Pretraining and Finetuning Language Models on Geospatial Networks for Accurate Address Matching

Saket Maheshwary*, **Arpan Paul***, **Saurabh Sohoney**
Last Mile, Amazon
{mahsaket, arppaul, sohoneys}@amazon.com

Abstract

We propose a novel framework for pretraining and fine-tuning language models with the goal of determining whether two addresses represent the same physical building. Address matching and building authoritative address catalogues are important to many applications and businesses, such as delivery services, online retail, emergency services, logistics, etc. We propose to view a collection of addresses as an address graph and curate inputs for language models by placing geospatially linked addresses in the same context. Our approach jointly integrates concepts from graph theory and weak supervision with address text and geospatial semantics. This integration enables us to generate informative and diverse address pairs, facilitating pretraining and fine-tuning in a self-supervised manner. Experiments and ablation studies on manually curated datasets and comparisons with state-of-the-art techniques demonstrate the efficacy of our approach. We achieve a 24.49% improvement in recall while maintaining 95% precision on average, in comparison to the current baseline across multiple geographies. Further, we deploy our proposed approach and show the positive impact of improving address matching on *geocode learning*.

1 Introduction

Entity matching (EM) (Barlaug and Gulla, 2021; Christen, 2019) aims to identify and link various representations of the same real-world entities across multiple databases. EM is a challenging task, particularly when entities are unstructured (Mudgal et al., 2018) and of limited data quality i.e. there is lack of completeness and consistency in their descriptions. Additionally, real-world EM tasks (Kasai et al., 2019) often have limited labeled data and require significant labeling effort to develop accurate models. In this paper, we pose address matching as an EM task to determine if two addresses represent the same physical building or not. Addresses are important to many businesses, such as logistics, online retail, and emergency services, as they are the primary source of information used to determine the location. They exhibit variations in writing styles and patterns, resulting in considerable discrepancies across similar addresses and their components (e.g., building, road). It is common to provide colloquial addresses that use landmarks and other points-of-interest (POI) to denote the place. For example[1], *"ABG Bank, Opp. Network Stone, Mahapurii"* and *"Plot No. 438 Taj Towers, ABG Bank, Mahapuri"* represent the same physical building but its hard to distinguish syntactically. Further, neighbourhood provided by a customer can also be known by other vernacular names or be a part of a larger neighbourhood. These synonyms are often used interchangeably, making it challenging to comprehend the addresses.

Language Models (LMs) have become the defacto approach to model real-world text. However, most of the efforts focus on general domain corpora. Recent studies (Gu et al., 2021; Liu et al., 2021; Yasunaga et al., 2022) show that domain-specific pretraining from scratch substantially outperforms continual pretraining of generic language models, thus demonstrating that the prevailing assumption in support of mixed-domain or general domain pretraining is not always applicable. Pretraining is followed by finetuning that specializes LMs by training it on in-domain dataset, but real-world data tends to be noisy. The LMs need to be exposed to diverse and high-quality examples for finetuning a pretrained model effectively as they directly affect the model's ability to comprehend. Lack of quality training data is a perennial problem (Thirumuruganathan et al., 2018; Kasai et al., 2019) for EM. Further, creating a representative

[*] Equal Contribution

[1] All examples are modified to preserve the privacy.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 763–773
November 12-16, 2024 ©2024 Association for Computational Linguistics

training set for address matching is challenging for multiple reasons — (1) Data distribution is heavily skewed towards negative pairs, i.e. no-match. (2) The average handle time for an annotator to label an address pair is *three times* higher on average when compared to other EM tasks. (3) Across addresses, it is very common that component values are vernacular, redundant, noisy, missing, or misspelled, thus leading to unstructured data problems. (4) Considering the current trend towards employing language models (LMs) for entity matching (Li et al., 2021, 2020), utilizing a few thousand samples result in over-fitting (Xie et al., 2019) the LMs. This necessitates having a more sophisticated approach for address matching.

In this paper, we tackle the above discussed challenges by proposing an effective strategy for pretraining and fine-tuning LMs that incorporates real-world knowledge among addresses via geospatial semantics. Given a corpus of addresses, we obtain links between addresses using historic delivery information and address text to create LM inputs by placing linked addresses in the same context window. Our approach thus provides a natural fusion of language-based and graph-based self-supervised learning. Our empirical evaluation shows significant improvements in pair-wise matching and geocode learning metrics compared to the existing baseline system and other state-of-the-art systems. Further, it should be noted that the structure of addresses are quite different for different geographies, hence the improvements observed across multiple geographies confirm the wide applicability and generic nature of our approach.

In summary, our main contributions are — (1) We introduce Neighbour Relation Prediction (NRP) training objective to pretrain LMs that enables the model understand neighbourhood level nuances and align on the address structure. (2) Our approach jointly integrates geospatial properties and address text with graph theory and weak supervision to curate diverse and informative address pairs to finetune the LMs in a self-supervised manner. (3) We deployed our solution for real-time geocode learning and evaluated its impacts on live traffic via online A/B experiments.

