Wi-Fi Floor Localization in an Unsupervised Manner

Liangliang Lin[1,2(✉)], Wei Shi[1], Muhammad Asim[1], Hui Zhang[1], Shuting Hu[3], and Jizhong Zhao[1]

[1] School of Computer Science and Technology, Department of Telecommunications, Xi'an Jiaotong University, Xi'an 710004, People's Republic of China
Lin_LL@126.com

[2] Information Department, Xi'an Conservatory of Music, Xi'an 710061, People's Republic of China

[3] Qinghai University, Xining 810016, Qinghai, People's Republic of China

Abstract. In recent decades, with the development of computer, indoor positioning applications have been developed rapidly. GPS has become one of the standards for outdoor positioning. However, there are great conditions for the use of GPS, GPS cannot be used indoors. At the same time, the indoor positioning scene has a great application prospect, through the use of indoor accessible signals (such as Wi-Fi, ZigBee, Bluetooth, UWB, etc.), according to the indoor environment and application, can be created based on the indoor positioning system. In the indoor positioning, there are two challenges, first of all, floor positioning, if the building has more than two layers, the second is planar positioning.

This paper solves the problem of floor positioning, and floor positioning based on Wi-Fi unsupervised recognition has attracted wide attention because it can get positioning results at a lower cost. In this paper, we try unsupervised indoor positioning methods, using only Wi-Fi crowdsourcing data. We get four months of data from seven-story buildings, by scanning the router's information. The application of neural network model can achieve unsupervised indoor positioning.

This clustering model aggregates all signals from the same floor into one class, and we use convolution neural networks, descending dimension feature extraction functions. The experiments show our solution obtains very high precision clustering results, so it can be summed up in this sense that the Wi-Fi crowdsourcing data can be used to locate in some way as the future direction of indoor positioning development.

Keywords: Indoor localization · CNN · K-means · Floor localization

This work is supported by NSFC Grants No. 61802299, 61772413, 61672424.

Q. Li et al. (Eds.): BROADNETS 2019, LNICST 303, pp. 52–69, 2019.
https://doi.org/10.1007/978-3-030-36442-7_4

1 Introduction

Indoor localization is the future for navigation purpose, in the current time of innovation, this is the most required work to be done for human beings' need. More than one & a half to two decades noteworthy research is happening in the area of indoor positioning. Which has guided to advancement of a few indoor localization frameworks (or arrangements) utilizing diverse signal systems for both research and business reasons. These arrangements are worked with various estimation strategies for example Lateration, Angulation, & Received Signal Strength. Accordingly, while building up an indoor localization framework it should be decided according to signal technologies accessible (as data origin) and estimation strategies which could be utilized along above mentioned advances. These choices are represented by the imperatives of utilization-case & for what execution measurements the framework is intended for. Conceptual diagram is exhibited under for recognizing distinctive indoor localization frameworks dependent on the signal system utilized as data origin in these frameworks with the end goal of positioning.

Nowadays, because of GPS and mobile phones in every person hands made a great achievement to know the location of a person or the mobile phone, but there is big hindrance to know the location of a person or mobile inside a building. GPS cannot detect the location inside the building, whenever the question of indoor localization comes on the tongue of any person, one must think of a storey building, any building having two or more floors, how can we determine the floor the person is at/on. The best way to sense the location of any mobile phone inside a wall blocked building in this modern period of technology is to use Wi-Fi modules. Because Wi-Fi modules are available in almost every building, so we can use the Wi-Fi RSS as access points(APs) to locate the mobile phones which are held by almost every person. There are many approaches to use Wi-Fi RSS to locate mobiles phone but the neural network approach has the greatest accuracy among others. We will elaborate the Artificial Neural Network, Deep Neural Network and Short Text Clustering technologies utilized in inside the building positioning systems.

The main contributions of our work can be summarized as follows:

- The real-time data of Wi-Fi RSS from a public mall building that gives a real experience of indoor surrounding. Analysis and results of each algorithm applied based on clustering performed which is accuracy attained.
- The method Latent Semantic Analysis was used for unsupervised dimensionality reduction because we have to learn deep feature representation from CNN in an unsupervised manner.
- Text clustering method was used in which there was one unsupervised dimensionality reduction function and one CNN model.

2 Related Works

2.1 iBeacon Based Floor Localization

iBeacon is a protocol systematized by Apple Inc. This is Apple's brand of a type of BLE-devices which broadcast & collect minute information for small ranges from surrounding portable electronic devices. This technology starts working when smart-phones, tablets and other devices are proximity to an iBeacon. They also look for applications in indoor localization systems that permits smart-phones to look for their estimated location by giving relative position information of smart-phone from an iBeacon in an Apple retail store. Consequently delivering proximity based indoor positioning system depends upon of that an iOS device receiving signal strength from an iBeacon can estimate its distance from other iBeacon. These ranges are sorted in these three ways: Immediate: within 2 m, Near: from 2 to 5 m, and Far: from 5 to 10 meters.

