Address Clustering for e-Commerce Applications

Vishal Kakkar
Flipkart Internet Private Limited
Bangalore, India
vishal.kakkar@flipkart.com

T. Ravindra Babu
Flipkart Internet Private Limited
Bangalore, India
ravindra.bt@flipkart.com

ABSTRACT

The customer addresses are important in e-Commerce for efficient shipment delivery. A predefined structure in addresses is not usually followed in developing countries. Further, some customer addresses are found to be noisy as they contain avoidable additional details such as directions to reach the place. In the presence of such challenges, understanding and equivalence mapping of the addresses becomes necessary for efficient shipment delivery as well as customer linking. We discuss the challenges with actual address data in Indian context. We propose effective methods for efficient large scale address clustering using conventional as well as deep learning approaches. We demonstrate effectiveness of these approaches through elaborate experimentation with real address dataset of an Indian e-commerce company. We further discuss effectiveness of such solution in fraud prediction models.

KEYWORDS

Geographical addresses, e-commerce, unsupervised learning, clustering, deep learning, cluster validation, machine learning

ACM Reference Format:
Vishal Kakkar and T. Ravindra Babu. 2018. Address Clustering for e-Commerce Applications. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR 2018 eCom).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Geographical addresses form basic references for delivering shipments ordered online with an e-Commerce organisation. These organisations focus on machine learning models for faster and accurate delivery of the orders. As the organisations expand to serve different business verticals such as perishable grocery, the need of delivering shipments within hours of the order placement becomes pivotal. The geographical addresses form basic building block for each of these services. Address is formally defined as *the one that specifies a location by reference to a thoroughfare or a landmark; or it specifies a point of postal delivery* [8]. Addresses and associated location information have multiple utilities such as linkage to legacy systems, dispatching aid for emergencies by governmental agencies, and many applications of geographical information systems [7].

In the online retail, customers at the time of their user registration specify one or more of their addresses. In reality, these addresses in developing countries contain following interesting challenges.

- Unlike places in Europe, US, Japan and North Korea, the written addresses in developing countries do not follow a prescribed structure [7].
- Every address does not readily have an attached geographical location.
- With multiple ethnic groups within a country like India, the names of the areas, groups of houses and the structure change from region to region.
- Members of same household write their common house address differently.
- The number of words of a typical address range from 2 to 20 words. In some cases, users in their eagerness to ensure that an ordered shipment properly reaches the address, they would add additional text like directions to reach a place, landmarks, times of their availability, phone numbers etc. There are many samples where an address would take around 50-150 words.
- Another interesting challenge is with postal codes known as postal index number or PIN code in India [19]. Due to limited literacy levels and rapid growth of localities within a city region, there is ambiguity about geographical zones corresponding to each PIN code. Thus, albeit PIN code forms unique reference a to broad locality, in practice, the number mentioned by the customers is not always accurate.

Such depiction of addresses pose challenges for a machine learning based automated solutions such as address classification [2], address clustering, fraud address identification or monkey typed address classification [3], partial or incomplete address classification, etc.

E-Commerce companies face a number of frauds such as reseller fraud, and fraudulent claims related to missing items or mis-shipments. The resellers are those customers who exploit online discounts or offers, and sell the items offline for profit. The e-Commerce companies limit the number of purchases made by a single customer in order to reach out to larger customer base. Under this constraint, resellers would vary their address patterns to register as a new user for making another purchase. The reseller fraud relates to an online fraud where some fraudulent customers buy items by making use of offers and discounts and selling them offline for profit. Such fraud reduces opportunity for genuine customers to make purchases. Machine learning models are built to identify of fraudulent transactions, such as, *reseller fraud* as discussed above, *missing item fraud* where a buyer claims that delivered package did not contain an ordered item, etc. The customer addresses written differently with fraud intent but belonging to the same customer form an important signal in such models. Similarly, identification

In: J. Degenhardt, G. Di Fabbrizio, S. Kallumadi, M. Kumar, Y.-C. Lin, A. Trotman, H. Zhao (eds.): Proceedings of the SIGIR 2018 eCom workshop, 12 July, 2018, Ann Arbor, Michigan, USA, published at http://ceur-ws.org

of those addresses of the customers belonging to same household but registered differently is another important application where similar addresses need identification.

In the presence of above challenges in our current work, we examine ways of clustering the same addresses that are written differently either as non-standard sequencing of words or as large set of additional words. An example for non-standard sequencing of words in addresses of same location but written differently is, *House No.xxx, 2nd Main, ISRO Layout, Bikasipura, Bangalore* vs. *House No.xxx, Bikasipura, ISRO Layout, 2nd Main, Bangalore.* Such addresses are very common in the current problem. Another important challenge is the massive address dataset that spans across the country, which needs to be clustered. Thus, in the presence of these challenges, the choice of clustering approach is important.

We examine multiple clustering schemes. After relative evaluation, we consider one base clustering scheme that requires single database pass to generate clusters. Its distinct advantage as non-iterative algorithm is ideally suited for mining large datasets. In terms of features, we consider two approaches. In Approach-1, we consider words directly as features. In Approach-2, we consider address word embeddings computed using huge address corpus. We provide interesting insights on word embedding in clustering addresses, in the presence of avoidable additional words such as directions to reach a place. The word embedding on address corpus is more suitable for the present application than using a generic pre-trained embeddings. We also examine affinity propagation approach to clustering using word embeddings. We carry out experimentation. We demonstrate effectiveness and efficiency of such approaches. We integrate following aspects in the paper.