2 Related Work

We can divide prior literature into three broad categories — rule-based, crowd-based and learning based solutions. Rule-based solutions either rely on pre-defined matching rules such as DNF (Arasu et al., 2009) or dynamically synthesized EM rules (Singh et al., 2017) to find matching pairs. While rule-based solutions are highly interpretable, they are time and resource-intensive requiring domain experts to define the rules and may perform poorly on unstructured data (Mudgal et al., 2018). To alleviate these drawbacks, crowd-based solutions (Maheshwary and Misra, 2018; Firmani et al., 2016; Wang et al., 2012) have been proposed that employ crowd-sourcing to manually identify matching tuples. However, such methods are time consuming and human labor cost is expensive which makes them not suitable for large scale real-world applications.

Recently (Maheshwary and Sohoney, 2023) leveraged active learning with graphs to improve matching performance for geospatial entities. Currently, the state-of-the-art solutions for EM now predominantly rely on deep learning or LM based approaches. Ditto (Li et al., 2020) casts EM as a sequence-pair classification problem based on fine-tuning pretrained LMs across different domains. GeoBERT (Liu et al., 2021) integrate semantics and geographic information in the pre-trained representations of POIs by mapping multiple geographic granularity into a unified latent space, to obtain the POI embeddings with geographic information. Recently proposed, GeoER (Balsebre et al., 2022) includes a transformer block, a geocoding block, and a neighbourhood block and is widely used in wide variety of geospatial systems.

3 Methodology

We present a self-supervised approach for pretraining and fine-tuning language models (LMs) with the aim of internalising spatial knowledge into LMs via geospatial semantics. Instead of viewing the address corpus as a list of addresses, we view it as an address graph, where each node in the graph represents an address and edges between nodes capture spatial relevance between addresses. The edges of an address graph can be created using various techniques; in our case, we use historical delivery data to sample address pairs and assign spatial links based on the H3 geospatial indexing system (Woźniak and Szymański, 2021) for model pretraining. We also introduce the Neighbour Relation Prediction (NRP) training objective to pretrain LMs. This objective enables the model to understand neighbourhood-level nuances and align with

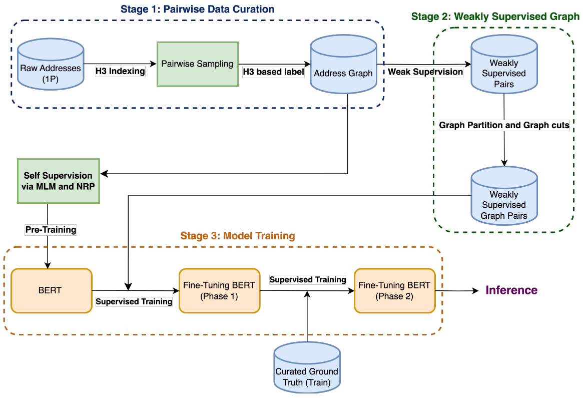
Stage 1: Pairwise Data Curation
Raw Addresses (1P)
H3 Indexing
Pairwise Sampling
H3 based label
Address Graph
Weak Supervision
Stage 2: Weakly Supervised Graph
Weakly Supervised Pairs
Graph Partition and Graph cuts
Weakly Supervised Graph Pairs
Self Supervision via MLM and NRP
Pre-Training
Stage 3: Model Training
BERT
Supervised Training
Fine-Tuning BERT (Phase 1)
Supervised Training
Fine-Tuning BERT (Phase 2)
Inference
Curated Ground Truth (Train)

Figure 1: Workflow of our proposed approach

the address structure. While fine-tuning, we generate spatial links among sample address pairs by integrating weak supervision with graph theory that leverages address text with historic delivery information. The intuition here is to let the model learn diverse variations across similar physical buildings within a neighbourhood. The workflow of our proposed framework is demonstrated in Figure 1 and discussed in Section 3.2 and 3.3.

3.1 Problem Statement

Let A_1 and A_2 denote a pair of address text. Entity Matching is a binary classification task that aims to determine a *match* or *no-match*. For our problem domain, a *match* represents an address pair $< A_i, A_j >$, belonging to the same physical building whereas *no-match* represents an address pair referring to different buildings. The entire Cartesian product becomes too large across addresses database, making it infeasible to run a high-recall classifier directly. Following the literature, standard practice is to decompose this problem into two steps: *blocking* and *matching*. Blocking filters obvious no-matches from the Cartesian product to obtain a candidate set. We use ElasticSearch (Gormley and Tong, 2015) with deep metric learning (Govind and Sohoney, 2022) to index the addresses and then filter obvious no-match addresses. We retrieve *top-k* candidates for every address and apply pairwise-matching.

