Graph based Active Learning for Accurate Matching of Customer Address

Anonymous Author(s)
email

Abstract

We propose a novel adaptation of graph-based active learning for customer address matching or de-duplication, with the aim to determine if two customer addresses represent the same physical building or not. For Amazon Logistics, improving address resolution positively impacts multiple downstream systems such as geocoding, route planning and delivery time estimations, leading to an efficient and reliable delivery experience, both for customers as well as delivery agents. Our proposed approach jointly leverages free-form address text, past delivery information and fundamental concepts from graph theory to retrieve informative and diverse record pairs to label. Experiments on a manually curated dataset across top performing techniques demonstrate the efficacy of our proposed approach. We achieve 14.2% improvement in recall while preserving 95% precision in comparison to the existing model in production as baseline across two geographies. Our approach is *data-efficient* as it requires only 37.2% of the labeled data to achieve the same results as 100% labeled data. We also introduce *delivery point geocode learning* as a downstream application of address matching. Offline experiments on few hundred thousand shipments show that when the base model is augmented with active learnt record pairs, the precision improved by 10.2% and defects reduced by 12.1%. We plan to complete the deployment for India and UAE by Q3 2022.

1 Introduction

Entity matching (EM), also known as entity resolution (ER) [11], aims at identifying and linking different representations of the same real-world entities across databases. EM is a challenging task for real-world applications, particularly when entities are highly unstructured [15] and of low quality, for example, there is lack of completeness and consistency in their descriptions. Further, real-world EM tasks [11] have limited access to labeled data and require substantial labeling effort to learn accurate EM models. For Amazon Logistics (AMZL), customer addresses play an important role in delivery planning, as the address is the primary source of information provided by customers regarding their location. Customers provide their addresses in free-form text fields at their own discretion, which may or may not follow a fixed pattern. Address writing styles and patterns are idiosyncratic the same as hand writing or signatures. This leads to a lot of variation in similar addresses and their components (building, campus, road, landmark). Some countries such as India (IN), United Arab Emirates (UAE), Kingdom of Saudi Arabia (KSA) and Egypt (EGY) do not adhere to standard addressing system that customers widely accept and use, thus making addresses inherently unstructured. Customers often use references to neighbourhoods, landmarks or POI. For example, it is common for customers in IN to provide colloquial addresses that use landmarks and other points of interest (POI) to denote the place, for example, *HDFC Bank, Opp. Cyber Pearl, Madhapur, Hyderabad*. Other customers may provide more structured addresses that intend to indicate the same place but also conform to local

Submitted to Amazon Machine Learning Conference (AMLC 2022). Do not distribute.

postal standards, for example, *Plot No. 18, Hardik Crown Towers, Hdfc Bank, Madhapur, Hyderabad*. A neighborhood provided by a customer can also be known by other vernacular names or be a part of a larger neighborhood. For example, *Khalifa City B* and *Shakhbout* refer to the same sub-locality within the larger *Khalifa* neighbourhood of Abu Dhabi city in UAE. Customers use these synonyms interchangeably, making address even more challenging to comprehend.

Address resolution aims to de-duplicate a given query address against a set of candidate addresses in the database via pair-wise address matching. Creating a representative training set for pair-wise matching is challenging for customer addresses for multiple reasons — (1) Data distribution is heavily skewed towards negative pairs, i.e. no-match. (2) Based on an analysis carried out by our manual data curation team, the average handle time (AHT) for an annotator to label a customer address pair is very high; *three* times higher on average when compared to other EM tasks. Also, the rate of disagreement among annotators for labeling customer address pairs is very high; *four* times higher on average when compared to other EM tasks. (3) Across customer addresses, it is very common that component values are vernacular, redundant, noisy, missing, or misspelled, thus leading to unstructured data problems. (4) No appropriate pre-labeled data exists to bootstrap classifiers, nor rules to automatically label training data through weak supervision. Labeling a large volume of pairs for a variety of scenarios in EM does not scale, hence prior studies in EM has adopted active learning. Active Learning (AL) is an effective paradigm for efficiently acquiring data [8, 12, 21] from a pool of unlabeled records.

