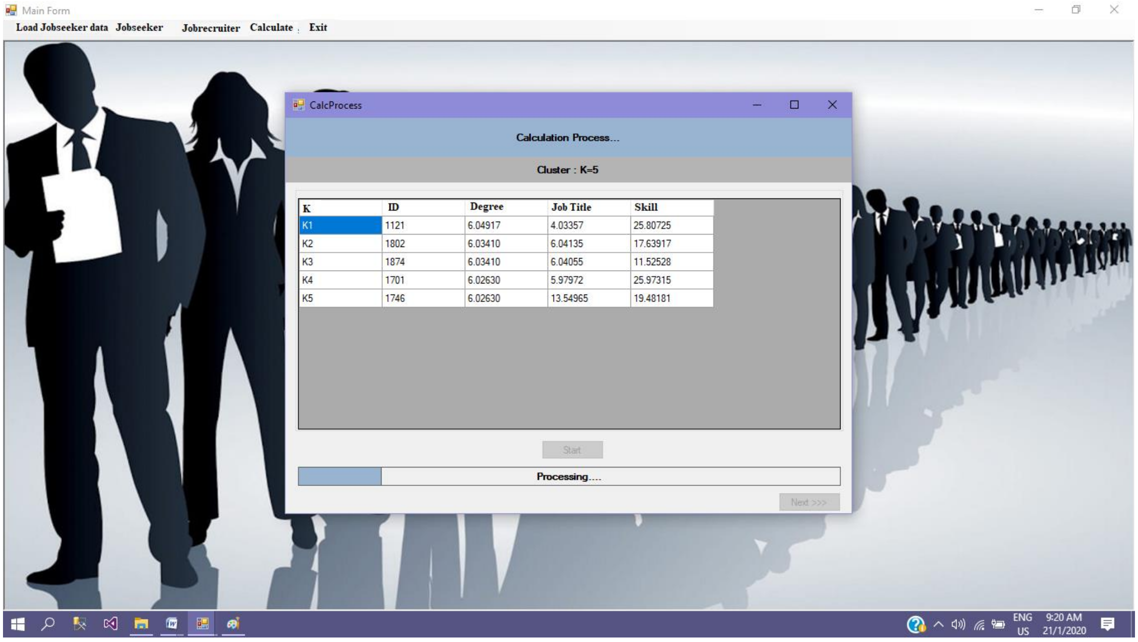
No of Cluster 5

| JSID | Name | Age | Email | Phoneno | Gender | Address | Marital Status | Degree | Job Title | Skill | Experie |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1556 | Nicholas Ruozzi | 30 | kumarkhunte@g... | 9098974046 | Male | Thaketa | Single | Bachelor(Comput... | Project Manager | Project Manager | 6 |
| 1524 | Huimin Ma | 24 | rkrishna93812@... | 7697808982 | Female | North Okkalapa | Single | Bachelor(Comput... | Java Developer | Java, Spring, Re... | 4 |
| 1149 | Kaung Thant | 27 | kaungthant@gm... | 8269808650 | Male | Pabedan | Married | Bachelor(Comput... | Sr. Systems Engi... | Access, Consulti... | 4 |
| 1896 | Jian Sun | 23 | yuyunaing@gmail... | 9406406619 | Male | ShwePyiThar To... | Single | Bachelor(Comput... | Cloud Foundry Ar... | Software Testing... | 6 |
| 1118 | Thiri Win | 35 | thiriwin@gmail.com | 7691952129 | Female | Latha Township | Single | Bachelor(Comput... | E-Commerce App... | Access, Apache,... | 5 |
| 1824 | Geoffrey J. Gordon | 24 | nyansoemaung@... | 8305283434 | Female | North Okkalapa | Single | Bachelor(Comput... | Mid Level/Interm... | .Net Front-End S... | 4 |
| 1378 | Christina Heinze | 28 | aadidhruv8@gm... | 9617252595 | Female | Pazundaung | Single | Bachelor(Comput... | Network Engineer | Network, LAN, ... | 5 |
| 1137 | Arkar Myat | 23 | arkarnyat@gmail... | 9098549630 | Male | Hlaing Township | Single | Bachelor(Comput... | Sr Software Staff ... | Responsive Mod... | 5 |
| 1037 | Mg Soe Htut | 25 | soehtut@gmail.c... | 8823087198 | Male | Hlaing Township | Single | Bachelor(Comput... | Domestic Outsou... | Development, IT,... | 2 |
| 1227 | Chue Chue Lay | 25 | chuechuelay@g... | 9617114874 | Female | Thingangyun | Single | Master(Computer ... | SAP Developer - ... | 4.x, SQL, Data M... | 2 |
| 1325 | Min Li | 27 | nirdosh00@gmail... | 9009318310 | Male | South Dagon | Single | Bachelor(Comput... | Front End Archite... | JavaScript, Embe... | 6 |
| 1798 | Joydeep Ghosh | 28 | khantkhine@gm... | 7389079424 | Male | Dawbon Township | Single | Bachelor(Comput... | Database Engineer | SQL Server, trigg... | 5 |