3.2 Tasks for Pretraining

Data Curation: Several works (Gao et al., 2020; Levine et al., 2021) show that LMs can learn stronger dependencies between words that were shown together in the same context during training, than words that were not. To effectively learn geospatial knowledge across addresses, we create LM inputs by placing spatially linked addresses in the same context. For address matching, we leveraged H3 grids (Woźniak and Szymański, 2021) as an approximate solution to retrieve positive and negative address pairs (Govind and Sohoney, 2022). The additional details on H3 grids are discussed in Appendix B. Specifically, we sample an anchor address from every H3 grid, T positive addresses are sampled from the H3 grid of same level L, T negative addresses are sampled from *1-skip* neighbouring grids (i.e. level $L-1$). We generate positive and negative pairs at different resolution levels to compile a more diverse training data. We assign a spatial link for anchor address with corresponding T positive addresses sampled from the same H3 grid to generate an address graph $\mathcal{G}$.

Training Objectives: To train the LM, we use two objectives. We apply the Masked Language Model (MLM) objective to encourage the LM to learn the inherent structure of addresses and their colloquial patterns. We also propose a Neighbour Relation Prediction (NRP) objective, which classifies the relation r of address X_a to X_b as $r \in \{Same, Different\}$. By distinguishing at neighbourhood level, NRP enables the LM to learn the relevance and variations in lexical structure between addresses across H3 grids, besides the capability learned in the vanilla Next Sentence Prediction (NSP) objective. To predict r, we use the representation of $[CLS]$ token, as used in NSP. The training objectives taken together, we optimize:

$$\mathcal{L} = \mathcal{L}_{MLM} + \mathcal{L}_{NRP} \tag{1}$$

$$= -\sum_i log\, p(x_i|h_i) - log\, p(r|h_{[CLS]}) \tag{2}$$

where x_i is each token of the input instance, [CLS] X_a [SEP] X_b [SEP], and h_i is its representation.

3.3 Tasks for Finetuning

We jointly leverage address text, historic delivery information and concepts from graph theory, namely graph partitioning, graph cuts and graph transitivity along with weak supervision to curate informative and diverse record pairs to finetune the pretrained model and determine if two addresses represent the same physical building or not. The data curation strategy for model fine-tuning is shown in Figure 2.

Weak Supervision: Given a list of unique address with corresponding geospatial attributes like address text, historic delivery geocodes, we aim to

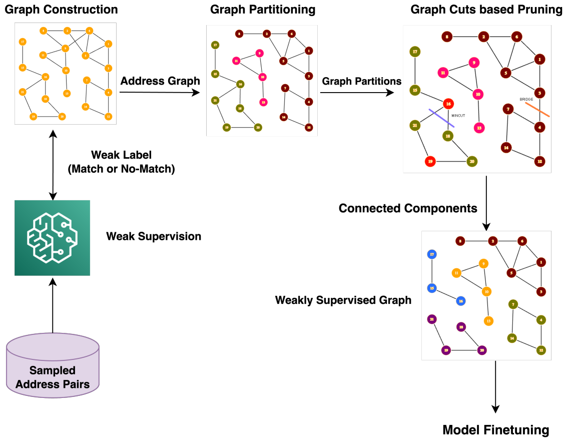

Figure 2: Self-supervised data curation strategy via graph based weak supervision for model finetuning

assign a weak label to pair of addresses. If the address pair refers to same physical building we call it *match* else *no-match*. We leverage stacked BiLSTM+CRF based address parser (Zhang et al., 2018; Panchendrarajan and Amaresan, 2018) to extract structured chunks of information (unit, building, road, etc.) from each address text. The details around address parsing are discussed in Appendix C. Further, we use historic geocodes associated with each address to learn a single geocode. A brute force approach would be to compute the centroid of geocode points from past deliveries. Centroids and medoids are prone to outliers, hence proving inaccurate in estimating geocodes (Forman, 2021). We use density-based methods to accurately approximate a single geocode from historical deliveries for each address via Kernel Density Estimation (KDE) (Scott, 1992). To determine a weak label for an address pair, we use KDE geocodes to determine proximity among addresses, along with similarity of respective address parser components.

Graph Construction: Each address is represented via a node and the edge between two nodes is determined via weak supervision to construct G. We add an edge for every matching pair, while we skip the edge for every non-matching pair. We leverage *transitivity* of an address graph G to discover *false negatives* from the predictions of weak supervision. However, given that the edges of the graph are derived via weak supervision, which are not always accurate, a wrongly predicted match edge can lead to a series of *false positives*.

Graph Partitioning and Graph Cuts: We use graph partitioning and graph cuts to find and remove likely false positive edges from the graph and obtain smaller connected components (CC) so that the set of nodes within the same CC represent addresses from the same physical building as shown in Figure 2. The idea is motivated from graph active learning work (Maheshwary and Sohoney, 2023) to which we make two notable changes — (1) we use weak supervision instead of multiple rounds of active learning which is expensive and time consuming, and (2) we leverage weak labels instead of probability prediction score of the model to determine an edge between nodes of the graph. After graph construction, we apply a single pass of Louvain algorithm (Blondel et al., 2008), a linear time operation to separate the nodes into multiple mutually exclusive graph partitions. We use graph cuts to prune weak links and isolated components. We leverage minimum cut (Akiba et al., 2016) and bridges as graph cut techniques to prune the *likely false positive* edges from the graph. The node pairs to cut are determined by setting a threshold on haversine distance. The details around formulation and choice of haversine distance are discussed in Appendix D. We remove min-cut edges from the graph to get a pruned graph G_{pruned}. To learn a graph label in self-supervised manner, we first compute all the CC in G_{pruned}. For all node pairs belonging to the same CC, we assign a *match* label else *no-match* label is assigned which are then used to finetune the model. To ensure *geospatial-diversity* among record pairs, we sample across a H3 grid (Woźniak and Szymański, 2021).