Our proposed approach jointly leverages free-form address text, geospatial properties (GPS points from past deliveries), graph theory (graph partitioning, graph cuts and transitivity) and active learning to sample informative and diverse record pairs to minimize cost of annotation. The most widely used approaches to sample records from an unlabeled pool for AL are based on uncertainty and diversity. Algorithms based on uncertainty may end up choosing uncertain yet uninformative repetitive data, while diversity-based methods may tend to select diverse yet easy examples for the model [13, 19]. Though hybrid AL frameworks that combine uncertainty and diversity overcomes the limitation of acquiring redundant samples, some studies [13, 14] found it to be ineffective for real-world EM tasks. Using graph signals for active learning has been effectively explored in prior literature [1, 17]. The positive impact observed from graph-based approaches for different tasks has inspired our work on graph active learning for customer address resolution. But due to the variety of challenges posed by customer addresses the existing techniques are insufficient and cannot be utilized in their current form for our problem space. In this paper, we address the above limitations through the following contributions — (1) We propose a novel adaptation of graph-based active learning to tackle a real-world problem of address resolution, particularly pair-wise matching. (2) We jointly leverage free-form address text, geospatial properties of addresses along with fundamental concepts from graph theory to retrieve informative record pairs. Our query strategy utilizes *disagreement* and *geospatial-diversity* to select record pairs to label in a data-efficient manner.(3) We also discuss and experimentally demonstrate how improving address matching positively impacts delivery point (DP) geocoding, a fundamental business problem that enables delivering packages to customers.

2 Background

This section introduces necessary background on concepts from graph theory and geospatial properties of customer addresses. An undirected and weighted graph G with no self-loops is a pair (V, E), where V is a set of vertices or nodes and E is a set of edges between the nodes, that is $E \subseteq \{\{u, v\} \mid u, v \in V \wedge u \neq v\}$. Each distinct text from customer address database is represented via a node and the edge between two nodes is determined based on a match or no-match prediction of a trained machine learning model. We use $n = |V|$ and $m = |E|$ to denote the number of nodes and edges.

Graph Partitioning: Graph partitioning [4] refers to a class of problems that deals with reducing a graph to multiple smaller graphs by partitioning its set of nodes into mutually exclusive groups. We use the Louvain algorithm [2], a graphical method to partition a graph based on network structure and edge relationships. Louvain is an unsupervised algorithm and consists of two important phases, *modularity optimization* and *community aggregation* [2]. In comparison to other graph partitioning

