Accurate Customer Address Matching via Weak Supervision for Geocode Learning

Arpan Paul*
arppaul@amazon.com
Last Mile, Amazon
Hyderabad, India

Saket Maheshwary*
mahsaket@amazon.com
Last Mile, Amazon
Hyderabad, India

Saurabh Sohoney
sohoneys@amazon.com
Last Mile, Amazon
Hyderabad, India

Abstract

Determining the precise location of customers is important for an efficient and reliable delivery experience, both for customers and delivery associates. Address text is a primary source of information provided by customers about their location. In this paper, we study the important and challenging task of matching free-form customer address text to determine if two addresses represent the same physical building. We introduce a novel address matching framework that leverages *transformer-based encoder* to prevent tedious and time-consuming efforts spent on manual feature engineering by the baseline model. Furthermore, our proposed framework employs *weak supervision* to leverage historic delivery information and generate high-quality labeled data. This reduces the requirement for massive amounts of labeled data, typically needed for transformer-based models. Our experiments on manually curated datasets demonstrate the effective and generic nature of our approach, as we achieve 15.57% improvement in recall at 95% precision, on average, compared to the current baseline model across *four* geographies. We also introduce *delivery point (DP) geocode learning* for cold-start addresses as a downstream application of customer address matching. In addition to offline experiments, we performed online A/B experiments for DP geocode learning with our proposed approach and observed delivery precision improved by 8.09% and delivery defects reduced by 11.78% on average across *four* geographies in comparison to the baseline model.

CCS Concepts

• **Information systems** → **Entity resolution**; **Language models**;
• **Computing methodologies** → **Natural language processing**;
• **Applied computing** → *Transportation.*

Keywords

Entity Matching, Weak Supervision, Language Models, Geocoding

ACM Reference Format:
Arpan Paul, Saket Maheshwary, and Saurabh Sohoney. 2024. Accurate Customer Address Matching via Weak Supervision for Geocode Learning. In *The 32nd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '24), October 29-November 1, 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3678717.3691277

*Both authors contributed equally.

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

ACM ISBN 979-8-4007-1107-7/24/10
https://doi.org/10.1145/3678717.3691277

1 Introduction

Entity matching (EM), also known as entity resolution (ER) [3, 9] 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 highly unstructured [29] and of limited data quality i.e. there is lack of completeness and consistency in their descriptions. Additionally, real-world EM tasks [22] often have limited labeled data and require significant labeling effort to develop accurate models. In this paper, we pose customer address matching as a entity matching task with the aim to determine if two addresses represent the same physical building or not.

For Amazon Logistics (AMZL), customer addresses are crucial for delivery planning as they provide the primary information regarding the customer's location. Customers typically provide their addresses in free-form text fields, which may or may not follow a consistent pattern. Address writing styles and patterns are idiosyncratic in the same way as hand writing or signatures. Across North America (NA) and Europe (EU), a significant number of outlier addresses exhibit major variations in writing patterns, resulting in considerable discrepancies across similar addresses and their components (e.g., building, campus, road, landmark). This necessitates having a more sophisticated address matching approach. The challenge becomes more pronounced in geographies like United Arab Emirates (UAE), Egypt (EG) and Kingdom of Saudi Arabia (KSA), where there is no widely accepted standard addressing system, thus making addresses inherently unstructured. Moreover, majority of addresses across these regions contain Arabic tokens, thus introducing an added complexity of multilingual address matching. Customers often use references to neighbourhoods, landmarks or points-of-interest (POI). For example, it is common for customers across geographies to provide colloquial addresses that use landmarks and other POI to denote the place, for example, *ABG Bank, Opp. Network Stone, Mahapurii*. Other customers may provide more structured addresses that intend to indicate the same place but also conform to local postal standards, for example, *Plot No. 438 Taj Towers, ABG Bank, Mahapuri*. In the aforementioned examples[1], both addresses refer to the same place in *Mahapuri* neighbourhood. The first example mentions *Network Stone* building as a landmark, refers *Opp.* for opposite and mentions *ABG Bank* as the place. In the second example, the number and name of the building *Taj Towers* is used to mention the same *ABG Bank* as the place. Further, neighborhood provided by a customer can also be known by other

[1]All examples in this paper are modified to preserve the privacy of customers.