3.4 Model Training

For pretraining, we create LM inputs by placing tens of millions of linked pairs together and masking a small percentage of tokens. We then train the LM with two self-supervised objectives: masked language modeling (MLM), which predicts masked tokens in the addresses, and Neighbour Relation Prediction (NRP), which classifies the relation between address pairs as *same* or *different* neighbourhood. For the MLM task on addresses, the positions that need to be masked are randomly selected. Among the selected positions, 80% of the time we replace that position with the [MASK] token, 10% by random words, and the remaining 10% is kept original. We observed that randomly selecting positions for masking provides marginal improvements in the pretraining performance against selecting specific positions.

Lastly, we propose a two-phase strategy for finetuning our pretrained LM model with additional fully connected layers. In the first phase, we freeze the first l layers of the pretrained LM and train the remaining layers using a few million weakly supervised graph-based labels as input. This enables the model to grasp the concept of geospatial proximity as well as retain spatial and lexical knowledge from the pretraining. During weak supervision, the use of BiLSTM+CRF (Panchendrarajan and Amaresan, 2018) address parser can introduce some noise, as can other weak learners, for example, geocode of an address determined from historic deliveries (Forman, 2021). However, our approach is robust as we tackle such noise during two-phase model fine-tuning. During the first phase of fine-tuning, the model tends to overfit the noise from weakly supervised graph labels. To overcome this limitation, we propose a second phase where we further freeze the rest of LM layers and fine-tune the fully connected layers on a few thousand high-quality address pairs curated by human annotators. The primary purpose of two-stage fine-tuning is to denoise such pairs while simultaneously learning proximity relations. The second stage prevents the model from overfitting the noise of weak labels by learning from manually curated data, thus making our proposed framework robust to noise.

4 Experiments

We did extensive offline experimentation to develop, refine, and validate our approach. In this section, we describe the experiments and discuss results across three diverse geographies $G1$, $G2$, and $G3$ to ensure our approach is generic and makes a positive impact across geographies with different address standards, writing styles, and language variations. These geographies belong to the South America, Europe, and Asia continents. Our experiments leverage historic delivery information and address text that contains information related to building, street, landmark, postal code, etc. Our proposed approach holds fair for all types of addresses, for example, urban, rural, commercial, household, etc., and locations. The structure of addresses and writing styles are diverse for these geographies; hence, the improvements observed across all these geographies confirm the wide applicability and generic nature of our approach. While we have limited the evaluation to certain geographies in this paper, our approach is robust for all types of addresses across any geographical continent. The positive results observed across multiple pairwise-matching and geocoding metrics demonstrate the efficacy and effectiveness of our approach.

4.1 Human-Labeled Data (HLD)

We did stratified sampling of addresses for each geography to cover all the linguistic and address writing styles and abbreviations across the country. The selection also ensures to consider the varied density of addresses, i.e., probable urban vs. rural/outskirts split to generate around $10K$ address pairs where 40% are from match class and 60% from no-match class for each geography, which are then manually labeled by the data annotation team.

4.2 Baselines

We evaluate the efficacy of our proposed approach in Table 1 against existing matching model (Baseline) and multiple state-of-the-art techniques that we discussed in related work section, namely CharEdit (Shapira and Storer, 2007), Ditto (Li et al., 2020), GeoBERT (Liu et al., 2021), Mistral 7B (Jiang et al., 2023; Peeters and Bizer, 2023), GeoER (Balsebre et al., 2022) and GAL (Maheshwary and Sohoney, 2023). The additional details on these baselines are discussed in Appendix A.

4.3 Parameter Settings

After experimenting with different LMs, we have settled on BERT (Devlin et al., 2018) as it offers the best trade-off between latency, operating cost, and quality. We begin our pretraining objective in each geography by initializing our language model with a 6-layer BERT model using the Hugging Face interface. We fine-tune the [CLS] token of the language model by adding two fully connected layers infused with BatchNorm and Dropout that act as a binary classifier. For all the geographies, we use Adam optimizer with an initial learning rate of $3e$-5, dropout of 0.15 and a batch size of 32 for 12 epochs and we do resort to early stopping to prevent overfitting.