| 1332 | Jacob D. Abernet... | 27 | gulabchouhan84... | 8818971282 | Male | Tamwe | Single | Master(Computer ... | Back End Softwa... | Java, SQL, Linux... | 0 |
| 1395 | Stephen Pasteris | 33 | chsonwani344@... | 9584556550 | Female | Insein Township | Single | Master(Computer ... | Tech Lead/Soft... | Agile, Analysis, A... | 6 |
| 1208 | Mg Myat Pyay | 30 | myatpyay@gmail.... | 8982750265 | Male | North Dagon | Single | Bachelor(Comput... | Business System... | Business Analyst,... | 5 |
| 1294 | Phyoe Wunna | 30 | phyoewunna@g... | 8120769108 | Male | Pazundaung | Single | Bachelor(Comput... | Telecom Busines... | System Engineer,... | 4 |
| 1177 | Soe Hein | 32 | soehein@gmail.c... | 9669980291 | Male | Tamwe | Single | Bachelor(Comput... | Lead Cloud Engi... | restful services, J... | 5 |
| 1593 | James Lu | 35 | pukesh111@gm... | 9669057226 | Female | Ahlon Township | Single | Master(Computer ... | SOC Analyst | SIEM, security ev... | 1 |
| 1811 | Murat A. Erdogdu | 30 | arkarnyat@gmail... | 9098549630 | Female | Ahlon Township | Married | Master(Computer ... | Software Engineer | angular js, html | 2 |
| 1937 | Jasper Snoek | 30 | kooothan@gmail.... | 9685514836 | Male | Hlaing Township | Married | Bachelor(Comput... | Data Scientist - H... | RF EE wireless E... | 2 |
| 1622 | Lennart Svensson | 30 | mk0739726@gm... | 9424172103 | Male | East Dagon | Married | Bachelor(Comput... | DevOps Engineer | AWS, devops, sy... | 2 |
| 1555 | Marius Kloft | 32 | tonderdileswer@... | 9926936466 | Female | Dala | Single | Master(Computer ... | Android Developer | Java, Web Servi... | 6 |

**Figure 4.10 Loading Job Seeker Data to Calculate**

The loaded job seeker dataset is converted into vector using word2vec.And then, when “Calculate” button is clicked, five numbers of clusters is used and first iteration of k-means clustering is initialized and processed to cluster as shown in

Figure 4.11.And then, the system shows five clusters of job seekers data  as shown in Figure 4.12.