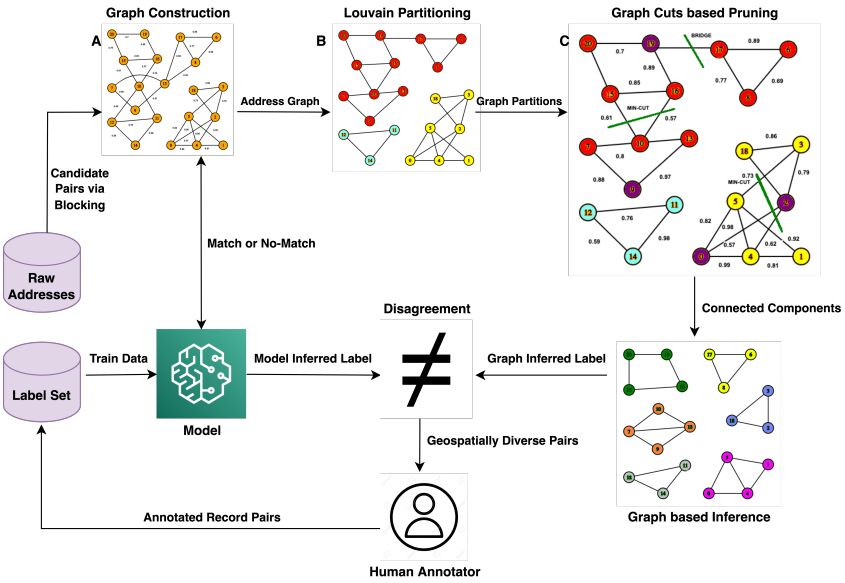

Figure 1: Our proposed workflow starts with raw addresses and apply blocking to get a pool of unlabeled pairs. The predictions of the base model on this pool are used to construct an address graph **A**. We apply Louvain on graph **A** to get partitions in **B**. We prune every partition via graph cuts (green line cutting edges show min-cut and bridge in graph **C**). Violet nodes in **C** are source and target nodes within connected component. The graph and model predictions in disagreement are human annotated after selecting geo-spatially diverse record pairs.

techniques, Louvain was preferred as it does not require us to input the number of communities, or the partition sizes before execution. Further, a single pass of Louvain is a *linear operation* in terms of the number of edges of the graph, thus allowing it to scale across millions of edges in a graph.

Graph Cuts: Graph cut techniques have been successfully applied to a number of real-world applications, for example, designing flow networks, computer vision, graphics and image processing problems [6]. We leverage *minimum cut* and *bridge* as graph cut techniques for the task of EM across customer addresses. For a pair of given nodes, source s and target t, the $s-t$ min-cut of a weighted graph is defined as the minimum sum of weights of (at least one) edges that when removed from the graph makes s and t disconnected. Bridge in an undirected graph G is an edge that when removed from a graph increases the number of connected components. In other words, if we remove an edge which is a bridges, the graph will no longer remain connected.

Geospatial Properties: For each node V in the graph G, two main modalities of information are available namely: the free-form address text and multiple GPS points based on successful past deliveries. GPS points are sometimes noisy as it depends on driver compliance. We use the GPS points associated with each address to learn a single delivery point (DP) to direct future deliveries. A brute force approach would be to compute the centroid of GPS points from past deliveries. Unfortunately, this can direct delivery associates to the middle of the street or to a different building. Centroids and medoids are prone to outliers, hence proving inaccurate in estimating delivery points [9]. We use density-based methods to accurately approximate a single delivery point from historical deliveries for each address via Kernel Density Estimation (KDE) [20]. KDE maximized delivery points are used to determine the geospatial proximity between a pair of graph nodes using haversine distance [7].

3 Methodology

Problem Statement: In our problem domain, a *match* represents an address pair belonging to the same physical building whereas *no-match* represents an address pair referring to different buildings. We consider a pool-based setting [1] for AL to effectively utilize the human annotators. A pool of

unlabeled record pairs $\mathcal{P}$ are generated via blocking. At each iteration of AL, it selects a batch of $\mathcal{K}$ instances and add them to the labeled corpus $\mathcal{L}$, removing them from $\mathcal{P}$. We perform AL for $\mathcal{T}$ iterations. Our task is to effectively sample $\mathcal{K}$ record pairs to be labeled by human annotators from $\mathcal{P}$, such that after re-training it on $\mathcal{L}$, the performance on an unseen test set is *maximized*. The details of our proposed method as demonstrated in Figure 1.

3.1 Proposed Approach

Blocking: Comparing every address in the database to every other address (Cartesian product) is not scalable. We typically decompose this problem into two steps: *blocking* and *matching*. Blocking filters obvious no-match cases (addresses that belong to a different district, state or postal code) from the Cartesian product to obtain a candidate set. We use ElasticSearch [10], with fastText embeddings [3] to index the addresses and then filter those addresses that are an obvious no-match. We retrieve *top-k* candidates for every customer addresses to get a pool of unlabeled candidate record pairs $\mathcal{P}$ and apply the trained classifier to determine pair-wise matching.