- A discussion of efficient clustering approaches
- Clustering using text similarity
- Word embedding of addresses
- Clustering using word embedding
- Experiments on large datasets
- Cluster evaluation

We organise the paper in the following manner. Section 2 contains motivation for solving the proposed problem and a discussion on related work. The data challenges and preprocessing are discussed in Section 3. Section 4 contains a discussion on proposed solutions. The proposed algorithms are tested on a huge corpus of actual Indian addresses. The results are discussed in Section 5. The application usecases are presented in Section 6. A summary of contributions of the work and outline of future work are provided in Section 7.

2 MOTIVATION AND RELATED WORK

The clustering of addresses is essential to identify groups of users that have same address but written differently. It helps to identify groups of same or similar users for reaching out to them with possible business initiatives as well as to identify potential fraudsters. The variability in addresses arises due to inadvertent entries, non-existence or non-conformance of address standards, and the nature of users who provide avoidable additional details. Apart from these sources, the user entered addresses contain many challenges in terms of spell variants, abbreviations, monkey typed addresses containing jumbled alphanumeric characters [3], and incomplete or partial addresses and wrong PIN codes. These aspects motivate us to identify an effective and efficient clustering algorithm that is suitable for clustering large address dataset.

We did not come across any work related to clustering of customer or postal addresses in the literature. We discuss works on related topics. The objective of clustering is to separate patterns into groups such that the patterns within a group are similar with reference to a chosen criterion than to those patterns in the other groups [18]. In data mining, clustering helps to discover groups and identify interesting hidden patterns [10]. The clustering can be broadly categorised into partitional, hierarchical, density based and grid based [10, 13]. These works further divide the cluster approach taxonomy as agglomerative vs divisive, monothetic vs polythetic and in terms of feature usage as hard vs fuzzy, deterministic vs stochastic, and incremental vs non-incremental. Addresses that we consider contain alphanumeric terms. Clustering such addresses is related to text clustering. The text documents for clustering can broadly be classified into words, sentences, short-text messages, paragraphs and large documents [1]. The work further discusses approaches to text feature selection and feature extraction with their relative advantages and disadvantages, distance and phrase based clustering, online algorithms for text streams, and overview on semi-supervised text clustering. The addresses are distinct from conventional text documents, which usually have large vocabulary. Thus some of the text clustering approaches are not directly applicable.

Address could be considered as short text documents in terms of length. In clustering short text documents, exact keyword matching is not found to be sufficient and text representation is expanded by making use of larger related documents [20]. Some approaches to find similarity are by query expansion based on related documents, not with the conventional focus of creating new query for information retrieval but to achieve pair-wise comparisons between short text snippets [17]. The work is based on measuring semantic similarities of short text in terms of novel kernel functions. Since the objective of current work is to find similarly written addresses, we examine one approach where we use text words directly for their similarity.

For large datasets, the choice of clustering algorithm needs to be efficient. Ideally, such algorithms should be able to cluster a large dataset with a single or two passes of the dataset since iterative algorithms seek multiple passes through the dataset which are prohibitively expensive. In view of this, we consider leader clustering algorithm [18] which generates clusters through a single pass. Earlier studies [4] demonstrated that the prototype selection by the algorithm is better than k-medoids. We discuss the leader algorithm in more detail in Section 4. For a general discussion on clustering approaches we refer the readers to [10, 13].

As discussed previously, the addresses contain a number of spell variations, variable sequence of keywords and additional text that indicate description of reaching the place, phone numbers, etc. Thus a word based matching can be challenging. For example, word based matching of words like *'apartment'* and *'apartmnts'* would be successful depending on threshold of number of characters, whereas the words *'apartment'* and *'apt.'* do not match although they represent the same. In view of this, we also consider word embeddings as bag of words for generating equivalent sets of words [15]. We shall discuss its usage and method of combining multiple keywords in address in Section 4.

Another important related area is evaluating the clusters. If the objective of clustering is data reduction through prototype selection, the clustering can be evaluated by labelling large enough sample of patterns and evaluate the clustering based prototype selection through conventional supervised learning metrics [5]. But in the present problem, there are a large set of same addresses that are written differently. Thus the number of clusters is very large which makes supervised learning based evaluation unwieldy. We discuss some approaches of evaluating the clusters [6, 10, 12, 13, 16]. A clustering structure is said to be valid if the clusters are not chance occurring or not occurred due to the choice of clustering algorithm. Three criteria to evaluate cluster validity are known as external, internal and relative. The external criteria refers to comparison of 'identified structure', as obtained through clustering, with an 'a priori structure'. The internal criteria examines whether identified structure is inherently appropriate for the data. The relative criteria compares two identified structures and computes their relative merit. The validation approaches to external criteria include statistical tests and Monte Carlo techniques to test whether the given dataset is random. We list out some statistics [10] such as rand statistic, Jaccard coefficient, Folkes and Mellows index, Huberts Γ statistic, and normalised Γ statistic. The Cophenetic Correlation Coefficient and Monte Carlo techniques are used for internal criteria validation. For relative criteria validity, Dunn index based statistics, and Davies-Bouldin index are used. Depending on nature of clustering algorithm, statistics for cluster validity differ. For detailed discussion on the statistics, we refer the reader to [10, 12]. Additionally we briefly discuss two frequently used statistics known as Silhoutte coefficient [16] and Calinski-Harabasz [6] index. The Silhoutte coefficient is bounded between -1 and 1, and it is based on mean distance of the objects to the objects in the same cluster and mean distance to objects in the nearest cluster. The score around zero indicates overlapping clusters. The score is higher for dense and well separated clusters. The disadvantages are that is has higher computational complexity and is higher for convex clusters. The Calinski-Harabasz index is based on within cluster and between cluster covariances. The score has higher when clusters are dense and well separated but favours convex clusters. In the present work, we consider silhoutte and Calinski-Harabasz scores in the evaluation of address clusters.