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 customer address even more challenging to comprehend.

The current baseline model employs a tree-based based address matching framework which relies on hand crafted features from free-form text as well as their components (building, road, landmark). The process of manual feature engineering is tedious and time-consuming. Further, this model can only operate on English addresses and relies on AWS Translate for translation of multilingual addresses. GeoER [2] is another state-of-the-art framework available in literature. However, the solution fails to work for cold start addresses due to insufficient geocode information. This is a major issue in countries where large proportion of addresses have no delivery history. GeoER also assumes the existence of neighbours based on identical names of their POI (Points of Interest). However, for free flowing addresses, neighbourhood learning is a separate challenge in itself. These limitations highlight the need for a more robust modeling approach. To overcome these challenges, we leverage state-of-the-art transformer models which do not have any additional dependencies and rely exclusively on raw address text pairs during inference.

Additionally, lack of quality training data is a perennial problem [22, 41] for EM. Creating a representative training set for pairwise 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. (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) Considering the current trend towards employing language model (LMs) based entity matching [25, 26], utilizing a few thousand samples result in overfitting [45] the LMs. With the aforementioned challenges in mind, to achieve state-of-the-art performance with transformer models, there arises a need for large amount of labeled data. Manually labeling a large volume of address pairs for a variety of scenarios in EM does not scale, hence prior studies has adopted Weak Supervision (WS). The key motivation driving this approach is that it is easier and cheaper to generate large volume of weakly labeled data instead of following the laborious and expensive process of manually annotating each sample with the correct label. By leveraging WS, we have the ability to train and fine-tune the transformer-based models on larger and more diverse datasets, leading to better generalization. Typical forms of WS either rely on domain knowledge [47] or text semantics and embedding distances between attributes or instances [16]. However, these techniques drastically fail to capture geospatial awareness of syntactically similar addresses. For example, addresses *'Palli Nou De Convent 44 , Algemi, Valencia, 46680'* and *'Palle Nou De Convent 18 - 2, Algemi, Valencia, 46680'* have a cosine similarity of 0.99 with FastText [4] embeddings but these two addresses belong to different buildings that are located a few hundred meters apart. As highlighted by the above example and due to the variety of challenges posed by customer addresses, the existing techniques are insufficient and cannot be utilized in their current form to achieve the desired performance.

In this paper, we try to tackle the above discussed challenges by proposing an novel end-to-end framework for matching customer addresses. We leverage transformer-based architecture to prevent efforts spent on manual feature engineering and employ weak supervision with various geospatial signals to leverage historic delivery information to generate diverse and quality labeled data. Our empirical evaluation shows significant improvements in pairwise matching and delivery point (DP) geocoding metrics compared to the existing system and other state-of-the-art baselines. Further, it should be noted that the structure of addresses are quite different for Spain (ES), Brazil (BR), Egypt (EG) and Kingdom of Saudi Arabia (KSA), hence the improvements across all geographies confirm the wide applicability and generic nature of our approach. In summary, our main contributions are as follows.

- We pose address matching as a sequence-pair classification task and fine-tune the pre-trained language model in two stages via weakly labeled and manually curated pairwise data. Our approach is equally effective for multilingual addresses as well.
- Our approach leverages geospatial properties and historic delivery information to create diverse signals and generate pairwise labeled data via weak supervision.
- We did extensive experimentation and ablation studies to show the effectiveness of weak supervision and transformer-based model against other baseline approaches across multiple metrics on four geographies.
- We highlight the positive impact observed via delivery point geocoding metrics, a fundamental business problem that enables delivering packages in a cost-effective manner. These improvements lead to better delivery planning, significant decrease in operation costs, and customer satisfaction.