4.4 Results

We split the HLD 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. A high precision (95% precision of match class) matching model is required for geocode learning, hence we evaluate it across three metrics — (1) Accuracy, (2) Recall at 95% Precision (R@95P),

Model	Accuracy (%)			R@95P (%)			PR-AUC		
	G1	G2	G3	G1	G2	G3	G1	G2	G3
Baseline	90.39	91.89	87.42	85.46	78.13	32.33	-	-	-
CharEdit	71.12	77.61	69.36	51.78	45.32	10.87	-	-	-
GeoBERT	91.89	94.83	85.44	87.02	85.71	23.23	96.98	94.08	82.81
Ditto	92.21	94.72	88.86	86.50	89.45	27.78	97.12	95.98	83.07
Mistral 7B	85.25	86.87	79.02	78.07	70.14	11.42	-	-	-
GAL	92.71	95.12	90.44	91.79	90.02	30.65	97.77	96.51	86.78
GeoER	92.92	95.01	89.98	92.12	89.91	28.46	97.94	96.39	86.08
Our Approach	**93.84**	**96.12**	**93.07**	**96.13**	**94.21**	**53.67**	**98.45**	**98.56**	**90.75**
Our Approach w/o pretraining	91.07	95.00	90.48	90.13	88.65	37.11	97.57	97.41	86.80
Our Approach w/o phase 1 finetuning	93.77	95.05	92.51	95.06	91.01	46.77	98.23	97.48	89.97
Our Approach w/o phase 2 finetuning	92.06	95.13	90.58	91.58	91.60	38.1	97.54	97.33	87.52

Table 1: Performance of various models across pair-wise matching metrics for three geographies

and (3) Precision-Recall area-under-the-curve (PR-AUC). The R@95P and PR-AUC numbers are corresponding to the match class to align the performance of the model for accurate geocode learning. From the Table 1, we observe that our approach significantly outperforms all the baselines. Overall on an average, our approach shows an improvement of 24.49% on R@95P and 4.94% on Accuracy across three geographies when compared to the current baseline. In comparison to the top performing state-of-the-art approach, our approach improves R@95P by 11.43%.

The performance of G3 is significantly lower than G1 and G2 in Table 1, as a majority of proportion of addresses in G3 are unstructured, i.e., the addresses are vernacular, redundant, noisy, and are missing key components from addresses like building or street information. Further, providing colloquial addresses that use landmarks and other points-of-interest (POI) to denote the same place is highly frequent in G3 compared to G1 and G2.

5 Real-world Application

Address matching is a fundamental problem to many business applications. In this section, we highlight the positive impact of improving address matching for geocoding and highlight the impact observed via online A/B experiment.

5.1 Preliminaries of Geocoding

Geocoding is the process of converting free-form address text to a geocode (pair of latitude-longitude). For this paper, we limit the scope of geocode learning for cold-start addresses. The key metrics to measure the quality of geocodes are – (1) *Delivery Precision* is the percentage of total shipments for which the actual delivery happened within a threshold distance $\mathcal{Z}$ from the planned location. (2) *Delivery Defects* is the percentage of total shipments for which the actual delivery happened outside of the threshold distance $\mathcal{Y}$ from the planned location. Hence, lower the value of outliers, better the metric. Dealing with new emerging addresses is important to many applications and businesses, such as delivery services, online retail, emergency services, logistics, etc. Any real-world problem associated with new addresses is particularly challenging due to the lack of availability of historic data. Address matching provides an effective solution for learning geocodes by matching new address against known reference list (database) for which geocode information is available. We then aggregate the geocodes of all matched addresses to learn a single geocode using Kernel Density Estimation (KDE) (Scott, 1992). Equation 3 below formulates the KDE P over the matched addresses M where $K(x;h)$ is a Gaussian kernel with haversine distance metric. The bandwidth h works as a smoothing parameter which we determine based on our use-case after validation.

$$P_h(x) = \frac{1}{|M|h} \sum_{n=M} K(x-n;h) \qquad (3)$$

5.2 Online A/B Experiment

After observing significant improvements during offline simulations, we launched an online A/B experiment on live traffic to determine the impact of our proposed approach on geocode learning. We performed the model dial-up in a phased manner — $10\%, 50\%, and\ 100\%$ traffic. We observed statistically significant improvements during one week of dial-up in each phase. During the A/B test period, our approach learnt geocodes for a few hundred thousand shipments, where we observed 14.68%

improvement in delivery precision and 8.79% reduction in delivery defects.

6 Analysis

We analysed our approach and show that it offers the best quality, latency, and operating cost.

6.1 Quantitative Analysis

To study the importance of different elements, we did an ablation study to show the effectiveness of various components involved in our proposed framework. We aim to highlight the importance of proposed domain-specific pretraining, and different phases of finetuning via this study. In Table 1, we show how removing each of these components impact the performance on HLD test data across multiple address matching metrics.

6.2 Qualitative Analysis

The address pair, *"ABG Bank, Opp. Network Stone, Mahapurii"* vs. *"Plot No. 438 Taj Towers, ABG Bank, Mahapuri"* is an example of *matching* address pair that was not correctly predicted by the existing baseline but is learnt correctly by our proposed approach. Further, we analysed the geocodes predicted by the baseline and our approach against the actual delivery location. The quality of predictions is highlighted through the following real-world scenario. *"ABG Bank, Opp. Network Stone, Mahapurii"* is a newly created address and Figure 3 shows that the existing baseline incorrectly matches this address against multiple addresses from the adjoining streets (gray dots), hence learning an inaccurate geocode (blue marker), resulting in a delivery defect when compared to the actual delivery location (black marker). With our approach, the model accurately matches new address with reference addresses from the same building (orange dots) to learn an accurate geocode (green marker).