3 DATA DESCRIPTION AND PREPROCESSING

We consider address data of over a million of Indian addresses. It is observed that rural addresses are short. The urban addresses are usually long. Table 1 contains average length of addresses across India at five representative states in the East, West, North, South and Central part of India. It can be observed from the table that average number of words per address is about 9, across the country. But the maximum number of words vary from 50 to 166 which is much larger than average number of words per address over any region. Such large number of words indicates avoidable additional details. As discussed earlier, examples of such details are the text on directions to reach a location, times of customer availability, etc. They pose challenges to the address clustering algorithms. Table 2 contains similar statistics for cities across India. Table 3 contains samples of long addresses and a compact form of the same as registered by another customer. House or apartment number is intentionally replaced by '***' in the tables. A sample of some long and some spurious addresses written by customers is provided in Table 4.

Table 1: Address length statistics as number of words in representative states across different regions within India

Location	Max	Mean	Median	Mode	Std Dev
India	166	9	8	6	5
Southern India (Karnataka)	90	11	10	9	5
Northern India (Punjab)	58	8	8	8	4
Eastern India (West Bengal)	65	9	8	7	4
Western India (Gujarat)	92	9	9	8	4
Central India (Madhya Pradesh)	53	9	8	7	4

Table 2: Address length statistics across a few cities in India

Location	Max	Mean	Median	Mode	Std Dev
Hyderabad	80	10	9.0	8	4.83
Bangalore	80	12	11	10	5.05
Mumbai	87	13	12	11	4.84
Pune	89	11	11	10	4.85
Surat	92	10	9	9	4.06
Ahmedabad	87	10	10	8	4.18
Chennai	110	10	10	8	4.44
Kolkata	93	9	8	7	4.44
Delhi	90	10	9	8	4.19

Table 5 contain samples of similar addresses that are written differently. Please note that the occurrences of abbreviations such as 'Nr' for 'near', mix of lower and upper cases, inconsistent delimiter usage, etc.

Additionally, we find that the addresses contain a number of special characters that must have been crept in due to inadvertent entries, storage and retrieval. We restrict preprocessing to converting all the addresses to lower case, and retaining only alphanumeric characters by replacing the rest by spaces.

4 PROPOSED APPROACHES

The discussion on address data in Section 3 provides insights on the data challenges. We need to cluster the same and similar addresses. In this context, we briefly discuss about Indian address system.

Table 3: Sample of Short and Long Addresses of Same Location

Sl. No.	Address
1	***, Titanium City Centre, 100 ft Anandnagar Road Next to’
	***, Titanium City Centre, 100 ft Anandnagar Road Next to Sachin Towers,Satellite, Close to Prahaladnagar, Satellite, Ahmedabad
2	The Grand Mall, Opp. SBI Zonal Office, S.M.Road, Ambawadi
	***, The Grand Mall, Opposite State Bank Zonal Office, S M Road„ Ambavadi, Surendra Mangaldas Rd, H Colony, Ambawadi, Ahmedabad, Gujarat 380015 Opposite State Bank Zonal Office

Table 4: Sample of Long and Spurious Addresses

Sl. No.	Address
1	House of ***(***); *** Bipin Ganguly road; (Near Dum Dum road); Area–Lal Bagan (Near PUKUR) which comes after Seth Bagan (ask near for Lal Bagan or seth bagan in Dum Dum road rickshaw stand near metro station)or After coming to 240 Bipin Ganguly road.come to NEW PARK (same address); South Dum Dum
2	Home The Club Chairman’s Message About Montana Vista Location Floor Plans Clubbing Experience Facilities Take a virtual tour Contact Us The Conclave Family The Conclave Club Eco Vista Club Verde Vista Club Rio Vista Club Montana Vista - By Conclave Contact Us Address Club Montana Vista The Uttorayon Township Matigara, NH-31 Siliguri - 734010 West Bengal "
3	Flat No. - ***, * Floor, M S Apartments, 190, Mondal Ganthi, Kaikhali, VIP Road, Kolkata 700052, West Bengal. (near Bhand Company) Landmark: Entry from either Kaikhali Market, VIP Road or Kaikhali, Jessore Road, Take Turn at Godrej Factory, Come Up-to Air India Quarters & Kendriya Bihar Back-Side Gate, at Left-Hand Side Yellow-Color 5-Storied Building, at 3rd Floor. ’

The postal index number or PIN code [11] is associated with each Indian address. India is divided into nine PINCODE zones [11, 19], eight of them are used for civilian use. The number of addresses across PINCODES is not uniform. Further, since the dataset considered for study is based on customers of an Indian e-commerce company, Flipkart, the set is unlikely to contain entire address database of the country. However they could be proportionately representative in view of Flipkart’s e-commerce penetration in the country. The number of addresses per PINCODE ranges from few hundreds to hundreds of thousands. We experiment clustering algorithms on each PINCODE as the addresses are similar within a PINCODE zone.