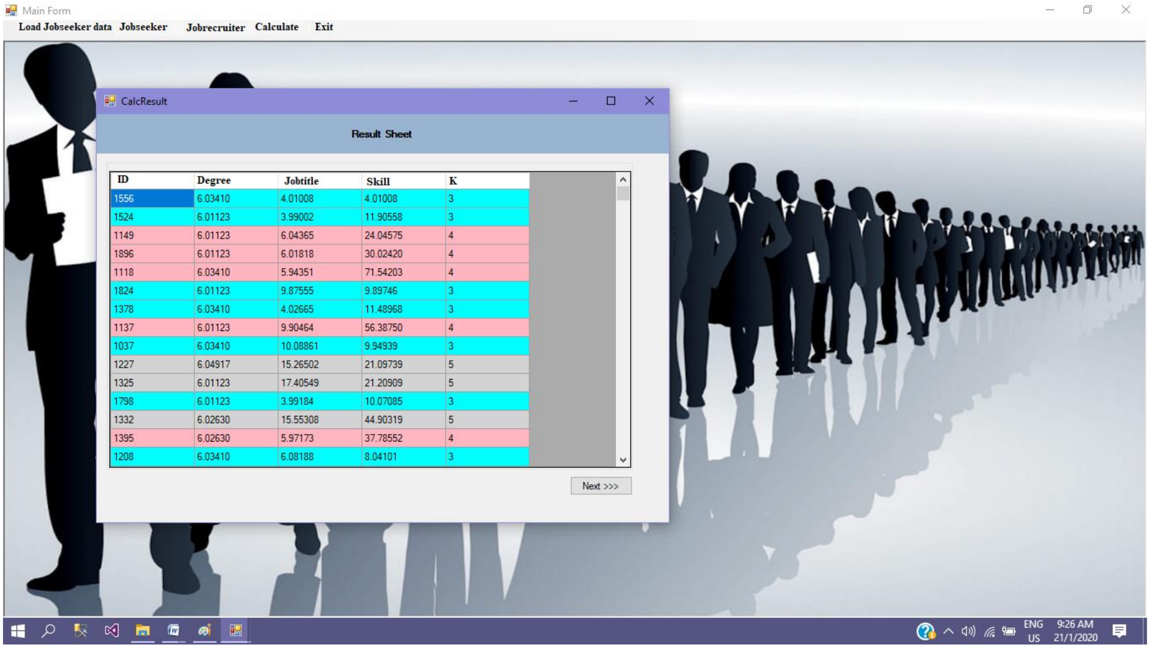
Calculation Process...

Cluster : K=5

| K | ID | Degree | Job Title | Skill |
|---|---|---|---|---|
| K1 | 1121 | 6.04917 | 4.03357 | 25.80725 |
| K2 | 1802 | 6.03410 | 6.04135 | 17.63917 |
| K3 | 1874 | 6.03410 | 6.04055 | 11.52528 |
| K4 | 1701 | 6.02630 | 5.97972 | 25.97315 |
| K5 | 1746 | 6.02630 | 13.54965 | 19.48181 |

Processing.....