6.3 Latency Analysis

We assessed the latency of our approach with Baseline, GeoER, and Mistral models. To evaluate the models on a common ground, the interface setup assumes a query address and a list of reference addresses as input, and outputs matched addresses. We built all models in PyTorch on the same machine configuration (g5.8xlarge). We observed that Baseline, GeoER and Mistral have higher inference latency, *3-times*, *5-times* and *20-times* respectively, thus requiring significantly more hardware to reach the same TPS (transactions per second).

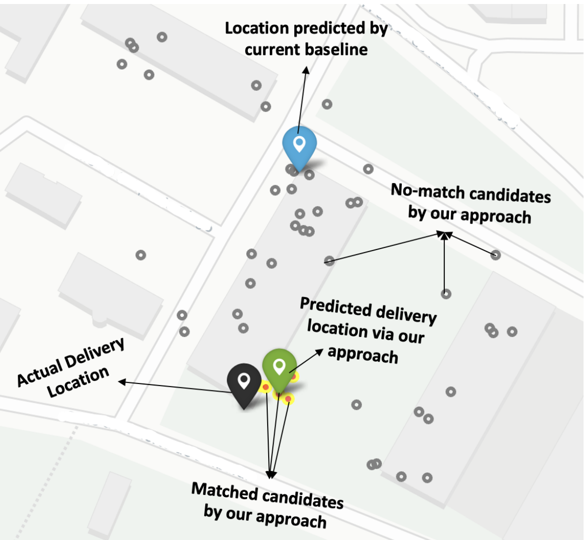
Location predicted by current baseline
No-match candidates by our approach
Predicted delivery location via our approach
Actual Delivery Location
Matched candidates by our approach

Figure 3: Quality of geocode predictions for the current baseline and our approach against the actual location

7 Conclusion

We proposed a novel framework for pretraining and fine-tuning LMs aimed at address matching. It integrates concepts from graph theory and weak supervision with address text and geospatial semantics to generate informative and diverse pairs, thus facilitating pretraining and fine-tuning in a self-supervised manner. We introduced Neighbour Relation Prediction (NRP) as a new pretraining objective. We deployed our approach for real-time geocode learning and presented results from online A/B experiments. We observed improvement in delivery precision and reduction of delivery defects. This led to better delivery planning, decrease in operation costs, and better customer experience.

8 Future Work

We are exploring ways to leverage LLMs as part of future directions. We explored synthetic truth generation via knowledge distillation, a popular way to effectively leverage LLMs. The latency constraints in deploying models for our problem statement in a real-world setting and the domain-specific nature of our problem prevent us from using LLMs directly, even via knowledge distillation. Further, comparisons with LLM-based baselines in Table 1 reveal that LLMs in their existing form might not be sufficient for our problem. In order to make it effective in our problem setting, we need to infuse geospatial domain knowledge within LLMs. As part of next steps, we are exploring ways to invest further in domain-specific LLMs for geospatial applications.

Limitations

For graph cuts, the source and target node pairs to cut are determined via the haversine distance between a given node pair. The geocode associated with each node is a KDE geocode, which is determined from real-world historic deliveries, which can be noisy. This can lead to incorrect pruning of edges, which will impact the learnt graph labels. Learning incorrect graph labels directly impacts the fine-tuning stage and eventually the model performance. The task of introducing orthogonal sources of information to disambiguate such scenarios and enhance the overall performance is taken up as part of our future work.

Ethical Statement

This work aims to develop a robust and computationally efficient solution for address matching, leveraging prior research on graph theory, weak supervision, and encoder-based transformer models. Our proposed model primarily makes a binary prediction, and the focus is on classification rather than generation; hence, the risks associated with generative content, for example, leaking any address-specific information, do not apply. Our systems follow stringent mechanisms to ensure that the datasets are anonymised and do not contain any identifiable or traceable information. The anonymised data elements are not combined with other elements or behaviour data that could cause them to be de-anonymised. We use it within well-defined handling standards and only for the purpose of improving the delivery experience. Thus, we respect the privacy and confidentiality of the customers and do not expose them to any potential harm or misuse. In this paper we have limited the evaluation to certain geographies, but the methodological innovations are generic in nature, and the same approach is applicable to all types of addresses for any geographical continent across the world. Our work maintains a purely objective approach and adheres to being fair and non-discriminative throughout our research and reporting process. Our work does not introduce any bias or prejudice either, as we do not make any assumptions or judgements based on the addresses or delivery information. Our work is intended to improve the delivery experience and is not associated with any direct negative social impact.

References

Takuya Akiba, Yoichi Iwata, Yosuke Sameshima, Naoto Mizuno, and Yosuke Yano. 2016. Cut tree construction from massive graphs. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 775–780. IEEE.

Arvind Arasu, Christopher Ré, and Dan Suciu. 2009. Large-scale deduplication with constraints using dedupalog. In *2009 IEEE 25th International Conference on Data Engineering*, pages 952–963. IEEE.

