Address De-duplication using Iterative *k*-Core Graph Decomposition

Sriganesh Balamurugan*
srigab@amazon.com
Amazon
India

Vaasudev Narayanan*
vaasudev@amazon.com
Amazon
India

Saurabh Sohoney
sohoneys@amazon.com
Amazon
India

ABSTRACT

A de-duplicated and complete address catalog is essential for any application or business which needs to manage large volumes of address data such as delivery logistics, first-responder services and government databases. For catalog creation, address data is usually procured from disparate sources, which often vary in quality, coverage, and introduce duplicates or variations of the same physical address. *Address de-duplication* is therefore a crucial step for creating a clean and unified address catalog. De-duplication is even more challenging at a global scale, due to diversity in address writing styles, which might lack standardized addressing systems and can be multi-lingual. In this paper, we formulate address de-duplication as an unsupervised graph clustering problem and propose SANGAM, a novel adaptation of the *k*-core graph decomposition algorithm. We evaluate this solution on diverse geographic regions around the world. In comparison to existing methods, we observe improvements on the F-beta measure for three datasets. Our key contributions are: (1) formulating address de-duplication as a graph clustering problem, (2) proposing SANGAM, a robust and generic de-duplication approach, and (3) validating its effectiveness on diverse geographies across three continents - Americas, Africa and Europe. (4) Further, we deploy our solution and show the positive impact on *geocode learning*, an essential application of our solution.

CCS CONCEPTS

• **Information systems** → **Clustering**; **Clustering and classification**; • **Applied computing** → **Transportation**.

KEYWORDS

Address De-duplication, Clustering, Graph Decomposition

ACM Reference Format:
Sriganesh Balamurugan, Vaasudev Narayanan, and Saurabh Sohoney. 2024. Address De-duplication using Iterative *k*-Core Graph Decomposition. In *The 32nd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '24), October 29-November 1, 2024, Atlanta, GA, USA.* ACM, 4 pages. https://doi.org/10.1145/3678717.3691290

*Both authors contributed equally to this research.

SIGSPATIAL '24, October 29-November 1, 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1107-7/24/10
https://doi.org/10.1145/3678717.3691290

1 INTRODUCTION

Accurately *de-duplicating* address data and organizing it in a structure that corresponds to real-world places is crucial for applications managing large address directories. For instance, a clean, high-coverage, de-duplicated address catalog helps emergency medical, fire and police units to improve dispatch coordination and reduce operational costs. It also allows government agencies to plan infrastructure upgrades and provide targeted utility services. Logistics firms can leverage address catalogs to derive accurate demographic insights, optimize delivery routes and transport efficiency.

However, addresses captured in free-form text fields may not always follow consistent writing patterns, and significant variations for the same address can exist. For example, "*Cl Jardines de Toquio*"[1], "*Plaza Jardines de Tokio*", "*Placa de los Jardines de Toquio*" all refer to the same street in a particular city. This variability renders simple matching approaches infeasible. The problem is further exacerbated in countries which lack a widely accepted standardized addressing system. Additionally, every country presents its own unique challenges and with increasing globalization, a country-agnostic solution is highly desirable for easy deployment. In order to arrive at a comprehensive address directory, conflating multiple partial sources of information (address datasets) is a viable option. However, this introduces the challenge of duplication due to different writing formats and quality of the sources. The aforementioned difficulties therefore necessitate a generic, scalable and robust approach for address de-duplication.

We pose address de-duplication as a graph clustering problem [1, 12] where the vertices of the graph represent individual addresses and the presence of an edge implies that the addresses represent the same building in physical world. Graph clustering is a fundamental problem in machine learning with applications in diverse problem domains [6, 8–10]. For instance, in bio-informatics, graph clustering has been leveraged to identify groups of genes with related functions, potentially allowing identification of co-regulated genes [5]. Clustering also enables organization of large-scale face datasets [14] for biometric access control and forensic applications.

Therefore, we propose Synoptic Address Normalization for Global Address Management, a novel adaptation of the *k*-core graph decomposition algorithm [3]. SANGAM is scalable and robust, as evidenced by our results across geographies which differ in scale and address writing styles. Notably, the approach is generic and enables seamless extension to new geographies and problem settings.

Our main contributions are summarized as follows: (1) We formulate address de-duplication as a graph clustering problem and propose SANGAM, a novel adaptation of the *k*-core graph decomposition algorithm, and is simple, general and scalable, (2) We

[1]All addresses in this paper are fictional

demonstrate the effectiveness of SANGAM by performing a comprehensive set of experiments on three diverse datasets, consistently outperforming competing baseline methods on the F-beta measure by 13% on average, (3) We discuss a significant positive impact of SANGAM by improving *geocode learning* by over 200 basis points, an essential application of our solution.