**Figure4.11 Clustering Using K-means**

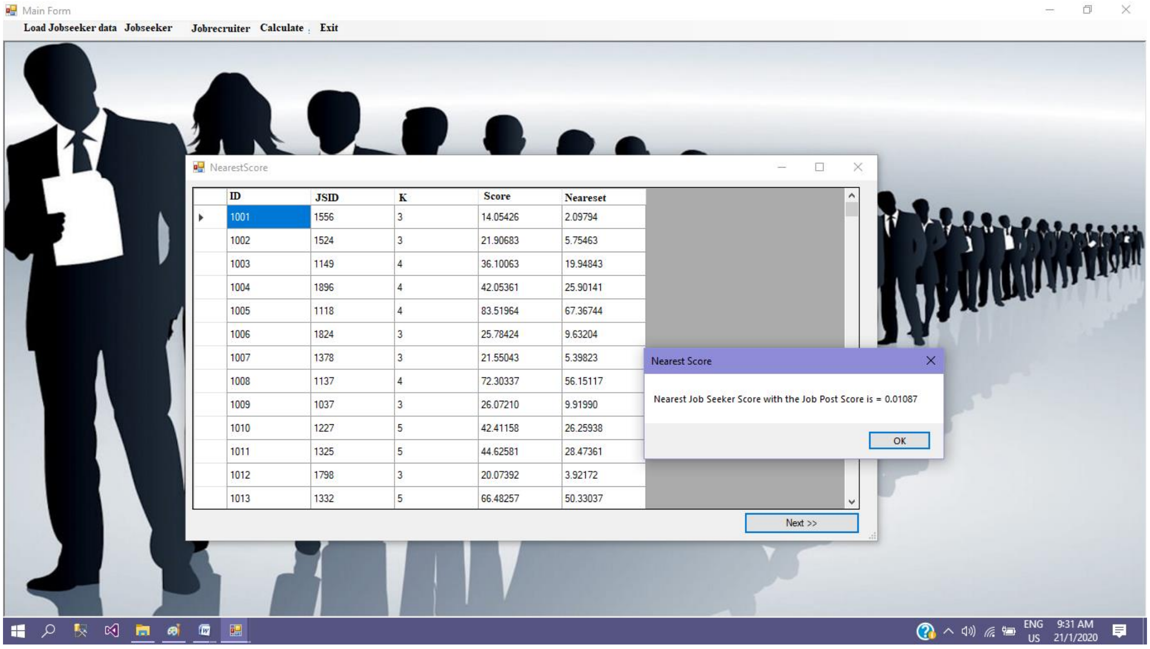


Result Sheet

| ID | Degree | Jobtitle | Skill | K |
|---|---|---|---|---|
| 1556 | 6.03410 | 4.01008 | 4.01008 | 3 |
| 1524 | 6.01123 | 3.99002 | 11.90558 | 3 |
| 1149 | 6.01123 | 6.04365 | 24.04575 | 4 |
| 1896 | 6.01123 | 6.01818 | 30.02420 | 4 |
| 1118 | 6.03410 | 5.94351 | 71.54203 | 4 |
| 1824 | 6.01123 | 9.87555 | 9.89746 | 3 |
| 1378 | 6.03410 | 4.02665 | 11.48968 | 3 |
| 1137 | 6.01123 | 9.90464 | 56.38750 | 4 |
| 1037 | 6.03410 | 10.08861 | 9.94939 | 3 |
| 1227 | 6.04917 | 15.26502 | 21.09739 | 5 |
| 1325 | 6.01123 | 17.40549 | 21.20909 | 5 |
| 1798 | 6.01123 | 3.99184 | 10.07085 | 3 |
| 1332 | 6.02630 | 15.55308 | 44.90319 | 5 |
| 1395 | 6.02630 | 5.97173 | 37.78552 | 4 |
| 1208 | 6.03410 | 6.08188 | 8.04101 | 3 |

**Figure4.12 Calculating Result for Five Clusters**

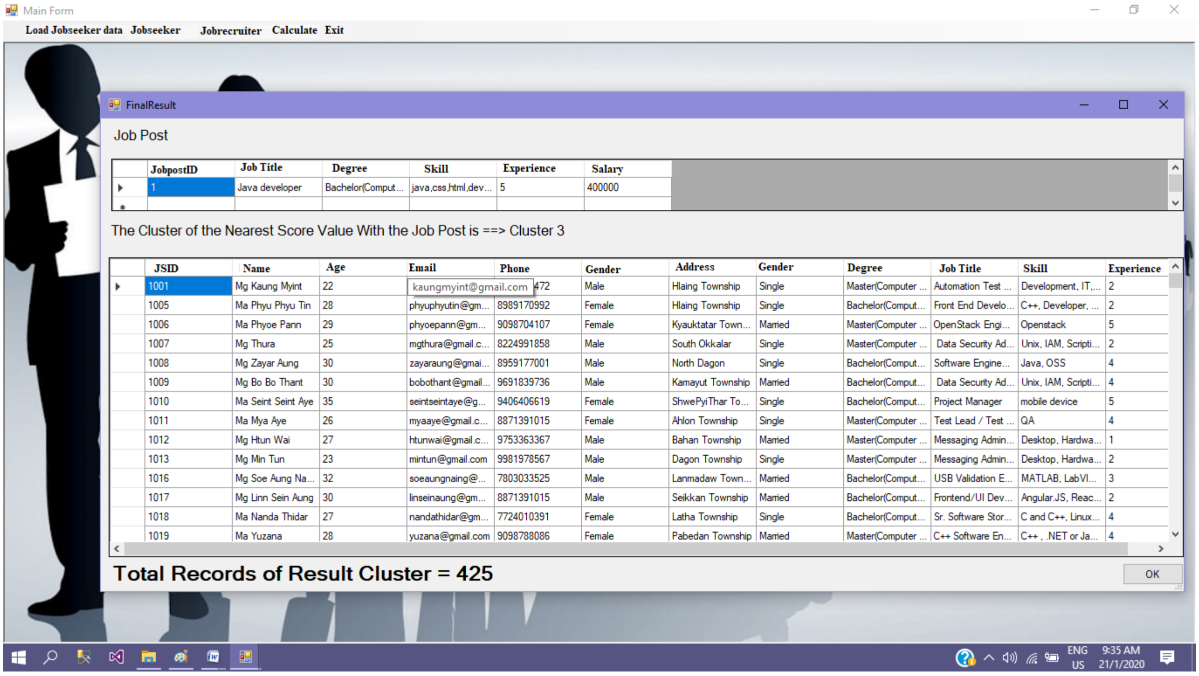
After calculating the result for five clusters, the system calculates total score of job vacancy and each job seeker. Total scores are calculated by summing all vector values of job title, degree and skill set of each job seeker and job recruiter. Then, the system shows the nearest score lists between each job seeker and job vacancy.