2.2 Short Text Clustering

Several studies have tried to conquer the scantiness of short text representation. There is a solution which is to extend and improve the setting of dataset. For instance, [1] proposed a strategy for enhancing the precision of short text clustering by enhancing their representation with extra features taken from Wikipedia, & [2] fuse semantic learning from a philosophy into text clustering. Be that as it may, this kind of work need strong NLP information and still utilize high-dimensional representation the outcome can be possibly in a misuse of memory as well as of calculation time. One more heading is to outline unique features in diminished space, for example, Latent Semantic Analysis (LSA) [3], Laplacian Eigenmaps (LE) [4], & Locality Preserving Indexing (LPI) [5]. Indeed, still a few scientists investigated some refined models to cluster short texts. For instance, [6] suggested a Dirichlet multinomial mixture model-based methodology for short text clustering. In addition, a few examinations even center the above both two streams. For instance, [7] suggested a novel system that advance the content features by utilizing machine interpretation and lessen the real features at the same time through matrix factorization methods.

2.3 Deep Neural Networks

As of late, it's a recovery of enthusiasm for DNN and numerous scientists have focused on utilizing Deep Learning to learn features. [8] utilize DAE to learn text representation. Amid the tweaking strategy, they utilize back-propagation to discover codes which are great at remaking the word count vector.

All the more as of late, scientists propose to utilize outer corpus to become familiar with a dispersed representation for every single word, known as word embedding [9], to improvise Deep Neural Network execution on NLP assignments. The Skip-gram and continuous bag-of-words models of Word2vec [10] suggest a basic single-layer architecture dependent upon the product of 2 word-vectors, & [11] present an another model for the representations of words, known as GloVe, that catches the worldwide corpus insights.

Motivated by Skip-gram of word2vec [10, 12], Skip-thought model [13] depicts a methodology for unsupervised learning of a nonexclusive, dispersed sentence encoder. Comparative as Skip-gram model, Skip-thought model [14] trains an encoder-decoder model which attempts to remake the encompassing sentences of an encoded sentence & discharged an off-the-rack encoder to remove sentence representation. This theory, let's have another point of view [15], advances a general self-trained Convolutional Neural Network structure that can adaptably couple different semantic features and accomplish a decent act upon single unsupervised learning task, short text clustering.

3 Methodology

To achieve the goal of floor localization, first we have to make clusters of each floor's Wi-Fi modules' Received Signal Strength, so that we can use the clusters of each floor to apply other algorithms to locate the exact floor of the mobile phone. My thesis is all about how to make clusters of each floors' Wi-Fi RSS. I have used a model of short text clustering to make clusters of the Wi-Fi RSS of each floor of a building. Because this model contains Convolutional Neural Network (CNN) model which empowers the clustering process like boosting it with a NOS gas cylinder. It is called Short Text Clustering (STC^2). To make use of short text clustering model, we had to convert the Wi-Fi RSS into text form, so we converted the Wi-Fi RSS data into matrix form to make it a feasible input for convolutional neural network. The main objective is explained in the Fig. 1, as follows:

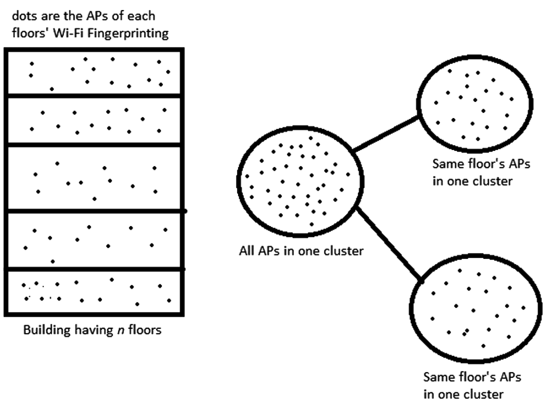

Fig. 1. Suppose a building having n number of floors, each floor have different amount of Wi-Fi modules, Access Points (Aps) or Wi-Fi RSS are represented as dots in the figure, all the Aps will be collected in one cluster first then they will divided into same floors' Aps clusters, we supposed that we are collecting only two floors Aps.

The aim is to make clusters of having some kind of resemblance between all short texts. So, let us suppose that we have a training texts dataset of n amount of numbers which is indicated as:

$$\mathbf{X} = \left\{\mathbf{X}_i\mathbf{:}\ \mathbf{X}_i \in \mathbb{R}^{d\ \bar{\times}\ 1}\right\}_{i=1,2,\ldots,n,} \tag{1}$$

In above equation d is called the imensionality of the original Baf-of-Words representation.