Table 5: Similar Addresses Written Differently

Sl. No.	Address
1	***, Rosewood Estate, Near Prernatirth Jain Temple, Jodhpur Gam, Satellite, Ahmedabad ’
2	***, Rosewood Estate, Jodhpur Gam, Satellite, Ahmedabad Near Prernatirth Derasar’
3	***, rosewood estate near prena tirth dersar jodhpur gam road sattelite ahmedabad ’
4	***, Rosewood Estate, Opp Prerna Tirth Jain Derasar, Jodhpur Gam, Satellite Ahmedabad-380015 (Gujarat) Prerna Tirth Jain Derasar’
5	***, Rosewood Estate, Nr Prernatirth Derasar, Jodhpur Gam, Satellite, Ahmedabad ’
6	***, Rosewood Estate Near Prenatirth Derasar Satellite Ahmedabad ’
7	***, Rosewood Estate,Near Prernatirth Derasar, Jodhpur Cross Roads,Satellite,Ahmedabad Prernatirth Derasar’

We consider a number of clustering algorithms. The objective is to obtain clustering of similar addresses. The data under consideration is text, and the number of candidate addresses is large. We look for representative patterns. These considerations eliminate popular k-means algorithm as number of clusters can not be predefined, and secondly the centroid is unlikely to be a pattern in the original dataset. As we cannot predetermine the number of clusters, and also based on earlier study comparing k-medoids to leader for prototype selection [4], k-medoids and its variants like PAM, CLARA and, CLARANS [14] are not considered. Based on these aspects, and its ability to identify effective prototypes [5], we consider leader clustering algorithm. A description of the algorithm is presented in Section 4.1.

In the Leader algorithm, we have the flexibility of defining the similarity threshold and the number of clusters is determined by that threshold. Further, prototypes generated by the leader algorithm will be the patterns from the original dataset. Another advantage of the algorithm is that it requires single pass through the large database to generate clusters. For pattern similarity we consider conventional text similarity approach as well as word embedding vectors. We discuss the algorithms, ways of combining words, and algorithm complexity in this section. We discuss leader algorithm in Section 4.1, affinity propagation in Section 4.2, leader with conventional text comparison in Section 4.3 and leader with word embeddings in Section 4.4.

4.1 Leader Clustering

The leader algorithm is presented in Algorithm 1. It is based on a predefined *similarity threshold*, ξ. Initially, a random pattern among the input patterns is selected as leader. Subsequently, similarity of every other pattern is compared with that of selected leaders. If the similarity of new pattern is more than the threshold, the corresponding pattern falls in the cluster with the initial leader. Otherwise, the pattern is identified as a new leader. The computation of leaders is continued till all the patterns are considered. We can tune the similarity threshold according to the task in hand. The number of leaders is directly proportional to the selected threshold. Therefore, smaller the similarity threshold, smaller will be the number of clusters, but it may affect intra-cluster distance of the clustering and high the threshold, more will be number of clusters, but it may affect inter-cluster distance of clustering.

Algorithm 1 Leader Clustering

```
Input: patterns: $P[1, ..., n]$ , similarity threshold: $\xi$
Output: $ldrpat[1, ..., k]$ , leaders which are cluster representatives
$ldrpat[1] \leftarrow P[1]$
$noofleaders \leftarrow 1$
for i=1 to n do
  for j=1 to $noofleaders$ do
    if $Similarity(P[i], ldrpat[j]) under \xi$ then
      $noofleaders \leftarrow noofleaders + 1$
      $ldrpat[noofleaders] = P[i]$
      break
    else
      $ldrpat[j] = P[i]$
    end if
  end for
end for
```

4.2 Affinity Propagation

Similar flexibility of having a representative pattern or prototype from the original dataset and avoid pre-specifying number of clusters a priori can be obtained by the Affinity propagation algorithm [9] as well. Affinity propagation creates clusters by exchanging messages between pairs of samples until a set of exemplars emerges, with each exemplar corresponding to a cluster. The Affinity Propagation algorithm takes as input a real numbers(k, k) for each data point k, referred to as a preference. Data points with large values for s(k,k) are more likely to be exemplars. If we don't have information about the preferences among the data points, the algorithm will treat each data point as potential exemplar.

The messages sent between pairs represent the suitability for one sample to be the exemplar of the other, which is updated in response to the values from other pairs. This updating happens iteratively until convergence, at which point the final exemplars are chosen, and hence the final clustering is given. Affinity Propagation does not require the number of clusters to be suggested a priori or estimated before running the algorithm and chooses the number of clusters based on the data provided. The main drawback of Affinity Propagation is its complexity. The algorithm has a time complexity of the order $O(n^2 * d * T)$, where n is the number of samples, d is maximum number of words in address and T is the number of iterations until convergence. Therefore, scalability is the concern for this algorithm.

We discuss following two address clustering approaches that uses Leader as base clustering algorithm.

4.3 Leader Clustering with Edit Distance

In case of address data, we need to find similarity between the addresses. Since the objective is to group similar addresses, the addresses are expected to have terms that are similar but variant in position, spelling and number of terms. In view of this, we consider actual term matching. We use edit distance with appropriate threshold.

Algorithm 2 Similarity

```
Input: patterns: $p1, p2$, similarity threshold: $\xi$, edit distance threshold: $\epsilon$
count = 0
for word1 in p1 do
  for word2 in p2 do
    if $editdistance(word1, word2) \leq \epsilon$ then
      $count \leftarrow count + 1$
    end if
  end for
end for
if $count \geq \xi * min(no\_of\_words\_p1, no\_of\_words\_p2)$ then
  return YES
else
  return NO
end if
```

Leader clustering algorithm is provided in Algorithm 1. The proposed approach has two parameters, such as, ξ for similarity threshold and ϵ for edit distance threshold in terms of number of characters. The similarity algorithm used in the algorithm is described in Algorithm 2. The worst case complexity of Algorithm 1 is $O(n^2)$. The worst case complexity of Algorithm 2 is $O(d^2 * m^2)$, where d is the maximum number of words in any pattern p and m is the maximum length of any word w in any pattern p. So, overall worst case complexity of the algorithm is $O((n * d * m)^2)$.