2 PROPOSED METHOD

2.1 Problem Formulation

Let $\mathcal{B}$ be the set of real-world buildings and $\mathcal{A}$ the set of delivery addresses. An unknown many-to-one mapping $f : \mathcal{A} \rightarrow \mathcal{B}$ exists that maps each address to its true corresponding building. We have a similarity function $M : \mathcal{A} \times \mathcal{A} \rightarrow [0, 1]$ that produces possibly noisy similarity scores s_{ij} between addresses $a_i, a_j \in \mathcal{A}$, representing building-level similarity. Our objective is to discover the mapping f, thereby identifying addresses belonging to the same building. We formulate this as an unsupervised graph clustering problem where the vertices V are the addresses $\mathcal{A}$, and the edge set E consists of weighted edges (a_i, a_j, s_{ij}). Addresses mapping to the same building should be clustered together. Mathematically, each address a_i has a cluster label $c_i = f(a_i)$, where $c_i = c_j$ if a_i and a_j belong to the same building and $c_i \neq c_j$ otherwise. The number of clusters (buildings) is not known beforehand.

Algorithm 1 SANGAM for Address De-duplication

Input: $\mathcal{A}$: Set of addresses, $\mathcal{B}$: Set of real-world buildings, τ_1, τ_2, $k_{\min}$ (min. core value)
$V = \{a_i \mid a_i \in A\}$
$E = \{(a_i, a_j, s_{ij}) | (a_i, a_j) \in A \times A, s_{ij} = \text{similarity}(a_i, a_j); s_{ij} >= \tau_1\}$
Output: $f : \mathcal{A} \rightarrow \mathcal{B}$

while *convergence criteria not satisfied* **do**
  $G = (V, E)$
  $V' = \emptyset$
  k_{max} = Max. core number in G
  for $k = k_{\max}$ **to** $k_{\min}$ **do**
    $G_k = \{g_t \mid g_t \in \text{k-core}(G)\}$ {Extract set of disjoint k-core sub-graphs}
    for all $g_t \in G_k$ **do**
      n_t = get_representative_node(g_t)
      $V' = V' \cup n_t$
    end for
    $G = G - G_k$
  end for
  $E' = \{(n_i, n_j, s'_{ij}) \mid (n_i, n_j) \in V' \times V'; s'_{ij} = \text{average\_linkage}(n_i, n_j); s'_{ij} >= \tau_2\}$ {Link representative nodes}
  $G' = (V', E')$
  if max_core_value(G') $\leq k_{\min}$ **then**
    $f(a_i) = n_i$ {Addresses under the same representative node get the same cluster label}
    return f
  end if
  $V = V'$
  $E = E'$
end while

2.2 Methodology

We present SANGAM, our proposed, iterative k-core graph decomposition approach for address de-duplication. An overview of SANGAM is provided in Figure 1. SANGAM takes as input a set of addresses $\mathcal{A}$ and pairwise address similarity scores $E = \{(a_i, a_j, s_{ij} | s_{ij} >= \tau_1)\}$, which define weighted edges for an undirected graph $G = (V, E)$. The key steps are: (1) k-core extraction to decompose G into dense sub-graphs likely representing buildings, (2) Selective merging of sub-graphs based on inter-sub-graph similarity computation, and finalizing clusters by mapping addresses to merged sub-graphs.

2.2.1 k-Core Extraction. This stage corresponds to Steps 1-2 in Figure 1. The k-core of a graph is defined as a maximal sub-graph in which every node has degree $\geq k$; for example G1 in Step 2 in Fig. 1 is a 3-core sub-graph. We start by extracting the maximum k-core sub-graph $G_{k_{max}}$ from graph G using the highest core number k_{max} present in G. This extracts the most densely connected sub-graph. We remove all vertices in $G_{k_{max}}$ from G. By construction, what remains of G contains no sub-graphs having core number k_{max}. We repeat this process, progressively reducing k to extract the remaining lower core sub-graphs from G. At each k value, G may contain multiple disjoint k-core sub-graphs. We extract all such sub-graphs before proceeding to the next k value. The motivation is that k-core sub-graphs with large k (>50, say) have greater internal connectivity, hence fewer false matches. This allows us to set a high precision upfront. SANGAM also obviates the need for adaptive heuristics to choose k, making our approach simple and general. We terminate when we reach the pre-defined minimum core number k_{min}. The vertices at the end of this process are assigned as single node clusters. The k_{min} hyperparameter gives us fine control over the number of single node clusters that we create, hence controlling recall. The output of this stage is a disjoint set of k-core sub-graphs $\mathbb{G} = \{G_1, G_2, ...\}$ extracted over k-values. To further improve recall, we selectively merge these sub-graphs as explained next.