2 Related Work

We can divide prior literature into three broad categories — rule-based, crowd-based and machine learning (ML) based solutions. Rule-based solutions either rely on pre-defined matching rules such as DNF [1, 19] or dynamically synthesized entity matching rules [38] to find matching pairs. While rule-based solutions have the advantage of being highly interpretable, they can be time and resource-intensive. They often require domain experts to define the rules and may perform poorly on unstructured data [29]. To alleviate the drawbacks of rule-based solutions, crowd-based solutions [14, 43] have been proposed that employ crowd-sourcing workers to manually identify matching tuples. However, such methods are time consuming and human labor cost is extremely expensive which makes them not suitable at Amazon scale.

The state-of-the-art solutions for entity resolution (ER) now predominantly rely on deep learning based approaches. This is mainly attributed to dynamically learn a distributed representation of entities. DeepER [13] represents each attribute as the aggregation of its words' representations using RNN and LSTM [36]. DeepMatcher [29] used a bidirectional RNN with attention mechanism for entity summarization. The decoding steps for both DeepER

and DeepMatcher are similar as both compare attribute representations using various comparison functions such as cosine similarity, element-wise subtraction etc. and then employ a dense layer to get the EM decision. Though DL based solutions [13, 29] have been proven successful in improving the general performance of EM task on structured and semi-structured data, their entity-centric paradigm share some common drawbacks — (1) using only component (entity) representation causes semantic sparsity and information dilution problem, (2) incapable of handling heterogeneous data types across entities and (3) entity assumes some form of inherit structure. Such limitations along with unstructured, free-form and vernacular nature of address text leads to performance degradation of existing models on customer address data.

Pre-trained language models (LMs), such as BERT [12], that have been trained on unsupervised language modeling tasks on massive text corpora have been used for entity matching and achieved better accuracy. Recently, [31] employed a supervised contrastive learning technique to pre-train RoBERTa base model and fine-tuned it for the product matching task across multiple sources using pairwise training data. The authors further proposed a source-aware sampling strategy designed to reduce noise during contrastive pre-training. This sampling strategy is not effective due to the challenges posed by customer addressees, thereby leading to limited gains in performance for matching customer addresses. Ditto [26] casts ER as a sequence-pair classification problem based on fine-tuning pre-trained LMs and obtains the best performance among all the existing supervised ER works. However, it also proposes domain knowledge injection to highlight specific spans of tokens which is not feasible in case of free flowing texts like customer addresses. Given the effectiveness of pre-trained LMs across a variety of EM tasks, we also leveraged transformer-based encoder that adopts a similar sequence pair serialization strategy but without the need of injecting domain knowledge to the model. Li et al. [25] makes use of a Siamese network structure based on BERT, both to speed up the blocking phase and compare candidate pairs in a sequential block. At Amazon's scale, utilizing sequential blocking and matching pipelines is not a viable option as it leads to significant increase in computational costs.

Despite the success language models, one bottleneck for fine-tuning LMs is the requirement of labeled data in large volume. When labeled data are scarce, the fine-tuned models often suffer from degraded performance, and the large number of parameters leads to extreme overfitting [45]. Unsupervised ER approaches [6], [44], [46] are designed to perform ER without labeling. ZeroER [44] learns the match and mismatch distributions based on Gaussian Mixture Models (GMMs). Despite the benefit of zero label requirement, unsupervised approaches are highly error sensitive and may suffer from poor ER results due noise and errors contained in real-world datasets. Among the geospatial ER works, GeoER [2] obtains the best performance among all the existing solutions. However, the solution fails to work for cold start addresses due to insufficient geocode information.

In our previous work [28], we addressed the customer address matching task by employing XGBoost with cost-sensitive learning to model the training data. While that approach utilized active learning combined with graph theory and historical delivery data for data curation, the model's architecture relied heavily on manual feature engineering. This earlier work serves as a baseline for our approach, and we provide a detailed performance comparison between our current approach and the baseline across multiple geographic regions in Table 2

Figure 1: Demonstrates the package distribution across different stop groups. The packages in close proximity within the same building are catered via same stop but packages from different buildings are catered via different stops by the driver.