Let's denote its tag/label set as:

$$\mathbf{T} = \{1, 2, \ldots, \mathrm{C}\} \tag{2}$$

and for word embedded pre-trained set, it could be represented as:

$$\mathbf{E} = \left\{\mathbf{e}(w_{\mathrm{i}}):\ \mathbf{e}(w_{\mathrm{i}})?\ \mathbf{R}^{\mathrm{duxl}}\right\}_{i=1,2,\ldots,|\mathrm{V}|,} \tag{3}$$

where d_w is the dimensionality of word vectors and $|\mathrm{V}|$ is the vocabulary size. For the purpose to learn the r-dimensional deep feature representation **h** from Convolutional Neural Network in an unsupervised way, few unsupervised dimensionality reduction techniques f_{dr} (X) are utilized to manage the learning of Convolutional Neural Network model. We will probably cluster these texts X into clusters C dependent on the learned deep feature representation during saving the semantic continuity.

As delineated in Fig. 2, the framework we are inspired from comprises of 3 segments, deep convolutional neural system (CNN) or all the more especially Dynamic Convolutional Neural Network (DCNN), unsupervised dimensionality reduction function and K-means module. In the rest segments, starting with the initial 2 segments separately, and after that give the trainable parameters & the target function to help learn the deep feature representation.

In the end, the final part discloses the solution to execute clustering algorithm on the learned features.

Deep learning is the main core part of the Artificial Intelligence technology. In deep learning, you can also find Convolutional Neural Networks (CNN), more specifically the Deep Convolutional Neural Network (DCNN). There are few different kinds of deep convolutional neural networks but in this section, we sum up an overview of one popular deep convolutional neural network, Dynamic Convolutional Neural Network (DCNN) [16] as an example of CNN in the upcoming sections, that is as the establishment of this inspired solution has been effectively suggested for the totally supervised learning task, text classification.

In Fig. 3, just take a neural network having 2 convolutional layers for instance, the network transforms raw input text to a powerful representation. Specifically, every single raw text vector x_i is projected into a matrix representation $\mathrm{S} \in \mathrm{R}^{d\mathrm{w} \times s}$ by looking up a word embedding E, in which s represents the length of one text. Let us have $\tilde{\mathbf{W}} = \{\mathrm{W}_i\}_{i=1,2}$ and W_O indicate the weights of the neural networks. The network describes a transformation $f(\cdot) : \mathrm{R}^{d \times 1} \rightarrow \mathrm{R}^{r \times 1}(d)\ r)$ which transforms an input raw text x to a r-dimensional deep representation h. After that we can apply these 3 fundamental operations outlined as pursues:

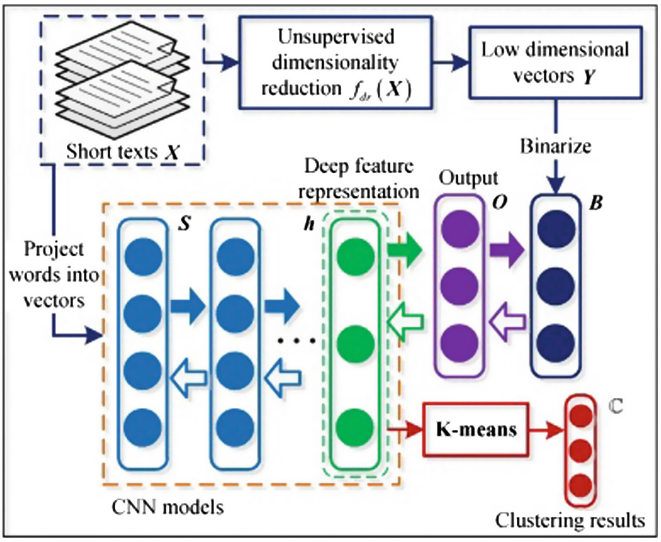

Fig. 2. The architecture of the framework we are inspired from called STC2 framework for short text clustering. Convolutional Neural Networks

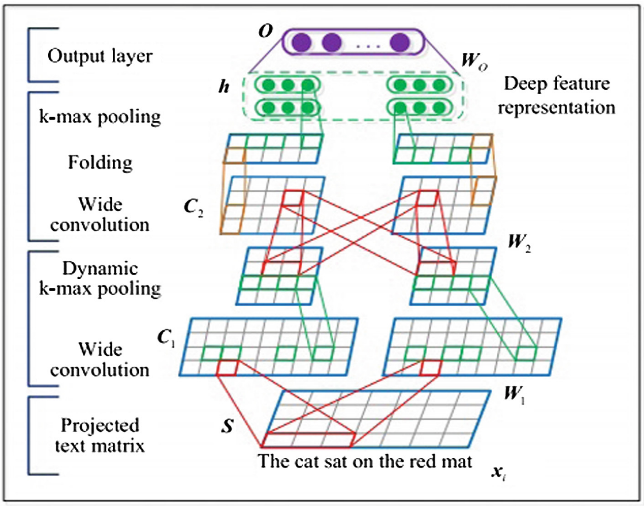

Fig. 3. In the dynamic convolutional neural network architecture [16]. After applying the process of word embedding, first we project the input text to the matrix feature, after that it passes through the wide convolutional layers, then folding layers & after that k-max pooling layers, after this process we are provided a deep feature representation just behind the output layer.