Base Model: This model serves traffic in production and is trained on a manually curated ground truth training data. It performs pair-wise matching of record pairs using the XGBoost [5] classifier. We use the same binary classification model $\mathcal{M}$ to initialize our active learning loop. The classifier $\mathcal{M}$ is used to get baseline model predictions for all unlabeled record pairs of the pool $\mathcal{P}$.

Graph Construction: Given a pool $\mathcal{P}$ of unlabeled address pairs retrieved via blocking and their corresponding pairwise predictions from base model $\mathcal{M}$, we construct an address graph G. Each node represents a distinct address. We add an edge to the graph for every matching record pair, while we skip the edge for every non-matching pair. The weights assigned to an edge between a pair of matched nodes is equal to the predicted probability score learnt by the base model. We construct this graph with the aim to retrieve graph signals and exploit them during the later steps of our proposed approach for querying the annotators and model re-training. We leverage *transitivity* of an address graph G to discover *false negatives* from base model predictions. For example, given three record pairs $[AB]$, $[BC]$ and $[AC]$ which have been predicted by the base learner as match, no-match and match respectively, we can infer using graph transitivity that $[BC]$ is also a matching pair, a likely false negative prediction of the base learner. However, given that the edges of the graph are derived from the predictions of the base model, and the base model is not always accurate, a wrongly predicted match edge can lead to a series of *false positive* record pairs. Therefore, we use Louvain based partitioning and graph cuts based pruning to find and remove likely false positive edges from the graph and obtain smaller connected components.

Louvain based Partitioning: In this phase, we partition the nodes of an address graph G. First, we compute all the connected components of G. Then for each connected component, we apply a single pass of Louvain algorithm [2] to separate the nodes into multiple mutually exclusive graph partitions. In addition to run-time efficiency, partitioning the graph reduces the number of false positive edges by $\sim 32\%$, thus making it efficient for graph cut algorithms to operate during the pruning step, which is essentially a polynomial operation in terms of the number of edges of the graph.

Graph Cuts based Pruning: In this phase, we prune the individual graph partitions to eliminate weak links and isolated components. We utilize graph cut techniques and geospatial properties of addresses to prune the *likely false positive* edges from the graph. We iterate over all the nodes of the graph and compute their haversine distance with all other nodes using location proximity as discussed in section 2. We then extract those node pairs $s - t$ from the graph for which the haversine distance is greater than a pre-defined threshold $\mathcal{N}$. This ensures that the pair of extracted addresses are likely to belong to two different physical buildings. To ensure minimal impact of GPS noise, we select only those nodes in the record pair to which substantial amount of deliveries have happened in the past. First, for each $s - t$ pair we check if there is a path between s and t nodes in the graph. In case we find a path, we calculate the min-cut of the graph considering the weights of the edges and remove min-cut edges from the graph. Second, we identify and remove bridge edges from the graph. In order

to avoid creating too many small components, we only remove the bridge edges connecting nodes that have at least three neighbours each.

Graph based Inference: After pruning, we assign a graph label to a pool of record pairs with the aim of obtaining more accurate graph signals so that it can help us identify incorrect predictions of the base model. In order to derive a graph label, we first compute all connected components in the pruned graph. For record pairs belonging to the same connected component, we assign a *match* graph label. If there is no path in the graph between record pairs, i.e. the nodes are not in the same connected component, we assign a *no-match* graph label.