2.2.2 Selective sub-graph merging & finalizing clusters. This stage, corresponding to Steps 3-5 in Figure 1, aims to improve recall by selectively merging k-core sub-graphs extracted in the previous stage, while reducing the noise of pairwise matching. The merging is based on a similarity measure between sub-graphs. Specifically, we define the similarity between two sub-graphs G_i and G_j as the *average linkage* [13] between them, though more sophisticated graph similarity metrics could also be used. In Algorithm 1, each sub-graph G_i is abstracted to a representative node n_i. We construct a graph $G' = (V', E')$ where V' is the set of representative nodes n_i and $E' = \{(n_i, n_j, s'_{ij}) | (n_i, n_j) \in V' \times V'; s'_{ij} = \text{average\_linkage}(n_i, n_j)\}$, illustrated in Step 3 of Fig. 1. Edges below a stricter threshold τ_2 are pruned since each node represents multiple addresses. We check if G' still contains sub-graphs with core numbers above the threshold k_{min}. If yes, we apply another iteration of k-core extraction on G' for further recall improvements. If no dense sub-graphs remain, the algorithm halts. At this point, each representative node in the final G' represents a real-world building. k-core sub-graphs below k_{min} are broken into degenerate or single node clusters.

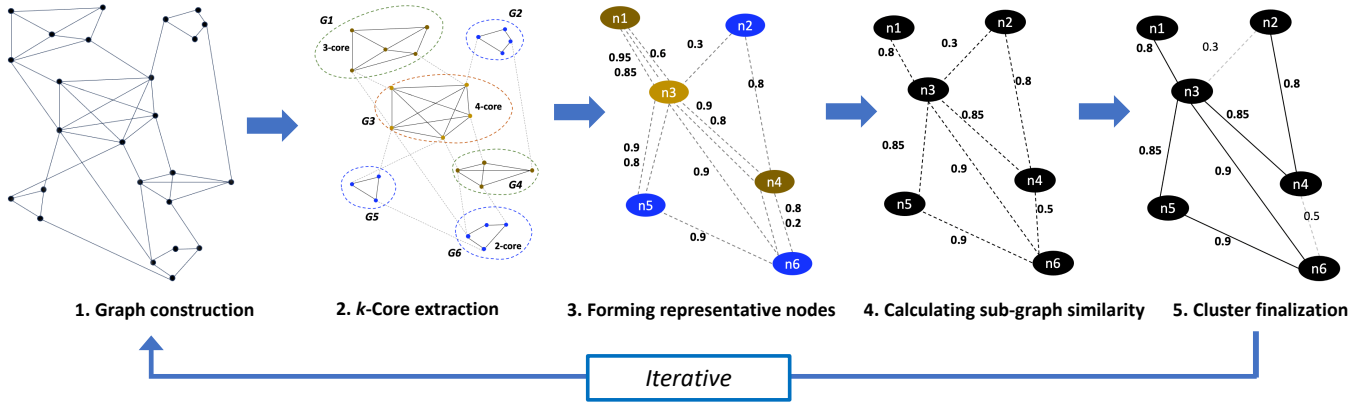

Figure 1: Overview of our proposed approach SANGAM

Table 1: Precision (Pr), Recall (Re) and F-beta score on three geographies – C1, C2 and C3. F-beta is the key metric and we report it for $\beta = \{1, 0.2\}$. $\beta = 1$ is the standard F1 measure. Best results for each geography are in bold, second best underlined.

	C1				C2				C3			
	Pr	Re	F-beta		Pr	Re	F-beta		Pr	Re	F-beta	
			$\beta = 1$	$\beta = 0.2$			$\beta = 1$	$\beta = 0.2$			$\beta = 1$	$\beta = 0.2$
Connected Components	56	88	68	81	90	8	15	82	98	73	<u>84</u>	<u>91</u>
Louvain Clustering	98	45	62	78	95	20	<u>33</u>	83	98	60	74	87
Token-based Heuristic	97	55	<u>70</u>	<u>82</u>	99	3	6	<u>85</u>	96	18	30	75
SANGAM (Ours)	98	68	**80**	**91**	95	54	**69**	**92**	99	83	**90**	**93**

3 EXPERIMENTS

3.1 Datasets

To demonstrate the effectiveness of our method, we applied it to address datasets from three distinct geographies across three continents: C1 (Europe), C2 (Americas), and C3 (Africa). Each region has its own unique characteristics - C1 and C2 have semi-structured address formats, while C3 is unstructured. All our datasets have a few million addresses, necessitating a scalable solution by design.