Pasquale Balsebre, Dezhong Yao, Gao Cong, and Zhen Hai. 2022. Geospatial entity resolution. In *Proceedings of the ACM Web Conference 2022*, WWW '22, New York, NY, USA. Association for Computing Machinery.

Nils Barlaug and Jon Atle Gulla. 2021. Neural networks for entity matching: A survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(3):1–37.

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*, (10):P10008.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754.

Nitin R Chopde and Mangesh Nichat. 2013. 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.

Peter Christen. 2019. Data linkage: The big picture. *Harvard Data Science Review*, 1(2).

Sam Comber and Daniel Arribas-Bel. 2019. Machine learning innovations in address matching: A practical comparison of word2vec and crfs. *Transactions in GIS*, 23(2):334–348.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Donatella Firmani, Barna Saha, and Divesh Srivastava. 2016. Online entity resolution using an oracle. *Proceedings of the VLDB Endowment*, 9(5):384–395.

George Forman. 2021. 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.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.".

Govind and Saurabh Sohoney. 2022. Learning geolocations for cold-start and hard-to-resolve addresses via deep metric learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 322–331, Abu Dhabi, UAE. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource deep entity resolution with transfer and active learning. *arXiv preprint arXiv:1906.08042*.

Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. 2021. The inductive bias of in-context learning: Rethinking pretraining example design. *arXiv preprint arXiv:2110.04541*.

Bing Li, Yukai Miao, Yaoshu Wang, Yifang Sun, and Wei Wang. 2021. Improving the efficiency and effectiveness for bert-based entity resolution. In *AAAI Conference on Artificial Intelligence*.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *CoRR*, abs/2004.00584.

Xiao Liu, Juan Hu, Qi Shen, and Huan Chen. 2021. Geo-bert pre-training model for query rewriting in poi search. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2209–2214.

Saket Maheshwary and Hemant Misra. 2018. Matching resumes to jobs via deep siamese network. In *Companion Proceedings of the The Web Conference 2018*, pages 87–88.

Saket Maheshwary and Saurabh Sohoney. 2023. Learning geolocation by accurately matching customer addresses via graph based active learning. In *Companion Proceedings of the ACM Web Conference 2023*, pages 457–463.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, An-Hai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, pages 19–34.

Rrubaa Panchendrarajan and Aravindh Amaresan. 2018. Bidirectional lstm-crf for named entity recognition. 32nd Pacific Asia Conference on Language, Information and Computation.

Ralph Peeters and Christian Bizer. 2023. Entity matching using large language models. *arXiv preprint arXiv:2310.11244*.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

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

Dana Shapira and James A Storer. 2007. Edit distance with move operations. *Journal of discrete algorithms*, 5(2):380–392.

Rohit Singh, Venkata Vamsikrishna Meduri, Ahmed Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. 2017. Synthesizing entity matching rules by examples. *Proceedings of the VLDB Endowment*, 11(2):189–202.

Saravanan Thirumuruganathan, Shameem Ahamed Puthiya Parambath, Mourad Ouzzani, Nan Tang, and Shafiq R. Joty. 2018. Reuse and adaptation for entity resolution through transfer learning. *CoRR*, abs/1809.11084.

Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. 2012. Crowder: Crowdsourcing entity resolution. *arXiv preprint arXiv:1208.1927*.

Szymon Woźniak and Piotr Szymański. 2021. Hex2vec: Context-aware embedding h3 hexagons with openstreetmap tags. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, pages 61–71.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. Unsupervised data augmentation. *CoRR*, abs/1904.12848.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*.

Yuan Zhang, Hongshen Chen, Yihong Zhao, Qun Liu, and Dawei Yin. 2018. Learning tag dependencies for sequence tagging. In *IJCAI*, pages 4581–4587.

Appendix

A Model Baselines

In this section, we discuss the details of some of the top performing the baselines.

- **Baseline:** Following (Comber and Arribas-Bel, 2019), we first parse the addresses using our address parser into address fields (unit, building, road, locality). Further we engineer features, such as cosine similarity and fuzzy match score of address pairs for all the parsed address fields to perform matching using the XGBoost (Chen and Guestrin, 2016).
- **GeoER:** The architecture of this model (Balsebre et al., 2022) includes a transformer block, a geocoding block, and a neighborhood block and is widely used in geospatial systems. It requires historic delivery information to perform effectively.
- **Ditto:** It casts EM as a sequence-pair classification for product matching and finetune pretrained LMs to obtain the best performance among all the existing supervised ER works (Li et al., 2020). Unlike product matching, specific spans of tokens are not readily available in case of free flowing texts like customer addresses. To make this model work effectively for matching, we use our address parser to extract structured components as specific token spans.
- **GeoBERT:** It integrate semantics and geographic information in the pre-trained representations of POIs (Liu et al., 2021) by mapping multiple geographic granularity into a unified latent space, which helps obtain the POI embeddings with geographic information. For our problem statement, we modify this approach to get building level embeddings.
- **Mistral 7B:** We use Mistral as our decoder-based generative large LM baseline. Our prompt is specifically crafted to incorporate both geospatial context and raw customer address text as input for the decoder model.
- **GAL:** Recently (Maheshwary and Sohoney, 2023) leveraged graph based active learning with XGBoost (Chen and Guestrin, 2016) classifier to improve matching performance for geospatial entities for buildings.