Query Strategy: Record pairs whose graph label is different from the label predicted by the base model are termed as *informative*. While margin-based and committee-based query strategies aim to select instances for which the learner or a committee of models produces non-confident predictions, our query strategy uses the connected components of the pruned graph to pick instances that are most likely predicted incorrectly by the base model. These disagreements between the graph and the base learner hint towards nuanced matching patterns that are not yet covered by the base learner. This can occur under following two scenarios. First, if the record pair has been predicted as no-match by the base learner but due to graph transitivity the graph inferred label is match. Second, if the record pair has been predicted as a match by the base learner but the corresponding edge was removed via graph partitioning step. We select an equal number of likely false-negatives and likely false-positives to prevent skewness in the data distribution. To ensure *geospatial-diversity* among informative record pairs, we sample across a grid based on the Military Grid Reference System (MGRS) [16]. This allows us to capture nuanced patterns in customer addresses across different regions of the geography and also ensures that different types of delivery areas (high versus low address volume areas) are well represented. In each iteration, we select a batch of $\mathcal{K}$ informative pairs, label it and augment labeled data to curated training data on which the base model $\mathcal{M}$ was trained. The augmented train set is used to re-train the base model and is evaluated on the same unseen test data.

4 Experiments

We did extensive offline experimentation to develop, refine and validate our proposed framework. In this section, we describe the experiments and discuss the results for India and UAE geography. Our results across multiple pairwise-matching and DP geocoding metrics demonstrate the efficacy and effectiveness of our proposed approach.

Curated Ground Truth (CGT): We did stratified sampling of addresses to curate ground truth. This way of sampling is likely to cover all the linguistics, address writing styles, and abbreviations across the country. The selection also ensures to consider medium and low address volume districts, thus accounting for the varied density of addresses. We generate close to $10K$ unique address pairs for each of the two geographies (IN and UAE), which were then manually labeled by the annotators.

Baselines: We compare and evaluate the efficacy of our proposed approach with *six* other techniques. (1) *Base Model:* The details of this model are already discussed in section 3.1. (2) *Random Sampling:* We randomly sample address pairs from the available pool of record pairs. (3) *Margin Sampling:* It follows uncertainty measure to sample record pairs. For binary classification, the difference between the match class score and no-match class score is computed. Lower the difference value, higher the uncertainty. (4) *Diversity Sampling:* We use MGRS to cluster the record pairs [16]. From every cluster, we sample the top uncertain record pairs. (5) *Hybrid Sampling:* A more generalized version of the previous baseline, it samples records pairs by balancing on uncertainty and diversity. (6) *Weak Supervision:* It uses Snorkel [18] framework to generate large volume of weak labels by modeling the output of labeling functions via a generative model. The labeling functions are designed using our domain knowledge and geospatial properties of addresses.

Evaluation Metrics and Results: We split the CGT data in 70-10-20 for training, validation and testing. The validation set is used only to tune the hyperparameters and the test set is held out during both training and validation. All the models were evaluated on same test dataset. To align with

Table 1: Performance of various models across multiple pair-wise matching metrics for IN and UAE

Model	Accuracy (%)		R@95 (%)		PR-AUC	
	IN	UAE	IN	UAE	IN	UAE
Base Model	80.49	85.23	23.94	72.32	0.925	0.959
Random Sampling	81.05	86.15	24.51	74.11	0.926	0.961
Margin Sampling	82.97	87.49	25.53	75.54	0.927	0.963
Diversity Sampling	83.34	87.96	26.04	76.27	0.931	0.965
Hybrid Sampling	83.78	88.13	26.97	76.86	0.931	0.967
Weak Supervision	83.21	86.88	27.72	77.04	0.932	0.966
Our Approach	**86.12**	**90.54**	**31.68**	**81.11**	**0.942**	**0.974**