Due to the variety of challenges posed by customer addresses, the existing techniques in the literature are insufficient and cannot be utilized in their current form.

3 Problem Statement

Let A_1 and A_2 denote a pair of free-form customer 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 $A_1 \times A_2$ becomes too large across customer 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 [17], with neighbourhood embeddings [18] to index the addresses and then filter those addresses that are an obvious no-match (addresses that belong to a different district, state or postal code). We retrieve *top-k* candidates for every address and apply pairwise-matching.

4 Methodology

In this section, we present the details of our proposed method. First, we discuss the preliminaries to geospatial signals. We then show how record pairs are retrieved and labeled using geospatial signals via weak supervision. Finally, we delve into two-stage fine-tuning of pre-trained transformer-based encoder.

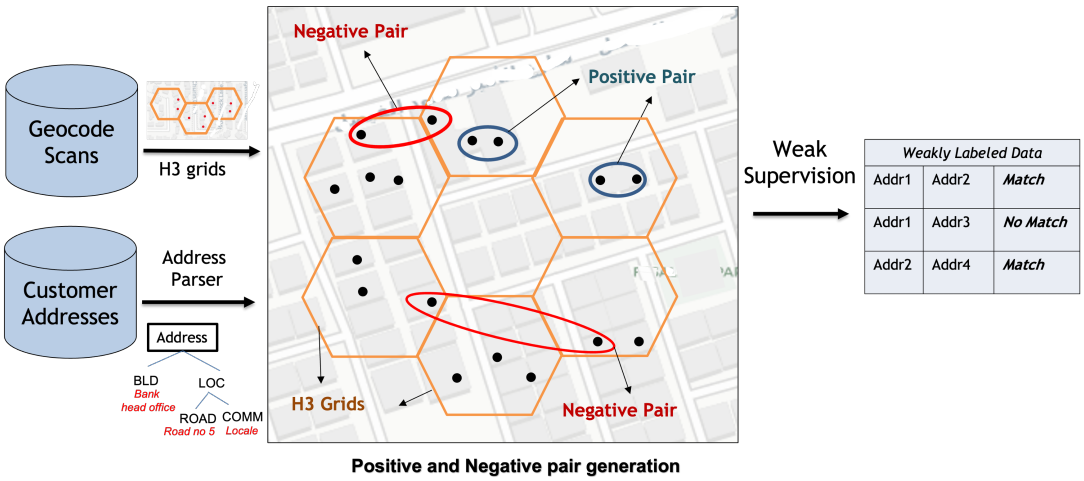

Figure 2: Demonstrates the workflow of proposed weak supervision pipeline. At first, we leverage address text and delivery geocodes as modalities to retrieve parsed components (building, road, etc.) and H3-grids respectively. In second step, we generate positive and negative pairs via Algorithm 1. The orange hexagons represent neighbourhood grids and black dots highlight address entities within each hexagon grids. We apply domain specific heuristics to define labeling functions which is used to determine the weak labels.

4.1 Preliminaries on Geospatial Signals

We describe geospatial signals that are extracted from successful historical deliveries. We use these signals to provide syntactic, linguistic as well as geospatial context.

4.1.1 Customer Addresses. Address text is the primary source of information provided by customers about their location. Customers provide their address in free-form text fields, in addition to city, state and district information available from drop-down. The free-form text contains information like unit or apartment number, street name, locality/sub-locality, landmarks etc. We run our experiments on a sample of addresses for which a successful delivery was made in past few years. Before training, we do basic pre-processing like upper-casing the text, handling white-spaces, punctuations and splitting alphanumeric characters interspersed into each other, for example, *B123,Khalifa CityB* $\rightarrow$ *B 123 , KHALIFA CITY B.* However, we expect our approach to overcome some inconsistencies like misspellings, compound words etc. which are still left after performing the pre-processing.