Wide One-Dimensional Convolution. This method $\mathrm{m} \in \mathrm{R}^{m}$ is employed to each single row of the sentence matrix $\mathrm{S} \in \mathbf{R}^{\mathrm{d_w} \times s}$, which produces a resulting matrix $\mathbf{C} \in \mathbf{R}^{\mathrm{d_w} \times (s+m-1)}$, m represents the convolutional filter's width.

Folding. In this method, every two rows in a feature map are easily added component wisely. A map having d_w rows, folding sends back a map of $d_w/2$ rows, consequently halving the size of the representation & producing a matrix feature as pursues:

$$\hat{\boldsymbol{C}} \in \bar{\mathbb{R}}^{(d_w/2) \times (s+m-1)}. \tag{4}$$

This is to be noted that the folding operation do not show any more than enough parameters.

Dynamic k-max Pooling. Supposing the pooling parameter as k, k-max pooling chooses the sub-matrix

$$\bar{\mathbf{C}} \in \bar{\mathbb{R}}^{(d_w/\bar{2})\bar{\times}k} \tag{5}$$

of the k highest values in every single row of the matrix $\hat{\mathbf{C}}$. For dynamic k-max pooling, the pooling parameter k is dynamically chosen for the purpose to permit for a smooth extraction of higher-order and longer-range features [16]. A constant pooling parameter k_{top} for the topmost convolutional layer, the parameter k of k-max pooling in the lth convolutional layer could be calculated as pursues:

$$k_1 = \max\left(k_{top}, \left\lceil \frac{L-l}{L} s \right\rceil\right) \tag{6}$$

where L is the total number of convolutional layers in the network.

3.1 Unsupervised Dimensionality Reduction

In pattern recognition, data mining and different sorts of data analysis applications, we regularly face high dimensional data. For instance, in face recognition, the extent of a training picture fix is generally bigger than 60 × 60, which relates to a vector with in excess of 3600 dimensions. Dimensionality reduction can likewise be viewed as the way toward determining a lot of degrees of opportunity which can be utilized to repeat the greater part of the changeability of a data set.

Unsupervised dimensionality reduction goes for representing to high-dimensional data in lower-dimensional spaces in a dedicated manner. Dimensionality reduction can be utilized for compression or denoising reasons, however data visualization still happens to be its highly noticeable applications. In the event that visualization is troublesome in high-dimensional space, maybe a (nearly) proportionate representation in a lower-dimensional space can improvise the lucidness of data. This is unequivocally the possibility which goes under the area of dimensionality reduction (DR).

There are a lot of different kinds of techniques for dimensionality reduction like the most popular Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Locality Preserving Indexing (LPI) and others but we are using Latent

Semantic Analysis, because it has the highest accuracy among other few techniques on our given data. As mentioned in Fig. 2, the dimensionality reduction algorithm can be represented as pursues:

$$\mathbf{Y} = f_{dr}(\mathbf{X}), \tag{7}$$

where, $Y \in R^{qxn}$ are the q-dimensional reduced latent space representations. Afterwards, we apply a famous dimensionality reduction method in this framework.

Latent Semantic Analysis (LSA) : LSA [3] is a very famous worldwide matrix factorization algorithm, that employs a dimension decreasing linear projection, Singular Value Decomposition (SVD), of the relating term/document matrix. Assume the rank of X is r̂, LSA deteriorates X in the result of 3 different matrices:

$$X = U \sum V^{T}$$

in above equation $\sum$ is equal to:

$$\sum = diag\ (\sigma_1, \ldots, \sigma_{\hat{r}})\ \text{and}\ \sigma_{\mathbf{1}} \geq \sigma_{\mathbf{2}} \geq \ \ldots \ \geq \ \sigma_{\hat{r}} \tag{8}$$

are the singular values of X, $U \in R^{dx\hat{r}}$ is a set of left singular vectors and $V \in R^{nx\hat{r}}$ is a set of right singular vectors. LSA utilizes the top q vectors in U as the transformation matrix to embed the original text features into a q-dimensional subspace Y [3].

The method mentioned above guarantees a superior exhibition in catching semantic similarity betwixt text in the reduced latent space representation Y compared to the original representation X, so, the execution of short text clustering could be additionally upgraded by the assistance of this system, self-taught Convultional Neural Network.

3.2 Training of the Model

ML (Machine Learning) is the consistent examination of estimations and quantifiable models that PC structures use to effectively play out a specific endeavor without using express rules, contingent upon precedents and deducing.

In unsupervised learning, the calculation assembles a numerical model from a dataset which contains just sources of info and no ideal yield marks. Unsupervised learning calculations are utilized to discover structure in the information, such as gathering or grouping of information focuses. Unsupervised learning can find designs in the information, and can bunch the contributions to classes, as in feature learning. Dimensionality reduction is the way toward diminishing the quantity of “features”, or inputs, in a dataset.