The clusters are represented by leaders which are essentially one of the patterns in the given dataset. Leader should be contrasted against a centroid which in most cases is not a pattern in the given dataset. The number of clusters identified depends on the similarity threshold ξ. The value of ξ is empirically chosen and is data dependent. However, it should be noted that the leaders are order dependent.

4.4 Leader Clustering with Word Embeddings

The clustering of addresses with appropriate edit distance provides good homogeneous clusters. But when the addresses contain additional words, beyond a certain *threshold*, such addresses form a different clusters. At this stage, it is educative to examine alternate approaches to group those addresses that have additional words which are reasonably rare but still belong to the same location. Such additional words include such rare descriptions like directions to reach a place, as we discussed in Section 3. Secondly, it is interesting to explore ways to bring all the vocabulary that are semantically similar but syntactically variant. This motivated us to examine word embeddings based clustering.

In this approach, instead of using edit distance based method, we will use word embeddings of all the addresses. Each word is embedded into vector using the continuous bag of words (CBOW) algorithm [15] by training exclusively on a large corpus of addresses by considering only those words that have occurred at least τ times, where τ=100 in the present case. We considered a window length of 3 and embedding size of 200. We combine individual vectors of each of these words in an address through averaging. We term our entire approach as *add2vec* (address to vector). To compute similarity between two addresses, we compute the cosine similarity between two address patterns.

Algorithm 3 *add2vec* Cosine similarity

```
Input: patterns: $p1, p2$, distance threshold: $\xi$
vec1 ← 0, vec2 ← 0
for word1 in p1 do
    vec1 ← vec1 + word2vec(word1)
end for
for word2 in p2 do
    vec2 ← vec2 + word2vec(word2)
end for
if $cosine\_similarity(vec1, vec2) \geq \xi$ then
    return YES
else
    return NO
end if
```

As we notice in Algorithm 2, edit distance based similarity is polynomially dependent on the number of words as well as the length of words in the pattern. Whereas, *add2vec* based cosine similarity linearly depends on the number of words in the pattern and constant vector length l. Therefore, the complexity of *add2vec* similarity algorithm is $O(2d)$, where d is the maximum number of words in any pattern p. So, overall, the worst case complexity of the algorithm is $O(n^2 * d)$, whereas worst case complexity of Affinity propagation algorithm with add2vec based cosine similarity is $O(n^2 * d * T)$.

The first advantage of using *add2vec* approach is that it is scalable and requires less computation time, given the address word embeddings. Secondly *word2vec* can capture spell variants which can go beyond the threshold chosen for term based comparison as discussed in Section 4.3. Also, it can handle additional uncommon words such as directions to reach a place.

Table 6: Algorithm Complexity

Algorithm	Complexity
Affinity Prorogation	$O(n^2 * d * T)$
Leader (edit distance)	$O((n * d * m)^2)$
Leader (add2vec)	$O(n^2 * d)$

Table 6 provides algorithm complexity of the three algorithms, affinity propagation, leader with edit distance and leader with word embeddings.

We carry out elaborate experimentation considering a large address corpus. We discuss the experimental considerations and the results in Section 5.

5 EXPERIMENTATION AND RESULTS

We carried out elaborate experimentation to compare and validate results generated by address clustering algorithms. We used both human evaluation and commonly used metrics to compute cluster validity.

In the setting of address clustering, the actual number of clusters in the dataset is not known, a priori. In the context of e-Commerce customers, it is unlikely that one would arrive at clusters of large sizes except in case of fraud scenario such as resellers or large apartment complexes. The reseller fraud modelling is one of the applications of the current exercise of address clustering. In the experiments as shown in Figure 2, maximum cluster size is of the order of about 500.

Let us consider a case of validating a clustering algorithm by making use of labeled patterns [5]. Consider a large multi-class labeled patterns with each class containing hundreds of patterns. With the objective of data reduction, we generate representative patterns or prototypes using a clustering algorithm with appropriate corresponding hyper-parameters. The prototypes form proper subset of original class-wise pattern set. The clustering algorithm can be validated for its performance of identifying appropriate prototypes. It is done by classifying the test patterns by considering prototypes alone. However, since in the current scenario, the number of clusters is large and cluster sizes are small, the approach is not applicable.

In Section 5.1, we discuss two metrics, known as, silhouette coefficient and Calinski-Harabasz index. The alternate approach to metrics based validation is by making use of human experts validating whether cluster members are homogeneous. The experimentation, results and insights are presented in Section 5.2.

5.1 Cluster Validity Metrics

We use following two metrics for demonstrating the goodness of the clustering approaches.

Silhouette Coefficient:

The Silhouette Coefficient score [16], is defined for each sample and it is composed of two scores internally as given below. Consider a pattern i, that is assigned to Cluster C. Let a(i) be average dissimilarity of i to all other patterns of C. Consider any other Cluster D. If d(i,D) is average dissimilarity of i to all patterns of D, b(i) is defined as minimum of average dissimilarity across all other clusters of the dataset.