4.1.2 Geocodes. A geocode denote a pair of latitude and longitude where a delivery associate marks a delivery as complete. Multiple GPS points are available based on successful past deliveries. GPS points are sometimes noisy as it depends on driver compliance. Additionally, in city canyons, where line of sight to GPS satellites is severely limited and in emerging geographies where GPS quality is not up to the mark, the GPS measurement error can be substantial [33]. To overcome these challenges, 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 [15]. We use density-based methods to accurately approximate a single delivery point from historical deliveries for each address via Kernel Density Estimation (KDE) [37]. KDE maximized delivery points are further used to determine the geospatial proximity between a pair of addresses using haversine distance [8]. The haversine distance, also called as great circle distance, is the shortest distance between two points on the surface of a sphere (Earth) with each point denoted by its latitude and longitude.

4.1.3 Stop Groups. Package sharing or stop-consolidation demonstrated in Figure 1 is a logistical approach where customer addresses are grouped by the driver in a single-vehicle stop. These can be individual units inside a multi-apartment building, standalone house with different floors, adjacent blocks within a same building having a common entry, etc. that the driver adheres to together after parking the vehicle. Instead of making individual trips for each stop, the driver combines the addresses that are in close proximity and belongs to the same building. This information gives a sense of proximate relationship between customer addresses and is useful for effective route planning as it allows us to avoid short transits, U-turns and parking costs for future deliveries.

4.2 Geospatially aware Weak Supervision

We leveraged address text along with geospatial signals to generate positive and negative weak labels for address pairs. This weakly labeled pairwise data is used to fine-tune the pre-trained language model. The weak supervision workflow is shown in Figure 2.

Let $Z = \{Addr_1, Addr_2..Addr_n\}$ be a list of unique address in a country C, each with corresponding geospatial attributes like address text A_i, geocodes lat_i, $long_i$, number of deliveries n_i, and stop-consolidation information sc_i. We aim to generate weakly labeled match and no-match address pairs, where each pair $< Addr_i, Addr_j >$, represents a comparison between two addresses from Z. The output is defined as: if $WS_{<Addr_i, Addr_j>}$ refer to same

Table 1: Weak labels generated by our approach. Blue text highlights a road and red text represents a building.

Address 1	Address 2	Weak label
Calle Las Nurwed N° 1 , Frokestisía Inam	C / Las Nurwed Numero 1, Frukistesia inam	Positive
Calle Tan Garnardo 20 Bajo A , Amistad	Calle Cóncatenate N 20 A , Amistad	Hard Negative
Taddeig Sant Creprasi N 6 , Richdort	Tadeo San Creprasio 12º, Richdort	Soft Negative

Algorithm 1: Weak supervision with geospatial signals

Variables : $Addr_i$ - Address entity i, A_i - address text of $Addr_i$, $BldNum_i$ - Building number for $Addr_i$, $Road_i$ - Road info for $Addr_i$, $StopCon_i$ - Stop assigned to $Addr_i$

Input: List of address entities $\{Addr_1, Addr_2 ....Addr_n\}$ with geospatial attributes

Output: Address entity pair $< Addr_x, Addr_y >$ with a match or no match weak label

Filter out addresses with number of deliveries $< d$
For each entity $Addr_i$, assign a $h3$ cluster based on lat_i $long_i$
for *each pair* $< Addr_x, Addr_y >$ *in a single h3 cluster* **do**
  if $Addr_x$ *and* $Addr_y$ *has same* $StopCon_s$ & *haversine* $(lat_x, long_x, lat_y, long_y) <= \beta_1$ & $BldNum_x = BldNum_y$ & $fuzzyRatio(Road_x, Road_y) > \eta_1$ **then**
    Mark $Addr_x$ and $Addr_y$ as a building match
  end
end
/* Hard negative with same building number */
for *each pair* $< Addr_x, Addr_y >$ *in a* ***neighbouring*** *h3 clusters* **do**
  if $fuzzyRatio(A_x, A_y) > \eta_2$ & *haversine* $(lat_x, long_x, lat_y, long_y) >= \beta_2$ & $BldNum_x = BldNum_y$ & $fuzzyRatio(Road_x, Road_y) < \eta_3$ **then**
    Mark $Addr_x$ and $Addr_y$ as a non building match
  end
end
/* Soft negative with similar road info */
for *each pair* $< Addr_x, Addr_y >$ *in a* ***neighbouring*** *h3 clusters* **do**
  if $fuzzyRatio(A_x, A_y) > \eta_2$ & *haversine* $(lat_x, long_x, lat_y, long_y) >= \beta_2$ & $BldNum_x != BldNum_y$ & $fuzzyRatio(Road_x, Road_y) > \eta_3$ **then**
    Mark $Addr_x$ and $Addr_y$ as a non building match
  end
end