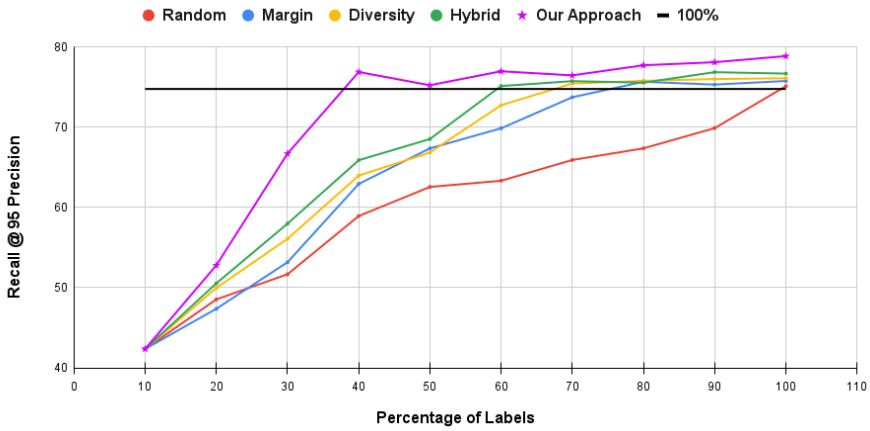

Figure 2: Performance of different sampling strategies with the labeled data for UAE

downstream application, a high precision (atleast 95% precision of match class) pairwise-matching model is required. We evaluate the pair-wise address matching model across three metrics namely — (1) Overall pair-wise accuracy (Accuracy), (2) Recall at 95% Precision (R@95P), and (3) Precision-Recall area-under-the-curve (PR-AUC). The R@95P and PR-AUC numbers are corresponding to the match class (pair representing same physical building) to align the performance of the model with the business objectives. Note that for all the AL baselines, we start with CGT training data as the initial labeled set. For all the sampling strategies, the record pairs retrieved by AL are labeled by the data annotation team. In the first experiment, the base model is trained on CGT train set. For every sampling strategy, we augment the obtained label record pairs to CGT train set and evaluate the performance on the same CGT test set. From Table 1, it can be observed that our proposed approach outperforms the top performing techniques by a significant margin.

We designed a second experiment with the aim to compare the *data-efficiency* of our approach with various sampling techniques. We compared multiple AL strategies to identify the approach that takes the smallest possible subset of total labeled data to match the performance (R@95P for positive class) observed using 100% of labels. We trained an initial classifier on 10% of the total available labeled data, and sampled 10% data in each iteration from the remaining pool using different AL based sampling strategies. We observed that all AL strategies performed better, requiring lesser data than random sampling. Our proposed approach performed the best, requiring only 37.2% of the labeled data to achieve the same results as 100% labeled data for UAE. The second best performing approach required nearly 67% more labeled data compared to our approach to achieve the results equivalent to 100% labeled data. Figure 2 highlights the comparison of percentage of labels required and R@95P for different sampling strategies for UAE. Similar impact was observed for IN as well.

Table 2: Delivery point geocoding performance across multiple metrics for IN and UAE

Model	Precision 0-50		DPMO >400	
	IN	UAE	IN	UAE
Production System	78.92	47.32	61k	177K
Our Approach	**86.81**	**56.71**	**54k**	**146K**

5 Real world application of Customer Address Matching

Preliminaries of Delivery Point Geocoding: Geocoding is the process of converting free-form customer address text to a geocode (pair of latitude-longitude). This geocode, called as delivery point (DP) geocodes, is used by the drivers to navigate and deliver the package to the customers. Learning high quality geocodes is also important for automatic route optimization, sequencing and container planning. Following are some of the key metrics used to measure the quality of DP geocodes. Following are some of the key metrics used to measure the quality of DP geocodes. (1) *Precision 0-50 meters* : It represents the percentage of total shipments for which delivery actually happened within *50 meters* distance from the planned delivery location. (2) *DPMO >400* : DPMO (*Defects Per Million Observations*) represents total number of shipments for which actual delivery happened outside *400 meters* distance of planned delivery point per million shipments. Lower the number of such shipments, better the metric. Having sufficient number of deliveries to an address allows us to learn reasonable quality geocodes by aggregating the past delivery locations [9]. Learning DP geocodes for cold-start addresses is particularly challenging because of lack of historical geocode data. Our real-time address matching service is an effective solution to the problem of DP learning for cold-start addresses. In order to learn a DP for newly created addresses, we match the new address against existing (known) addresses in the reference list (database) for which geocode information is available. We aggregate the geocodes of matched addresses to learn a single DP geocode using KDE.