3.2 Evaluation Metrics

We evaluate our method on the following metrics to accurately assess the quality of the clustering – (1) pair-wise precision (Pr), (2) pair-wise recall (Re) and (3) F-beta score [2]. Here, lower precision indicates over-matched clusters (OM), where addresses from multiple buildings are erroneously grouped, while lower recall signals under-matched clusters (UM), where addresses from the same building are split across nodes. In different applications, over-matching and under-matching have different costs and F-beta is a convenient measure to quantify the trade-off. Notably, we calculate pairwise precision instead of clustering precision. Empirically, we have observed that pairwise precision approximates clustering precision but requires less effort to curate, enabling rapid experimentation. We compute these metrics on a stratified test set.

3.3 Baselines

We compare our method with the following baselines: (1) Connected Components (CC) [7] – CC forms clusters by taking a transitive closure of all edges in the graph. Specifically, if (a, b) and (a, c) are two edges, transitively (a, c) is also added to the edge set. All disjoint sub-graphs after performing the transitive closure operation on all edges are declared as clusters. Thresholds for CC are fine-tuned independently and it serves as a sensible lower bound. (2) Louvain Clustering [4] – Louvain is a community detection method in large graphs based on the concept of modularity [11], which is a measure of relative density of edges inside communities with respect to edges outside communities. (3) Token-based heuristic – we compare with a heuristic which groups addresses based on extracted address tokens such as building number, street address.

3.4 Results

Table 1 summarizes our results. As discussed in Sec. 3.2, F-beta is our key performance indicator, which appropriately weights precision and recall. We note that in many applications, precision is more important than recall, therefore in addition to $\beta = 1$ (equivalent to the standard F1 measure) we also report F-beta with $\beta = 0.2$. SANGAM outperforms all baselines across all geographies on both β values, achieving significant improvements over competing baseline methods – 900 bps in C1, 1900 bps in C2 and 1800 bps in C3. For C1, although CC has higher recall, it suffers from low precision, consequently a lower F-beta score. On the other hand, Louvain displays the opposite trend with high precision and low recall. We infer that it is able to extract dense edges but in the process is creating multiple duplicate clusters. The results clearly indicate the

usefulness of the individual components in SANGAM. The iterative k-core extraction procedure starting from the highest core value in the graph enables us to set a high precision upfront. Our sub-graph merging strategy then gives a lever for us to progressively increase recall, while maintaining acceptable precision. As a result, SANGAM strikes the right balance between precision and recall, creating dense, cohesive clusters with minimal duplication. This flexibility and simplicity is highly desirable, given the scale and the diversity of delivery logistics operations. We note that for fair comparison across baselines, we opted to maximize F-beta.

4 ABLATION STUDIES & ANALYSIS

4.1 Results across multiple iterations of k-core

We analyzed the performance across incremental iterations of SANGAM for the C2 dataset. In the first iteration, extracting the highest k-core sub-graphs provides high precision (99%), though with much lower recall (9%). As lower k-core sub-graphs are merged in subsequent iterations, recall steadily improves (9%→20%→54%) with a reasonable precision decline (99%→98%→95%). The consistent increase in F-beta measure (75%→80%→92%) further demonstrates SANGAM's robustness. By initially detecting tightly connected sub-graphs, then expanding to less dense sub-graphs, SANGAM balances precision and recall to effectively identify building clusters.

4.2 Analysis of core values

We hypothesize that densely connected sub-graphs are more likely to represent buildings, while sparse sub-graphs are less likely to represent buildings. To test this, we analyze SANGAM's behavior across k_{min} values for the C3 dataset over a reasonable k_{min} range (3→12). We observe a minimal F-beta change (92%→89%) and infer that SANGAM extracted most building clusters within higher k-cores, validating our hypothesis. It also demonstrates the algorithm's insensitivity to k_{min} hyperparameter. The stable F-beta across k_{min} also supports SANGAM's approach of incrementally extracting all k-core sub-graphs from the maximum core value.