| ID | JSID | K | Score | Neareset |
|---|---|---|---|---|
| 1001 | 1556 | 3 | 14.05426 | 2.09794 |
| 1002 | 1524 | 3 | 21.90683 | 5.75463 |
| 1003 | 1149 | 4 | 36.10063 | 19.94843 |
| 1004 | 1896 | 4 | 42.05361 | 25.90141 |
| 1005 | 1118 | 4 | 83.51964 | 67.36744 |
| 1006 | 1824 | 3 | 25.78424 | 9.63204 |
| 1007 | 1378 | 3 | 21.55043 | 5.39823 |
| 1008 | 1137 | 4 | 72.30337 | 56.15117 |
| 1009 | 1037 | 3 | 26.07210 | 9.91990 |
| 1010 | 1227 | 5 | 42.41158 | 26.25938 |
| 1011 | 1325 | 5 | 44.62581 | 28.47361 |
| 1012 | 1798 | 3 | 20.07392 | 3.92172 |
| 1013 | 1332 | 5 | 66.48257 | 50.33037 |

Nearest Job Seeker Score with the Job Post Score is = 0.01087

**Figure 4.13 Showing Nearest Job Seeker Score with job vacancy**



Job Post

| JobpostID | Job Title | Degree | Skill | Experience | Salary |
|---|---|---|---|---|---|
| 1 | Java developer | Bachelor(Comput... | java,css,html,dev... | 5 | 400000 |

The Cluster of the Nearest Score Value With the Job Post is ==> Cluster 3

| JSID | Name | Age | Email | Phone | Gender | Address | Gender | Degree | Job Title | Skill | Experience |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1001 | Mg Kaung Myint | 22 | kaungmyint@gmail.com | 472 | Male | Hlaing Township | Single | Master(Computer ... | Automation Test ... | Development, IT,... | 2 |
| 1005 | Ma Phyu Phyu Tin | 28 | phyuphyutin@gm... | 8989170992 | Female | Hlaing Township | Single | Bachelor(Comput... | Front End Develo... | C++, Developer, ... | 2 |
| 1006 | Ma Phyoe Pann | 29 | phyoepann@gm... | 9098704107 | Female | Kyauktatar Town... | Married | Master(Computer ... | OpenStack Engi... | Openstack | 5 |
| 1007 | Mg Thura | 25 | mgthura@gmail.c... | 8224991858 | Male | South Okkalar | Single | Master(Computer ... | Data Security Ad... | Unix, IAM, Scripti... | 2 |
| 1008 | Mg Zayar Aung | 30 | zayaraung@gmai... | 8959177001 | Male | North Dagon | Single | Bachelor(Comput... | Software Engine... | Java, OSS | 4 |
| 1009 | Mg Bo Bo Thant | 30 | bobothant@gmail... | 9691839736 | Male | Kamayut Township | Married | Bachelor(Comput... | Data Security Ad... | Unix, IAM, Scripti... | 4 |
| 1010 | Ma Seint Seint Aye | 35 | seintseintaye@g... | 9406406619 | Female | ShwePyiThar To... | Single | Bachelor(Comput... | Project Manager | mobile device | 5 |
| 1011 | Ma Mya Aye | 26 | myaaye@gmail.c... | 8871391015 | Female | Ahlon Township | Single | Master(Computer ... | Test Lead / Test ... | QA | 4 |
| 1012 | Mg Htun Wai | 27 | htunwai@gmail.c... | 9753363367 | Male | Bahan Township | Married | Master(Computer ... | Messaging Admin... | Desktop, Hardwa... | 1 |
| 1013 | Mg Min Tun | 23 | mintun@gmail.com | 9981978567 | Male | Dagon Township | Single | Master(Computer ... | Messaging Admin... | Desktop, Hardwa... | 2 |
| 1016 | Mg Soe Aung Na... | 32 | soeaungnaing@... | 7803033525 | Male | Lanmadaw Town... | Married | Bachelor(Comput... | USB Validation E... | MATLAB, LabVI... | 3 |
| 1017 | Mg Linn Sein Aung | 30 | linseinaung@gm... | 8871391015 | Male | Seikkan Township | Married | Bachelor(Comput... | Frontend/UI Dev... | Angular.JS, Reac... | 2 |
| 1018 | Ma Nanda Thidar | 27 | nandathidar@gm... | 7724010391 | Female | Latha Township | Single | Bachelor(Comput... | Sr. Software Stor... | C and C++, Linux... | 4 |
| 1019 | Ma Yuzana | 28 | yuzana@gmail.com | 9098788086 | Female | Pabedan Township | Married | Master(Computer ... | C++ Software En... | C++ , .NET or Ja... | 4 |