The short text clustering model we are using also contains a Convolutional Neural Network (CNN) model, more particularly Dynamic Convolutional Neural Network (DCNN). In this model the input is actually the word embedded form of the original raw text vectors. It has the deep feature respresentation h, which connected to the output in order to get the first random value from the binarized form of Low dimensional vector *Y*.

The final layer of CNN is an output layer as follows:

$$\mathrm{O} = \mathrm{W}_o\mathrm{h}, \tag{9}$$

in which, h is the deep feature representation, $\mathrm{O} \in \mathrm{R}^{\mathrm{q}}$ is the output vector & $\mathrm{W}_{\mathrm{o}} \in \mathrm{R}^{\mathrm{qxr}}$ is weight matrix.

For the purpose to integrate the latent semantic features Y, firstly, we have to binarize the real-valued vectors Y to the binary codes B by fixing the threshold to be the median vector *median* (Y). At that point, the output vector O is utilized to set the binary codes B through q logistic operations as pursues:

$$p_i = \frac{\exp(\mathbf{O}_{\mathrm{i}})}{1 + \exp(\mathbf{O}_{\mathrm{i}})}$$

All parameters to be trained are denoted as θ

$$\theta = \{\mathbf{E}, \tilde{\mathbf{W}}, \mathbf{W}_o\}. \tag{10}$$

Given the training text collection X, & the pre-trained binary codes B, the log probability of the parameters could be formulated as pursues:

$$\mathrm{J}(\boldsymbol{\theta}) = \sum_{i=1}^{\mathrm{n}} \log p\,(\mathbf{b}_i|\mathbf{X}_{\mathrm{i}},\boldsymbol{\theta}). \tag{11}$$

We have followed the already work done by [16], by the method of back propagation, we train the network with mini batches & applied the gradient based optimization utilizing the Adagrad update rule [17]. For regularization, we utilize dropout with 50% to the penultimate layer [16, 18].

3.3 K-Means for Clustering

K-means clustering is a procedure for vector quantization, at first started from signal processing, which is conspicuous for cluster analysis in data mining. K-means clustering plans to fragment n observations into k clusters, after that every single observation has a spot with the cluster with the nearest mean, filling in as a clustering model. Which concludes in a division of the data space into Voronoi cells.

Let us suppose we are given set of observations (x_1, x_2, …, x_n), in which every single observation is a d-dimensional real vector, k-means clustering intends to divide the n observations into k which is less than or equal to n sets $\mathrm{S} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within cluster sum of squares (WCSS) (i.e. variance or change). Suitably, the goal is to discover:

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{X} - \boldsymbol{\mu}_i\|^2 = \underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} |S_i| \ \mathrm{Var}\ S_i$$

in which μ_i is the mean of points in S_i. This is almost equal to minimizing the pairwise squared variations of points in the identical cluster:

$$\underset{\mathrm{S}}{\arg\min} \sum_{i=1}^{k} \frac{\mathbf{1}}{2|S_i|} \sum_{\mathrm{x,y} \in s_i} \|\mathrm{x - y}\|^{\mathbf{2}}$$

The equivalent couln be derived by this equation:

$$\sum_{\mathrm{X} \in S_i} \|\mathrm{X} - \boldsymbol{\mu}_i\|^2 = \sum_{\mathrm{x} \neq \mathrm{y} \in S_i} (\mathrm{X} - \boldsymbol{\mu}_i)(\boldsymbol{\mu}_i - \mathrm{y})$$

The total variance is fixed, it is almost equal to maximizing the sum of squared fluctuations betwixt points in *different* clusters (between cluster sum of squares, BCSS), the law of total variance is followed.

Finally getting the short texts, then initially use the trained deep neural network to acquire the semantic representations h, afterward utilize conventional K-means function to accomplish clustering.

4 Evaluation

4.1 Experimental Setup

We test the model on the given dataset by Tencent company, they have given us the dataset of a shopping mall, shopping mall had 7 floors. All the Wi-Fi RSS from all the 7 floors' Wi-Fi modules. They have used different mobile phones models of different brands to collect the data from the Wi-Fi routers of the chosen shopping mall, the time period of the collected data is approximately 4 months. Different mobile phones have different strength of catching or detecting the signals from each floor, so they have collected the data in a way so that it can be supposed like a public experiment. Because for public, everyone has a different kind of mobile phone, different kind of mobile phone brand, so that is why they have tried to collect it like a public user.

Tencent company has given one dataset in which there are approximately 2 million Received Signal Strength (RSS) of all the Wi-Fi modules on all seven floors of the shopping mall. The given dataset includes both labeled and unlabeled data. We took a sample set from the given data set, we took almost 20,000 RSS from the original dataset of all seven floors. Our experiment or project is to cluster only two floors, so we chose first and the seventh floor from our sample dataset. We did not use the RSS from second to sixth floor. First, we separated a sample data set from the original RSS dataset then we collected the RSS of first and the seventh floor from our collected sample dataset.