4.3 Qualitative Analysis

Two types of noise exist from the pair-wise matcher output - false positives (incorrect matches) and false negatives (missed matches). The sub-graph aggregation process in SANGAM helps mitigate both. For example, in the 1^{st} iteration (see Step-2 in Fig. 1), edges connecting the sub-graphs G4 and G6 have similarity scores of 0.2 and 0.8, the latter being a False Positive. SANGAM by design does not give undue importance to the false positive edge. For example, "*Rua Carlos 941, Torre 1, Mooca City*", "*Rua Carlos 941, Bloco 3, Mooca City*" and "*Rua Carlos 941, Bl 4, Mooca City*" are addresses in a particular multi-building complex in geography C3. These correspond to three distinct buildings (1, 3 and 4), but are false positives, SANGAM is accurately able to create 3 building clusters. This is because we start the sub-graph extraction with the largest available core value (k), instead of relying on some heuristic for selecting the value of k, thus giving more importance to the denser sub-graphs over others. This significantly reduces the risk of introducing false positives. Similarly, SANGAM is adept in handling the issue of missing matches as well. For example, "*Plaza de los Jardines de Tokio 29*", "*Apt-4, 29, Plaza Jardines de Tokio*" and "*Placa dels Jardins de Toquio*" are different variations of a street address in geography C1 are not matched completely from the pair-wise matcher output (false negatives). However, the iterative approach in SANGAM aided by *average linkage* ensures that the similar sub-graphs (building clusters) with sufficient similarity are selectively merged together, leading to recall improvement without drop in precision.

5 REAL-WORLD APPLICATION

We analyze the positive impact of SANGAM on geocoding, the process of converting free-form address text into precise geocodes (latitude and longitude pairs). We compared the geocodes predicted by the existing model with those generated by our approach against the actual delivery locations. Our approach led to a 205 bps improvement in geocode accuracy. Additionally, we observed a significant reduction in anomalous geocodes by 54 bps. This enhancement not only ensures more accurate geocoding but also improves overall delivery precision.

6 CONCLUSION

In this paper, we pose address de-duplication as a graph clustering problem, and propose SANGAM, a novel adaptation of the k-core graph decomposition algorithm to solve the de-duplication problem. As evidenced by a comprehensive set of experiments across multiple diverse geographies, our algorithm is simple, general, scalable and robust, outperforming relevant competing baselines. We also discuss the benefits of our method on the important application of geocode learning. As part of future work, we plan to explore Graph Neural Networks to incorporate additional modalities as well as extend our work for address hierarchy creation.

REFERENCES

[1] Charu C Aggarwal and Haixun Wang. 2010. A survey of clustering algorithms for graph data. *Managing and mining graph data* (2010), 275–301.

[2] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*. Vol. 463. ACM press New York.

[3] Vladimir Batagelj and Matjaz Zaversnik. 2003. An o (m) algorithm for cores decomposition of networks. *arXiv preprint cs/0310049* (2003).

[4] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[5] Yizong Cheng, Chen Lu, and Nan Wang. 2013. Local k-core clustering for gene networks. In *2013 IEEE International Conference on Bioinformatics and Biomedicine*. IEEE, 9–15.

[6] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3-5 (2010), 75–174.

[7] John Hopcroft and Robert Tarjan. 1973. Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM* 16, 6 (1973), 372–378.

[8] Xue Jiao, Yonggang Chen, and Rui Dong. 2020. An unsupervised image segmentation method combining graph clustering and high-level feature representation. *Neurocomputing* 409 (2020), 83–92.

[9] P Liu, X Wang, CH Hu, and TH Hu. 2012. Bioinformatics analysis with graph-based clustering to detect gastric cancer-related pathways. *Genet Mol Res* 11, 3 (2012), 3497–3504.

[10] Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert E Tarjan. 2007. Clustering social networks. In *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 56–67.

[11] Mark EJ Newman. 2006. Modularity and community structure in networks. *Proceedings of the national academy of sciences* 103, 23 (2006), 8577–8582.

[12] Satu Elisa Schaeffer. 2007. Graph clustering. *Computer science review* 1, 1 (2007), 27–64.

[13] Hamid K Seifoddini. 1989. Single linkage versus average linkage clustering in machine cells formation applications. *Computers & Industrial Engineering* 16, 3 (1989), 419–426.

[14] Lei Yang, Dapeng Chen, Xiaohang Zhan, Rui Zhao, Chen Change Loy, and Dahua Lin. 2020. Learning to cluster faces via confidence and connectivity estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13369–13378.