physical building we call it *match* (denoted by 1) else *no-match* (denoted by 0).

In addition to basic pre-processing on address text like upper-casing and address specific stop-word removal, we also used a deep learning based parser. The address parser extracts structured chunks of information from each free-form customer address text. For example, given an address text, *'Bukhari St , 8734, Flat no 3307, Rayeha'* the parser extracts components as *Apt: "3307"*, *Bld: "8734"*, *Road: "Bukhari St"*, *Locality: "Rayeha"*. We use stacked BiLSTM+CRF [48], a deep learning architecture for address chunking tasks across all geographies. The parser uses BiLSTM [36] that captures the semantics from free-form text for chunking task and use fastText embeddings for address token representations. The structured components extracted from parser are utilized for creating rules for weak supervision. Note that the address components extracted by the parser are exclusively employed for the weak supervision pipeline only and not used during model fine-training or inference.

In order to mitigate noisy geocoding scans, we consider only those address entities for which atleast d successful deliveries were made in the past. Our objective is to sample positive matches from the addresses that lie relatively closer, while negatives are to be sampled when they lie much farther away. However, the challenge we face is the expensive computation of the haversine distance [8] of each address to every other address. Further, even using some spatial data structure such as Ball Tree [30] involves significant computation overhead. To overcome this, we propose to leverage *H3* based geospatial indexing [35] system as an approximate solution to sample positive and negative addresses in an optimal manner.

H3 is a hierarchical spatial data structure 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. 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 centerpoint 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 (refer to Figure 6 for illustration on this link). 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 an address, T likely positive addresses are sampled from its *H3* grid of level L and T likely negative addresses are sampled from *one-skip* and *two-skip* neighbouring grids of an address. We generate pairs for a few different resolutions as varying resolution helps to generate a more diverse pairwise training data as it can encode very a fine-grained as well as a coarse grained comparison of addresses. The positive and negative pairs are weakly labeled the following manner:

Positive pairs: We limit our sampling to address pairs in the same H3 grid and then further filter the pairs based on haversine distance. We leverage address parser to calculate component similarity scores for address pairs. We then label address pairs with high fuzzy similarity scores for road and building components as *match*. Further, stop groups signal ensure we minimize the number of false positives.(Algorithm 1, line 4-7).
Hard negative pairs: We sample address pairs that are situated in neighbouring H3 grids (as shown in figure 2) as this ensures that the addresses are sufficiently distant and reduce the likelihood of false negatives. We then retrieve only those pairs with high fuzzy similarity scores based on address text as it assists the model in recognizing subtle distinctions. For hard negatives, we ensure the addresses have same building numbers and different road names with certain distance guardrail. (Algorithm 1, line 10-16).
Soft negative pairs: In this case too, we select the address pairs which belong to neighbouring grids but have a high textual similarity between the address texts. Here, we choose the address pairs with different building numbers but same street and mark them as negative. (Algorithm 1, line 18-22). Table 1 highlights the weak labels generated by our approach for different scenarios.