In order to convert Wi-Fi RSS into text data, we processed the data, the convolutional neural network input is a matrix, in which each row of the matrix is representing the sentence and each column is representing the word2vector encoding of each word in the sentence. We had the address of the Wi-Fi router with its RSS, so in this paired form (Wi-Fi address, RSS), we received the data, so for each pair, we repeat

(Wi-Fi address, RSS) up to the value we got from adding 100 to the RSS, so it forms a line of sentences, then carry out word2vector training and finally input to the CNN model.

As my graduate project, I picked only 1st and 7th floors Wi-Fi RSS from the sampled data which we picked from the original 7 floors dataset. In which first floor had 40,000 (Wi-Fi address, RSS) pairs and seventh floor had 18,000 (Wi-Fi address, RSS) pairs. The matrix we formed for the input, each row of the matrix has 20 pairs of (Wi-Fi address, RSS), so for first floor we calculate the number of rows for the matrix by the formula as follows:

Number of rows is equal to the number of floors (Wi-Fi address, RSS) pairs divided by number of pairs in each row

Rows = Pairs/20

For the 1st floor: No: of rows = 40000/20 = 2000

For the 7th floor: No: of rows = 18000/20 = 900

So, now we have an uneven quantity of rows, which is not feasible to train, so we choose to make them an even and equal number of rows, for that purpose we select 900 rows as minimum amount of rows, so we have to find a way to decrease the number of rows of first floor. Although it was the toughest decision to do it by the help of computer but the easiest way too. So we run a random function in MATLAB to choose the random values from data and delete them from data until the data has 900 rows which is equal to 7th floors' number of rows.

For the input for the Unsupervised Dimensionality Reduction, we used the labeled first and the seventh floors' RSS, as shown in the Table 1.

Table 1. Transformed input matrix for CNN on the sample data of only first and the seventh floor

Dataset	Number of rows	(Wi-Fi Address, RSS) Pairs
1st Floor	900	18000
2nd Floor	900	18000
Total	1800	36000

4.2 Pre-trained Word Vectors

The purpose to input dataset into the Dynamic Convolutional Neural Network model, we used the same sample data of the first and the seventh floors and utilized the freely accessible word2vec tool for the training purpose of word embeddings, almost all the parameters are same as in [10] for the training of word vectors on Google News setting, aside from of vector dimensionality utilizing 48 and minimize count utilizing 5. The inclusion of these learned vectors on our example dataset is recorded in Table 1, and the words not existing in the set of pre-trained words are instated haphazardly.

4.3 Evaluation Metrics

The clustering execution is assessed by contrasting the clustering results of data and the labels/tags given by the text corpus. Two metrics, the accuracy (ACC) & the normalized mutual information metric (NMI), are utilized to quantify the clustering execution [19, 20]. Given a text x_i, let c_i and t_i be the gotten cluster label and the label given by the corpus, separately. Accuracy is calculated by this formula:

$$ACC = \frac{\sum_{i=1}^{n} \delta(t_i, map(c_i))}{n},$$

in which, n is the total number of texts, $\delta(x, y)$ is the indicator function that becomes 1 if x is equal to y and becomes zero if x is not equal to y, and $map(c_i)$ is the permutation mapping function which maps every single cluster label c_i to the almost equal label from the text data by Hungarian algorithm [21].

Normalized mutual information [22] betwixt tag/label set T and cluster set C is a popular metric utilized for calculating clustering tasks. NMI is denoted as pursues:

$$NMI(\mathbf{T},\mathbb{C}) = \frac{MI(\mathbf{T},\mathbb{C})}{\sqrt{H(\mathbf{T})H(\mathbb{C})}} \tag{12}$$

in which, MI(T,C) represents mutual information betwixt T and C, H(.) represents entropy and $\sqrt{H(\mathbf{T})H(\mathrm{C})}$ is utilized for normalizing the mutual information for limiting it between 0 and 1.

4.4 Settings for Hyperparameter

A large portion of the parameters are set consistently for the sampled dataset. This CNN model [15], there are 2 convolutional layers in the network. The convolutional filters' width has set to be 3 for both of them. The top k-max pooling's value in Eqs. (3–4) has set to be 5 for k. On the first convolutional layer the number of feature maps is 12, and on the second convolutional layer there are 8 feature maps. After the both convolutional layers there is a folding layer. Moving on to the dimension of word embeddings, we set d_w as 48. At long last, the dimension of the deep feature representation r is set to 480. Also, we fix the learning rate λ to 0.01 and the mini batch training size to 200 only for STC2-Latent Semantic Analysis and Latent Dirichlet Allocation (LDA) and mini batch training size to 16 for all other methods. In Eqs. (3–7), q is the output size which is fixed equivalent to the best dimensions of subspace in the benchmark method.

For beginning centroids have noteworthy effect on the results of clustering while using the K-means method, we rehash K-means for numerous times with arbitrary starting centroids (explicitly, for statistical significance 100 times) in [20]. The every subspace vectors are normalized to 1 preceding applying K-means and the last outcomes revealed are the half of the sum of five evaluations with all the clustering algorithms on our sampled dataset.