Offline Evaluation and Impact: We perform offline evaluation and observed the impact on DP geocoding across actual set of deliveries done to all the cold-start addresses. For both IN and UAE, the deliveries that happened from January 2021 to March 2022 across all delivery stations were considered for creating the reference set (known addresses). Deliveries across the span of first two weeks of April 2021 were used as test set. During the test period, a few hundred thousand deliveries were done on cold-start addresses against which we evaluated our approach. We re-trained the base model by augmenting the initial CGT train set with record pairs sampled via our approach and observed the impact on DP geocoding metrics. From Table 2, significant improvements can be observed for both the DP geocoding metrics with our approach compared across IN and UAE.

6 Analysis

Quantitative Analysis: In order to study the importance of different elements of our approach, we did an ablation study to demonstrate the effectiveness of different components involved in our proposed framework. We aim to highlight the importance of graph clustering, bridges and min-cut via this study. In Table 3, we show how removing each of these components impact the performance on CGT test data across multiple metrics for UAE. Similar impact was observed for IN as well.
Qualitative Analysis: We analysed the DP geocodes predicted by the base model and our approach against the actual delivery location. The quality of predictions is highlighted through the following real-world scenario. *Down Town, Damac Maison Mall St., 3451, Downtown Burj, Dubai* is a newly created address on our e-commerce website. Figure 3 shows that the base model incorrectly matches the new address against multiple addresses from the adjoining streets (yellow points), hence learning an inaccurate DP (black point), resulting in a delivery defect when compared to the actual delivery location (green point). With our approach, the model accurately matches new address with reference set addresses from the from same building (orange points) to learn an accurate DP (violet point).

Table 3: Ablation study to show the impact of each component of our approach for UAE

Model	Accuracy (%)	R@95P (%)	PR AUC
Our Approach without clustering	87.96	78.77	0.967
Our Approach without bridges	89.15	79.69	0.971
Our Approach without min-cut	87.08	76.74	0.962
Our Approach	**90.54**	**81.11**	**0.974**