4.3 Model Training

Pre-trained Model: Language models (LMs) based on the Transformer [42] architecture, such as BERT [12] or GPT-2 [34], have achieved state-of-the-art performance for a wide range of natural language processing (NLP) tasks [20, 49]. This success is primarily attributed to the self-attention mechanism used in the architecture. Research studies [10, 40] have revealed that the shallow layers of the model capture lexical meaning, while the deeper layers capture the syntactic and semantic meanings of the input sequence after pre-training. A distinct advantage of pre-trained LMs is their ability to learn the semantics of words more effectively than conventional word embedding techniques, such as word2vec, GloVe [32], or FastText. This is because the transformer architecture calculates token embeddings from all the tokens in the input sequence, enabling it to generate highly contextualized embeddings that capture the semantic and contextual understanding of the words. Therefore, to leverage transformer-based encoder we formulated the problem of address text matching as a sequence-pair classification task and fine-tuned a variation of pretrained BERT [12] — a LM trained on masked language modeling (MLM) and next-sentence prediction (NSP), which outperformed other models in our experiments for address matching.

Two Stage Finetuning: Fine-tuning is a common approach for domain adaptation. During fine-tuning, the pre-trained BERT model is initialized to the pre-trained weights and biases, and all model parameters undergo gradient updates. We proposed a two stage fine-tuning strategy where in the first phase, we fine-tune our off-the-shelf pretrained model on a diverse dataset of weakly labeled pairwise data, generated through our weak supervision pipeline outlined in section 4.2. This stage enables the model to grasp the concept of proximity and spatial relationships within the customer address domain. It becomes even more significant in geographies like EG and KSA where a substantial volume of bilingual address texts exists. Through training on an large dataset

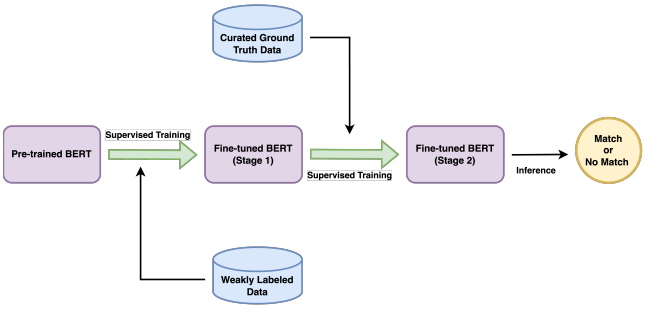

Figure 3: Demonstrates the two-stage workflow of model finetuning. In the first stage, we use tens of thousands of weakly labeled data and in the second stage we used a few hundred thousand manually curated address pairs.

of cross-lingual address pairs, our model acquires the ability to grasp the proximity between English and its Arabic counterpart for ES and SA, thus eliminating the necessity for an additional translation service. Further, we observed that the model tends to overfit the noise of weak labels when fine-tuned only on weakly labeled data. To overcome this limitation, we propose a second stage, where we aim to overcome the limitation posed by noisy labels from the first stage. In the second stage, we initialize our model weights and biases from the artifact obtained from first phase and further fine-tune the model on a small high quality dataset curated by human annotators. This phase prevents the model from overfitting the noise of weak labels by learning from manually curated data. It is also important to note that we don't rely on any geospatial signals for training our language model — a must have functionality for matching cold-start addresses. We first serialise the address pairs,

$$< Addr_1, Addr_2 >= [CLS]Addr_1[SEP]Addr_2[SEP]$$

where $[SEP]$ token separates the two sequences and $[CLS]$ token encodes the address pair into a d_{hidden} dimensional vector. Subsequently, this $[CLS]$ token is fed into a fully connected layer for building-level classification. Our optimization approach involves employing cross-entropy loss for binary classification of building-level matches.

5 Experiments

We did extensive offline experimentation to develop, refine and validate our proposed entity matching framework across multiple geographies (ES, BR, KSA and EG). In this section, we describe experiments that demonstrate the efficacy and effectiveness of our proposed approach.