4.5 Results and Analysis

The details of different accuracy (ACC) and normalized mutual information (NMI) percentage of according to the change of epoch size on the model are shown in Table 2. The method we used for Unsupervised Dimensionality Reduction is Latent Dirichlet Allocation (LDA). Applying this method to the short text clustering model we get some fine results. On different size epoch we get variance in the accuracy (ACC) and the normalized mutual information (NMI). Detailed ACC and NMI are in the table below:

Table 2. ACC and the NMI of epoch size 1–6, at epoch size 2, the ACC is the highest.

Latent Dirichlet Allocation (LDA) in unsupervised dimensionality reduction		
Epoch (size)	ACC (%)	NMI (%)
1	61.385	8.9928
2	61.4125	8.9652
3	61.3899	8.9341
4	61.2851	8.9159
5	61.1852	8.0905
6	61.0258	8.0058
7	60.369	7.9928

In Table 3, you can see the details of different accuracy (ACC) and normalized mutual information (NMI) percentage of according to the change of epoch size on the model. The method we used for Unsupervised Dimensionality Reduction is Average Embedding (AE). Applying this method to the short text clustering model we get some fine results. On different size epoch we get variance in the accuracy (ACC) and the normalizaed mutual information (NMI). Detailed ACC and NMI are in the table below:

Table 3. ACC and the NMI of epoch size 1–6, at epoch size 2, the ACC is the highest.

Average Embedding (AE) in unsupervised dimensionality reduction		
Epoch (size)	ACC (%)	NMI (%)
1	66.385	7.9928
2	66.3537	7.9652
3	66.3099	7.9431
4	66.385	7.9928
5	66.4852	8.0905
6	66.3788	8.0058
7	66.385	7.9928

In Table 4, you can see the details of different accuracy (ACC) and normalized mutual information (NMI) percentage of according to the change of epoch size on the model. The method we used for Unsupervised Dimensionality Reduction is Spectral Laplacian Eigenmaps (Spectral-LE). Applying this method to the short text clustering model we get some fine results. On different size epoch we get variance in the accuracy (ACC) and the normalizaed mutual information (NMI). Detailed ACC and NMI are in the table below:

Table 4. ACC and the NMI of epoch size 1–6, at epoch size 1, the ACC is the highest.

Spectral Laplacian Eigenmaps (Spectral-LE) in unsupervised dimensionality reduction		
Epoch (size)	ACC (%)	NMI (%)
1	83.2332	47.3152
2	83.2112	47.3100
3	83.2008	47.3023
4	83.1933	47.2933
5	83.1900	47.2902
6	83.1806	47.0126

In Table 5, shown the details of different accuracy (ACC) and normalized mutual information (NMI) percentage of according to the change of epoch size on the model. The method we used for Unsupervised Dimensionality Reduction is Latent Semantic Analysis (LSA). Applying this method to the short text clustering model we get some fine results. On different size epoch we get variance in the accuracy (ACC) and the normalizaed mutual information (NMI). Detailed ACC and NMI are in the table below:

Table 5. ACC and the NMI of epoch size 1–6, at epoch size 1, the ACC is the highest.

Latent Semantic Analysis (LSA) in Unsupervised Dimensionality Reduction		
Epoch (size)	ACC (%)	NMI (%)
1	60.0638	4.9477
2	60.0603	4.9463
3	60.0563	4.9449
4	60.0530	4.9432
5	60.0513	4.9396
6	60.0496	4.9385

This algorithm Average Embedding (AE) with Short Text Clustering method is shown in Table 6, shown the details of different accuracy (ACC) and normalized mutual information (NMI) percentage of according to the change of epoch size on the model. Applying this method to the short text clustering model we get some fine results. On different size epoch we get variance in the accuracy (ACC) and the normalizaed mutual information (NMI). Mini batch size is set to be 16 for this method and detailed ACC and NMI are in the table below:

Table 6. ACC and the NMI of epoch size 1–6, at epoch size 1, the ACC is the highest.

STC2-Average Embedding (STC2-AE) in unsupervised dimensionality reduction		
Epoch (size)	ACC (%)	NMI (%)
1	64.6013	6.4831
2	68.3164	10.076
3	68.6075	10.3594
4	68.5349	10.3079
5	68.6657	10.4475
6	68.6523	10.4325

Another algorithm with Short Text Clustering method is shown in Table 7, called Latent Semantic Analysis (STC2-LSA), shown the details of different accuracy (ACC) and normalized mutual information (NMI) percentage of according to the change of epoch size on the model. Applying this method to the short text clustering model we get some fine results. On different size epoch we get variance in the accuracy (ACC) and the normalizaed mutual information (NMI). Mini batch size is set to 200 for this method and detailed ACC and NMI are in the table below:

Table 7. ACC and the NMI of epoch size 1–6, at epoch size 1, the ACC is the highest.