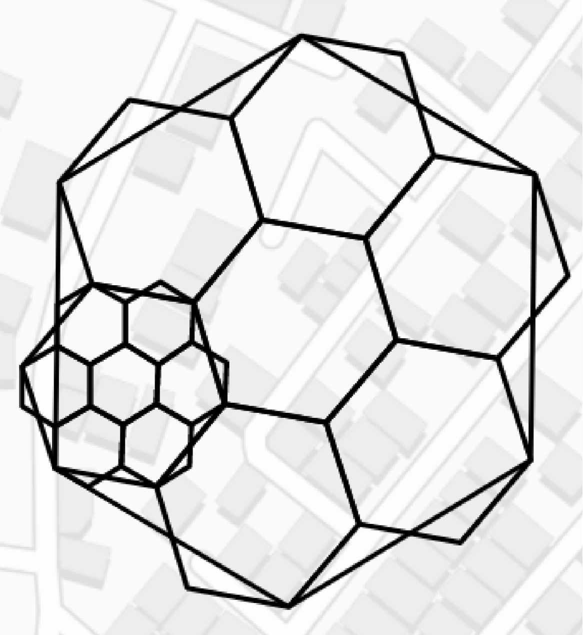

Figure 4: Demonstrates the hierarchy of H3 parent and its seven child grids

B H3 Hexagonal Grids

H3 is a hexagonal hierarchical geospatial indexing[2] spatial data structure (Woźniak and Szymański, 2021; Govind and Sohoney, 2022) which subdivides the space into buckets of hexagonal grids. Each hexagonal grid has seven hexagon grids as children in the hierarchy below it, thereby a hexagon of resolution L have seven child hexagons of resolution $L + 1$ and so on as shown in Figure 4. These hexagonal grids provide more uniform coverage of the Earth's surface compared to squares or rectangles, offer better adjacency, and their hierarchical nature allows for efficient handling of large-scale spatial data. Using a hexagon as the cell shape is critical for H3. Hexagons have only one distance between a hexagon's center-point and its neighbour's, compared to two distances for squares or three distances for triangles. This property greatly simplifies performing analysis and smoothing over gradients. We briefly explored other indexing methods, but they came with their own disadvantages. QuadTrees and R-Trees are efficient but can become complex. Geohash uses rectangular grids, which can distort spatial queries. Hilbert curves, while useful, are less intuitive. Keeping the aforementioned comparisons in mind, we went with the H3 index for sampling address pairs.

[2] https://h3geo.org/

C Address Parsing

The address parser extracts structured chunks of information from each free-form customer address text. Extracting such structured or meaningful information is a sequence tagging or entity extraction problem. For example, given the free-form address text, *"Bukharaon St, 123, Flat no. 321, Mahapurii"*, the components extracted from address parser are – *Apartment: "321"*, *Building: "123"*, *Road: "Bukharaon St"*, *Locality: "Mahapurii"*. We use stacked BiLSTM+CRF (Zhang et al., 2018), a deep learning architecture for address chunking tasks across all geographies. The parser uses BiLSTM (Schuster and Paliwal, 1997) that captures the semantics from free-form text for chunking task and use fastText embeddings (Bojanowski et al., 2017) for address token representations. The structured components extracted from parser are utilized for creating rules for weak supervision. We compute the fuzzy similarity scores between same parsed components for an address pair to generate a weak label from address parser. Note that the address components extracted by the parser are exclusively employed during weak supervision only and not used during model inference.

D Haversine Distance

The Haversine distance (Chopde and Nichat, 2013) is used to calculate the distance between two points on the surface of a sphere, given their latitudes and longitudes. This distance metric is particularly useful in navigation and geography because it accounts for the spherical shape of the Earth. Also known as great circle distance, this formula accurately computes the the shortest path over the Earth's surface, making it essential for navigation and geospatial analysis. Its simplicity is another key benefit; the formula is easy to implement and relies on basic trigonometric functions, making it accessible for a wide variety of applications.

$$d = 2 \cdot R \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cdot\cos(\phi_2)\cdot\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \tag{4}$$

Additionally, the it provides good accuracy for distances up to a few thousand kilometers, ensuring reliable results for most practical purposes. Lastly, it avoids complications associated with other distance formulas, such as the Law of Cosines, by not requiring special cases for certain point positions, thereby enhancing its usability in various scenarios. The haversine distance d between two points is computed as shown in equation 4, where R is is the Earth's radius (mean radius = $6,371$ km), ϕ_1 and ϕ_2 are the latitudes of the two points (in radians) with $\Delta\phi$ as the difference between latitudes, λ_1 and λ_2 are the longitudes of the two points (in radians) with $\Delta\lambda$ as the difference between longitudes.