With the above definitions, Silhouette Coefficient score for pattern i is defined as:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{1}$$

The average Silhouette Coefficient score is considered as a metric. The score is lie in the range of -1 and +1, where -1 indicates the incorrect clustering and +1 indicates the highly dense clusters.

Calinski-Harabasz index: For k clusters, the Calinski-Harabaz score [6], also known as Variation Ratio Criterion (VRC), is defined as average Between Group Sum of Squares (WGSS) to average

Within Group Sum of Squares (WGSS). We follow the same notation as used in [6].

Consider that there are k clusters. Given the mean of squared distances of between all samples, $\overline{d}^2$ and mean of squared distance within a group, g, $\overline{d_g}^2$, the weighted mean of the different between overall and within group mean squared differences is given by,

$$A_k = \frac{1}{n-k}\sum_{p=1}^{k}(n_p - p)(\overline{d}^2 - \overline{d_p}^2) \tag{2}$$

The between group sum of squares is given by,

$$BGSS = \frac{1}{2}((k-1)\overline{d}^2 + (n-k)A_k) \tag{3}$$

The within group sum of squares is given by,

$$WGSS = \frac{1}{2}\sum_{p=1}^{k}(n_p - p)\overline{d_p}^2 \tag{4}$$

The Calinski-Harabasz score is given by,

$$s = \frac{BGSS/k - 1}{WGSS/n - k} \tag{5}$$

The score is higher for dense and well separated clusters.

5.2 Results

We generate word embeddings by training a large corpus of addresses of over a million of customer addresses across India by considering those words that have minimum support of 100. The approach identifies all spell variants that are close by in cosine similarity sense. In Table 7, we present spelling variants captured by Clustering with edit distance and Clustering with word2vec algorithms. The spell variants that are shown against embeddings are in addition to those identified using edit distance.

Table 7: Spell Variants of *Apartment* and *College* Captured by Edit-Distance and Address Embeddings

Algorithm	Variants
Edit-dist	aparmnt, aparent, aparmtment, aparetment, apparment, aparment, apprmnts, aparmnts, apartemnets
Embeddings	appartment, appt, apt, apartments, apparment, aprtment, appartement, appts, appartments
Edit-dist	collage, colloge, coolege, cottege, callage, coolage, collega, callege
Embeddings	collage, collge, colleage, clg, colege, colleg, colg, cllg

We aim to cluster similar addresses. Practically, the addresses are similar within a given geographical locality. The locality is captured well by PIN code as discussed in Section 4. For the exercises, we ignore error in PIN codes as mentioned by the customers. This usually forms a small percentage of the addresses. Further, their presence would only lead to singleton or clusters of very small size. Even within a PIN code the number of addresses reaches a hundreds of thousand addresses. Thus we restrict clustering to a given PIN code. We conducted PIN code-wise experiments across the country. We carry out experiments for choosing appropriate thresholds for clustering. Figure 1 contains number of clusters per PIN code using *add2vec*. We consider similarity as threshold as against conventional measure of dissimilarity for clustering. As similarity increases, number of clusters increases.

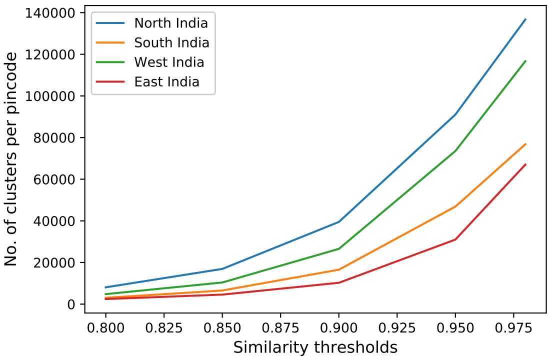

Figure 1: PIN Code wise Clusters Across India for different Thresholds

We repeat this experiment for different States within India for various similarity thresholds. We further study the number of addresses per cluster for all the states. For illustration, we present the results for one state of Punjab in India. Figure 2 contains number of cluster members for each cluster for thresholds, 0.85(top-left),0.90(top-right),0.95(bottom-left),0.98(bottom-right). Following observations can be made from these plots.

- No. of singleton clusters increases with increasing similarity threshold
- No. of cluster members is of the order of $\{2^7$ to $2^{12}\}$ for 0.85, $\{2^{6.7}$ to $2^{11}\}$ for 0.90, $\{2^6$ to $2^9\}$ for 0.95 and $\{2^5$ to $2^9\}$
- No. of cluster members per cluster is near optimal (change-over point) for the score of 0.95

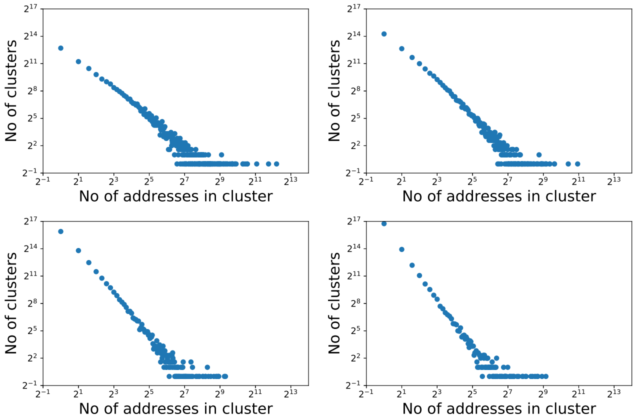

Figure 2: No. of addresses per cluster for thresholds 0.85 (top-left), 0.90(top-right), 0.95(bottom-left),0.98(bottom-right)

The cluster metrics for each of the above thresholds is placed in Table 8. It can be noted from the table the metrics increase with increasing similarity value till 0.95 and reduce subsequently.