STC2-Latent Semantic Analysis (STC2-LSA) in unsupervised dimensionality reduction		
Epoch (size)	ACC (%)	NMI (%)
1	77.1674	23.5357
2	78.635	26.9315
3	78.4185	26.0886
4	77.8677	24.7327
5	77.6968	24.3019
6	70.4093	14.7743

One more algorithm with Short Text Clustering method is shown in Table 8, called Laplacian Eigenmaps (STC2-LE), shown the details of different accuracy (ACC) and normalized mutual information (NMI) percentage of according to the change of epoch size on the model. Applying this method to the short text clustering model we get some fine results. On different size epoch we get variance in the accuracy (ACC) and the normalizaed mutual information (NMI). Mini batch size is set to 16 for this, detailed ACC and NMI are in the table below:

Table 8. ACC and the NMI of epoch size 1–6, at epoch size 1, the ACC is the highest.

STC2-Laplacian Eigenmaps (STC2-LE) in unsupervised dimensionality reduction		
Epoch (size)	ACC (%)	NMI (%)
1	82.2819	38.8306
2	82.2381	42.5264
3	82.0092	42.1357
4	81.8671	41.9982
5	81.7322	41.8866
6	81.5236	41.7569

The results we successfully got, with two measures of ACC and NMI on our sampled dataset, we get some promising results from our applied Unsupervised Dimensionality Reduction method STC2-Laplacian Eigenmaps (STC2-LE), which concludes that the approach is the effective approach to collect some handy semantic features for clustering of the Wi-Fi RSS of the buildings.

5 Conclusion

In this paper, we work on clustering for the use of indoor localization, making use of Artificial Neural Network and other algorithms, was experimented, tested and evaluated with several methods of Unsupervised Dimensionality Reduction methods. The goal was to successfully put same floors' Wi-Fi RSS to a cluster where only exists the same floors' Wi-Fi RSS.

After a comprehensive study of earlier research in indoor localization systems based on various signal technologies, an approach to successfully cluster the same floors' RSS was developed by the help of a model called short text clustering. Short text clustering was chosen because of the inclusion of Convolutional Neural Network model in it, and the signal technology approach, for that we chose Wi-Fi routers inside the building, it was chosen because of the fact that it is readily available in the buildings. Research was conducted to see with which Unsupervised Dimensionality Reduction method the clustering has the highest accuracy, so Latent Semantic Analysis (LSA) method was used because of the highest accuracy factor. The conclusion of the evaluation is that short text clustering model can be considered as a viable candidate for clustering the Wi-Fi RSS into same clusters.

The system developed can give a 78% accuracy of clustering two floors as compared to other methods and algorithms. These results are good enough to continue this approach for the other floors of the building.

References

1. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 787–788. ACM (2007)
2. Fodeh, S., Punch, B., Tan, P.-N.: On ontology-driven document clustering using core semantic features. Knowl. Inf. Syst. **28**(2), 395–421 (2011)
3. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JAsIs **41**(6), 391–407 (1990)
4. Ng, A.Y., Jordan, M.I., Weiss, Y., et al.: On spectral clustering: Analysis and an algorithm. Adv. Neural. Inf. Process. Syst. **2**, 849–856 (2002)
5. He, X., Niyogi, P.: Locality preserving projections. In: Neural Information Processing Systems, vol. 16, pp. 153–160. MIT (2004)
6. Yin, J., Wang, J.: A dirichlet multinomial mixture model-based approach for short text clustering. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 233–242. ACM (2014)
7. Tang, J., Wang, X., Gao, H., Hu, X., Liu, H.: Enriching short text representation in microblog for clustering. Front. Comput. Sci. **6**(1), 88–101 (2012)
8. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
9. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394 (2010)
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
11. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of the Empiricial Methods in Natural Language Processing, p. 12 (2014)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781(2013)
13. Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., et al.: Skip-thought vectors. In: Advances in Neural Information Processing Systems, pp. 3276–3284 (2015)
14. Wang, X., Gupta, A. Unsupervised learning of visual representations using videos, arXiv preprint arXiv:1505.00687. (2015)
15. Jiaming, X., Bo, X., Wang, P., Zheng, S., Tian, G., Zhao, J.: Self-taught convolutional neural networks for short text clustering. Neural Netw. **88**, 22–31 (2017)
16. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (2014)
17. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
18. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (2014)

19. Cai, D., He, X., Han, J.: Document clustering using locality preserving indexing. IEEE Trans. Knowl. Data Eng. **17**(12), 1624–1637 (2005)
20. Huang, P., Huang, Y., Wang, W., Wang, L.: Deep embedding network for clustering. In: 2014 22nd International Conference on Pattern Recognition (ICPR), pp. 1532–1537. IEEE (2014)
21. Papadimitriou, C.H., Steiglitz, K.: Combinatorial optimization: algorithms and complexity. Courier Corporation (1998)
22. Chen, W.-Y., Song, Y., Bai, H., Lin, C.-J., Chang, E.Y.: Parallel spectral clustering in distributed systems. IEEE Trans. Pattern Anal. Mach. Intell. **33**(3), 568–586 (2011)