Total Records of Result Cluster = 425

**Figure 4.14 Showing Final Result of Relevant Clustered Job Seeker Data**

And then, the nearest job seeker score with the job vacancy is calculated as shown in Figure 4.13. Finally, the system shows the relevant cluster of job seekers that match with job vacancy as shown in Figure 4.14.In this system, two types of data are categorized: hardware and software job posts. According to the recruiter’s job profile, the evaluated cluster is evaluated.

## 4.7System Evaluation of Job Matching

In this system, the accuracy method is used to calculate the relevancy of job matching method between recruiter's job profile and job seekers' profiles. In the system evaluation, as the predicted class, software job seekers and hardware job seekers are used and as the actual class, software job post and hardware job post are used as shown in Table 4.1. If software job seekers are evaluated for software job post, this situation is true positive and if no, it is false negative.

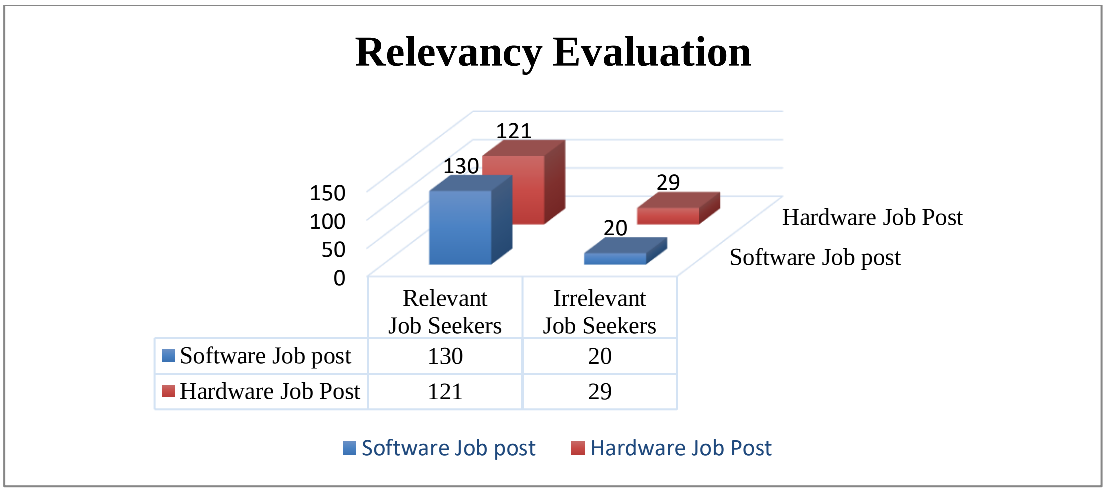
$$\text{Accuracy} = TP+TN/TP+FP+FN+TN \quad (4.1)$$

**Table4.1: Accuracy Formula Table**

| | Predicted Class | | |
|---|---|---|---|
| Actual Class | Category | Software Job Seekers | Hardware Job Seekers |
| | Software Job Post | True Positive | False Negative |
| | Hardware Job Post | False Positive | True Negative |

**Table4.2: System Evaluation for Job Matching**

| Category | Relevant job seekers | Irrelevant job seekers | Accuracy % |
|---|---|---|---|
| Software job post | 130 | 20 | 0.866666667 |
| Hardware job post | 121 | 29 | 0.806666667 |



| | Relevant Job Seekers | Irrelevant Job Seekers |
|---|---|---|
| 🟦 Software Job post | 130 | 20 |
| 🟥 Hardware Job Post | 121 | 29 |

**Figure4.15: Relevancy Evaluation Diagram**

The relevancy of matching between job seekers and job recruiters is evaluated by using Equation 4.1[10]. Evaluation for the relevancy of software job post and hardware job post are calculated.