Table 8: Cluster Metrics for Punjab, India for different thresholds

Similarity Threshold	Silhoutte Score	Calinsky Harbaz index
0.8	0.56	69.52
0.85	0.56	76.41
0.9	0.55	90.72
0.95	0.61	121.48
0.98	0.54	107.33

Based on the empirical evaluation and observations from the plots in Figures 1 and 2, and Table 8, we consider a threshold of 0.95 for leader with *add2vec*.

For leader clustering with edit distance, where we compare actual words between two addresses using edit distance, we use two thresholds. The first threshold, ϵ, is on the edit distance between two words and the second threshold, ξ, is on the number of words that could be different between two addresses so that they could be placed in the same clusters. The choice of ϵ depends on word length. We carried out experiments for different values of ϵ and ξ. The values for ϵ are chosen as {1,2} for word lengths of {≤4,≥5} respectively. Based on empirical evaluation, we consider a value of 0.95 for the threshold, ξ.

We present quality of clustering results of proposed Leader clustering algorithm with add2vec similarity, Leader clustering algorithm with edit distance similarity and Affinity propagation algorithm with *add2vec* similarity. To visualise homogeneity of clusters generated by each of these algorithms, we have chosen a sample of 1000 addresses randomly from a pincode. The results for similar address are shown in Tables 9, 10, 11 .

Table 9: Sample Clusters with Word Embeddings,*add2vec*

Sl. No.	Clusters with Affinity Propagation with *add2vec*
1	Tata consultancy services Electronic city phase 2 Electronic city
2	Tata Consultancy Services Electronic City Phase IINear XIME
3	TATA Consultancy ServicesNo. 42Think campusElectronic City Phase IIBangalore - 560100KarnatakaIndia TATA Consultancy Services
4	Tata consultancy servicesThink campus electronic city phase 2

It can be observed from the tables, that cluster quality with affinity propagation with *add2vec* similarity and leader with *add2vec* produce similar clusters. But computation complexity of affinity propagation is very high. We observe that the algorithm is not scalable for large data. In case of leader with edit distance, by virtue of choice of ξ and ϵ, the clusters contained a patterns that does not belong to the cluster. An example is shown in italics, in Table 10.

Table 10: Sample Clusters with Leader with edit distance

Sl. No.	Clusters with leader with *edit distance*
1	Tata consultancy services Electronic city phase 2 Electronic city
2	Tata consultancy servicesThink campus electronic city phase 2
3	*Gate number 5, wipro technologies electronic city, Electronics City*
4	Tata Consultancy Services Electronic City Phase IINear XIME
5	TATA Consultancy ServicesNo. 42Think campusElectronic City Phase IIBangalore - 560100KarnatakaIndia TATA Consultancy Services

Table 11: Sample Clusters with Affinity Propagation with Word Embeddings

Sl. No.	Clusters with Affinity Propagation with *embeddings*
1	Tata consultancy services Electronic city phase 2 Electronic city
2	Tata Consultancy Services Electronic City Phase IINear XIME
3	TATA Consultancy ServicesNo. 42Think campusElectronic City Phase IIBangalore - 560100KarnatakaIndia TATA Consultancy Services
4	TATA Consultancy Services Electronic City Phase II Electronic city phase 2

For the above samples, we computed cluster metrics and placed them in Table 12. The CPU time presented in seconds is based on random sample of 10,000 addresses. The CPU time on Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz with *Sklearn 0.19.1* implementation of python 2.7 is presented in Column-4.

Table 12: Experimental Values for Cluster Metrics

Algorithm	Sil. Coeff.	Cal. Har. index	CPU Time. (sec)
Affinity Prorogation	**0.378**	42.08	349
Leader (add2vec)	0.366	**53.51**	**31.3**
Leader (edit distance)	-0.327	5.232	1948

5.3 Clustering addresses with *add2vec*

We show the effectiveness of clustering approach with word embeddings in Table 13. Here we consider an address of leader or cluster prototype and compare its similarity as new words get added. Both the leader and final address with additional words are taken from actual address database of Flipkart. For the sake of brevity we show additional words in small bursts instead of considering them word by word. The table contain address and its similarity with the address in row-1. The corresponding similarity is with itself in row-1 and hence it is 1.0. Further in order to appreciate the changes from previous address, relatively new terms are shown in italics. Column-2 contains cosine similarity of the address to the leader-address (row-1). We note from the table, even after doubling of number of words from 16 of the leader to the address in the last row that contained 32 words, the similarity remains within the chosen threshold of 0.95. However this approach will have difficulty when the additional words are frequent words that appear in the address.

Table 13: Example of Effectiveness of Leader Clustering with word embeddings(*add2vec*)

Customer Address	Cosine Similarity
la renon healthcare pvt ltd 711 iscon elegance prahlad nagar cross road s g highway ahmedabad	1.0
la renon healthcare pvt ltd 711 iscon elegance prahlad nagar cross road s g highway ahmedabad *201 limited*	0.989
la renon healthcare pvt ltd 711 iscon elegance prahlad nagar cross road s g highway ahmedabad 201 limited *fax 91 office shapath india*	0.962
la renon healthcare pvt ltd 711 iscon elegance prahlad nagar cross road s g highway ahmedabad 201 limited fax 91 office shapath india *380015 p 5 1001*	0.966
a renon healthcare pvt ltd 711 iscon elegance prahlad nagar cross road s g highway ahmedabad 201 limited fax 91 office shapath india 380015 p 5 1001 *gujarat roads circle 1000 793046*	0.955

6 APPLICATIONS

We deployed this solution in conjunction with supervised machine learning model for classifying resellers and found significant gains in fraud prevention.