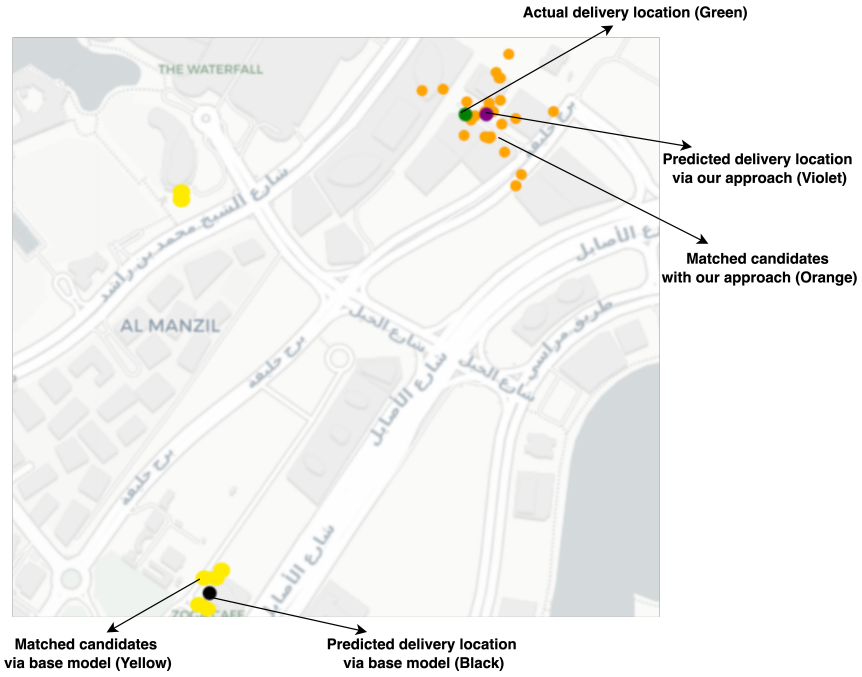

Figure 3: Demonstrates the quality of actual and predicted delivery locations

7 Conclusion

We propose a novel adaptation of graph-based active learning for customer address resolution. In comparison to existing baselines, experiments on manually curated dataset demonstrate that our proposed approach is highly effective for both IN and UAE. Our proposed approach is data-efficient as it requires only 37.2% of the labeled data to achieve the same results as 100% labeled data. Our offline experiments on DP geocoding show that when base model is augmented with active learnt pairs, significant improvement is observed across multiple business metrics for DP geocoding.

Customer Problem Statement

We introduced *delivery point geocode learning* as a downstream application of customer address matching. Improving address matching positively impacts delivery point geocoding as it enables delivering packages to customers in a cost-effective manner. These improvements lead to better delivery planning, significant decrease in operation costs, and customer satisfaction. Learning inaccurate delivery point geocodes leads to bad driver experience, which may require delivery associates to walk-around to find exact house numbers and determine the correct delivery location. This effort may get repeated each day by different drivers. Too many wrongly located delivery point geocodes leads to inefficient route planning, sending packages on the wrong vehicles, thus even missing the delivery date we have promised to customers, hence loss in customer trust. Further, periodic repetition of such events will lead to escalation and complaints, as delivery locations are difficult to locate and packages are not received as expected.

References

[1] M. Bilgic and L. Mihalkova. Active learning for networked data. In *ICML*, 2010.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, (10):P10008, 2008.

[3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[4] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz. Recent advances in graph partitioning. *Algorithm engineering*, pages 117–158, 2016.

[5] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on discovery and data mining*, pages 785–794, 2016.

[6] X. Chen, J. Zhexue Haung, F. Nie, R. Chen, and Q. Wu. A self-balanced min-cut algorithm for image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2061–2069, 2017.

[7] N. R. Chopde and M. Nichat. Landmark based shortest path detection by using a* and haversine formula. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(2):298–302, 2013.

[8] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

[9] G. Forman. Getting your package to the right place: Supervised machine learning for geolocation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 403–419. Springer, 2021.

[10] C. Gormley and Z. Tong. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.", 2015.

[11] J. Kasai, K. Qian, S. Gurajada, Y. Li, and L. Popa. Low-resource deep entity resolution with transfer and active learning. *arXiv preprint arXiv:1906.08042*, 2019.

[12] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier, 1994.

[13] K. Margatina, G. Vernikos, L. Barrault, and N. Aletras. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*, 2021.

[14] V. V. Meduri, L. Popa, P. Sen, and M. Sarwat. A comprehensive benchmark framework for active learning methods in entity matching. In *Proceedings of the 2020 ACM SIGMOD Conference on Management of Data*, pages 1133–1147, 2020.

[15] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, and V. Raghavendra. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, pages 19–34, 2018.

[16] T. Ningsheng, Y. Chongjun, Y. LiuZhong, and L. Yuan. An address regional tessellation method for spatial subdivision and geocoding in digital earth system. *International Journal of Digital Earth*, 8(10):825–839, 2015.

[17] A. Primpeli and C. Bizer. Graph-boosted active learning for multi-source entity resolution. In *International Semantic Web Conference*, pages 182–199. Springer, 2021.

[18] A. Ratner, S. Bach, Stephen H, and C. Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access, 2017.

[19] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. int. conf. on machine learning, 2001.

[20] D. W. Scott. Multivariate density estimation: Theory, practice and visualisation. john willey and sons. *Inc., New York*, 1992.

[21] B. Settles. Active learning literature survey. 2009.