During the clustering process the training data of job seekers posts "1000" and number of clusters “5” is used. According to the experimental result, true rate for relevant software job seeker posts are 130 and irrelevant posts are 20 records and true rate of relevant hardware job posts are 121 and irrelevant posts are 29. Therefore, the accuracy rate of software job and hardware job are 87% and 81% respectively as shown in Table 4.2. In Figure 4.15, the relevancy evaluation result is described for relevancy result of both software and hardware job recruiters’ job posts and job seekers’ profiles.

# CHAPTER 5
# CONCLUSION

This system isimplemented as a cluster-based job matching process that supports effective and easy matching between job posts of recruiters and profiles of job seekers. It can provide recommendation to the most suitable candidates for recruiters and job seekers can receiveany job vacancies notifications. Moreover, this system can solve the difficulties of traditional recruitment system and the manually work by posting in many job websites can be reduced. This system focus the job matching for IT related jobs especially for software related jobs and hardware related jobs. The accuracy of relevant matching is evaluated with classification accuracy method. According to the evaluation result, this system can provide the more relevant job matching between job seekers data and recruiter's job posts.

## 5.1. Benefits of the Proposed System

The previous studies used k-means clustering just provides the similar clusters of data but gives less relevant data. A job matching system is implemented using k-means and word2vec in this system. Word2vec is used to output the clusters with semantically similar words. As a result, using k-means clustering and word2vec model, recruiters can get the most relevant candidates that fit employers' needs even though they cannot get the identical job seekers that fit employers' requirements.

## 5.2. Limitation and Further Extension

This system can match for IT related jobs such as software jobs and hardware jobs. Other jobs that are not related with IT, cannot be applied in this system. Moreover, this system needs more time to convert semantic vector values of job seekers and recruiter data using word2vec model.

Beyond the work in this system, the proposed system can be extended to advance Job Matching System for all areas of job fields. The work can be extended and evaluated on a large resume dataset to find the relevant candidates. Supervised machine learning methods can be used based on the past history of selected candidates in order to rank the resumes and select the suitable candidate.

# AUTHOR'S PUBLICATION

[1] PhyoPyae Sone, Khine Moe Nwe, "Cluster-Based Job Matching System", the Conference on NationalParallel and Soft Computing (NPSC), University of Computer Studies, Yangon, Myanmar, 2020 (To be appeared).

# REFERENCES

[1] A. Drigas, “An Expert System for Job Matching of the Unemployed”, Expert Systems with ApplicationsVolume 26, Issue 2, Pages 217-224, February 2004.

[2] AnikaGupta, Deepak Garg, “Applying Data Mining Techniques in Job Recommender System for Considering Candidate Job Preferences”, International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, India, 24-27 September 2014.

[3] AzreenAzman, “Effective Method for Sentiment Lexical Dictionary Enrichment Based on Word2vec for Sentiment Analysis”,the 4th International Conference on Information Retrieval and Knowledge Management (ICIRKM),Kota Kinabalu, Malaysia, March 2018.

[4] Edgar Altszyler, Mariano Sigman,Sidarta Ribeiro and Diego FernándezSlezak, “Comparative Study of LSA vs Word2vec Embeddings in Small Corpora: ACase Study in Dreams Database”,arXiv: 1610.01520v2, April 2017.

[5] May Thu Naing, “Job Recommender System on Android Smartphone byUsing Hybrid Approach”,M.C.Sc., University of Computer Studies, Yangon, 2014.

[6] MayuriVerma, “Cluster Based Ranking Index for Enhancing Recruitment Process Using Text Mining and Machine Learning”, International Journal of Computer Applications (IJCA) (0975 – 8887) Volume 157 – No 9, January 2017.

[7] Mikolov T, Chen K, CorradoG, Dean J,”Efficient Estimation of Word Representations in Vector Space”, in Proceedings of Workshop at International Conference on Learning Representation (ICLR) , October2013.

[8] S. T. Al-Otaibi, M. Ykhlef, "A Survey of Job Recommender Systems", International Journal of the Physical Sciences (IJPS) , vol. 7, no. 29, pp. 5127-5142, July 2012.

[9] S. Zheng, W. Hong, N. Zhang, F. Yang, "Job Recommender Systems: A Survey", the 7th International Conference on Computer Science & Education(ICCSE), pp. 920-924, 14-17 July 2012.

[10] “Classification:Accuracy”.[Online]. Available:https://developers.google.com/machine-learning/crash-course/classification/ accuracy. (Accessed November 1, 2019).