In general, the address clustering solution has multiple applications in the e-Commerce companies. A effective algorithm links all the users belonging to the same address but written differently either by deliberate attempt with fraud intent or due to inadvertence. This in turn has multiple applications such as (a) reaching out to different groups of users, (b) plan multiple shipment delivery initiatives, (c) machine learning models to detect frauds, etc. A pre-emptive action at the time of ordering can prevent fraudulent transactions. In summary, this forms an important signal for machine learning models in Trust and Safety domain of e-commerce transactions.

7 SUMMARY AND FUTURE WORK

We consider a practical problem of clustering addresses in places where the addresses do not follow a predefined structure. It is compounded by limited literacy of the customer which result in spell variations, merged words where space separator between two address components is missed, abbreviations, inadvertent separation of a joint word, etc. We consider a clustering algorithm that is suitable for mining large datasets, known as Leader. It uses single database scan for forming clusters. In terms of features, we consider text components of address directly and word embeddings of the entire address corpus independently for different approaches. With word embeddings, for a given address, we obtain vector representation by averaging each of the word embeddings. In addition to these algorithms, we also study the utility of affinity propagation algorithm and highlight its limitations. We present the algorithms, discuss relative advantage and disadvantages along with the elaborate experimental results. We carry out elaborate experiments and demonstrate that leader clustering with word embeddings, which we term as *add2vec* provides address clustering solution. The clustering approaches of leader with edit-distance could lead to outliers entering a cluster. And the affinity propagation algorithm is found to be non-scalable.

Another advantage of the clustering approach is that we obtain address prototypes for a clusters of similar address. The huge corpus of address of hundreds of thousands of addresses for each PIN code is reduced to smaller set of prototype vectors, which is a proper subset of original address dataset. This makes comparison of addresses, and insertion of a new address to the existing set of clusters as efficient operations, since we only consider the prototypes for these operations.

As future work, we propose to evaluate the use of tf-idf weighting of individual word embeddings to obtain representative address vector for more effective clusters and hierarchical clustering using the same approach with different valid thresholds for scaling to larger datasets. Also, we considered a window size of 3 for CBOW and embedding of 200. The effect of choice of hyper parameters on clustering performance is also considered for future work.

REFERENCES

[1] C.C. Aggarwal and C. Zhai. 2012. *A survey of text clustering algorithms,In Mining text data.* Springer, Boston, MA. 77–128 pages.

[2] T.R. Babu, A. Chatterjee, S. Khandeparker, A.V. Subhash, and S. Gupta. 2015. *Geographical address classification without using geolocation coordinates.* ACM.

[3] T.R. Babu and V. Kakkar. 2017. *Address Fraud: Monkey Typed Address Classification for e-Commerce Applications.* http://sigir-ecom.weebly.com/uploads/1/0/2/9/102947274/paper_21.pdf

[4] T. R. Babu and M. N. Murty. 2001. Comparison of genetic algorithm based prototype selection schemes. *Pattern Recognition* 34, 2 (2001), 523–525.

[5] T. R. Babu, M. N. Murty, and V. K. Agrawal. 2005. *On simultaneous selection of prototypes and features in large data.* Berlin, Heidelberg.

[6] T. Calinski and J. Harabasz. 1974. *A dendrite method for cluster analysis.*

[7] C.A. Davis and F.T.' Fonseca. 2007. *Assessing the certainty of locations produced by an address geocoding system.*

[8] FGDC Subcommittee for Culture and Demographic Data. 2001. *United States Thoroughfare, Landmark, and Postal Address Data Standard.* https://www.fgdc.gov/standards/projects/address-data/AddressDataStandardPart01

[9] B. J. Frey and D. Dueck. 2007. Clustering by passing messages between data points. *Science* 315, 5814 (2007), 972–976.

[10] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. 2001. On clustering validation techniques. *Journal of Intelligent Information Systems* 17, 2-3 (2001), 107–145.

[11] India Post [n. d.]. PIN Code. Retrieved May 1, 2018 from https://www.indiapost.gov.in/MBE/Pages/Content/Pincode.aspx

[12] A. K. Jain and R. C. Dubes. 1988. *Algorithms for clustering data.* Englewood Cliffs: Prentice Hall.

[13] Murty M.N. Jain A.K. and Flynn P.J. 1999. Data clustering: a review. *ACM computing surveys (CSUR)* 31, 3 (1999), 264–323.

[14] L. Kaufman and P. J. Rousseeuw. 2009. *Finding groups in data: an introduction to cluster analysis.* Vol. 344. John Wiley & Sons.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. (2013). https://arxiv.org/abs/1301.3781

[16] P. J. Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65.

[17] M Sahami and T. D. Heilman. 2006. *A web-based kernel function for measuring the similarity of short text snippets.*

[18] H Spath. 1982. *Cluster Analysis - Algorithms for data reduction and classification of objects.* Ellis Horwood Limited, Chichester.

[19] Universal Postal Union. 2018. Postal addressing systems in member countries. (2018). http://www.upu.int/en/activities/addressing/postal-addressing-systems-in-member-countries.html

[20] C. Zhai. 2008. *Statistical Language Models for Information Retrieval (Synthesis Lectures on Human Language Technologies).* Morgan & Claypool Publishers.