5.1 Data

5.1.1 Curated Ground Truth (CGT). We did stratified sampling of addresses for each geography to cover all the linguistic and address writing styles, abbreviations across the country. The selection also ensures the inclusion of one postal code with medium address volume and one with low address volume, taking into account the diverse density of addresses, such as the potential urban versus rural/outskirts distribution. We use geolocation of historic deliveries

Table 2: Performance of different state of the models on CGT test set across ES, BR, KSA and EG

Model	Recall@95P				Recall@90P				PR-AUC			
	ES	BR	KSA	EG	ES	BR	KSA	EG	ES	BR	KSA	EG
DeepMatcher	72.17	74.3	37.44	32.58	86.04	89.95	61.49	58.77	92.57	92.88	87.64	85.12
GeoER	85.71	-	-	24.78	90.84	-	-	49.57	94.08	-	-	82.01
Baseline Model	88.57	90.81	53.85	53.17	92.41	95.2	75.01	74.5	97.35	98.06	90.57	90.14
Ditto	90.07	93.75	58.61	69.37	93.46	97.31	77.18	81.38	97.39	98.61	91.29	91.27
MoEBert	59.89	91.79	27.21	36.31	81.22	93.92	65.72	72.93	93.89	97.77	89.01	89.86
Mistral	70.68	78.90	32.45	67.01	-	-	-	-	-	-	-	-
GeoBERT	90.02	92.12	54.76	71.62	87.82	91.59	73.64	78.13	96.39	97.94	90.51	91.05
Our Approach	**91.97**	**97.16**	**67.08**	**82.08**	**96.07**	**98.4**	**81.25**	**88.74**	**97.92**	**98.96**	**93.18**	**94.46**

and cosine similarity of address text to generate 8346, 8651, 15730 and 12371 address pairs for ES, BR, KSA and EG respectively which are manually labeled by our data annotation team.

5.1.2 Weakly Supervised Data. We randomly shuffle the H3 index list and generate around a few million pairs of matches and no-matches for each geography. We then sample an equal proportion of match and no-match pairs to prevent class imbalance.

5.2 Baselines

We compare and evaluate the efficacy of our proposed approach with multiple state-of-the-art baseline techniques.

- **Baseline Model:** Following Comber et al. [11], we first parse the address text using our address parser (discussed in Section 4.2) into address fields (unit, building, road, locality). Further we engineer features, such as cosine similarity and fuzzy match score of record pairs for all the parsed address fields to perform pair-wise matching (binary classification) using the XGBoost [7] classifier. This classifier serves real-time traffic [28] and was trained on manually curated ground truth data along with active learnt address pairs.
- **DeepMatcher:** DeepMatcher [29] is a traditional deep learning solution which uses bidirectional RNNs for attribute values aggregation, attribute comparison, and attention mechanism for attribute soft alignment. It leverages FastText [4] to train the word embeddings.
- **GeoER:** The architecture [2] of this model includes a transformer block, a geocoding block, and a neighborhood block. In order to accommodate the limitations for this model as described in 1, we have modified the code and computed the metrics only for ES. It fails for geographies like BR and KSA due to lack of historic delivery data. For EG, we have evaluated against a sample of test set where the geocode information is available.
- **Ditto:** It casts EM as a sequence-pair classification problem based on fine-tuning pre-trained LMs and obtains one of the best performance among all the existing supervised ER works. Ditto [26] also proposes domain knowledge injection to highlight specific spans of tokens. Unlike product matching, specific spans of tokens are not readily availbale in case of free flowing texts like customer addresses. To make this model work effectively for address matching, we use our address parser (discussed in Section 4.2) to extract structured components as specific token spans.
- **MoEBERT:** We use mixture-of-experts [50] to inject lexical structure of addresses and spatial knowledge with to aim to increase model capacity and inference speed for matching address text.
- **Mistral:** We use Mistral [21] 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 which enables the model to determine if an address pair is match or no-match.
- **GeoBERT:** It integrate semantics and geographic information in the pre-trained representations of POIs [27] by mapping multiple geographic granularity into a unified latent space, which helps obtain the POI embeddings with geographic information.

5.3 Parameter Settings

We have initialized the model with a publicly available pre-trained models using Hugging Face interface for each geography. In case of ES, the Albeto base [5] is used as the base model to initialize address embeddings. In BR, we use a pre-trained Portuguese Bert-base [39] , and for KSA/EG, we use Bilingual Bert [24] for initialization. We fine-tune the *[CLS]* token following two neural network layers with BatchNorm and Dropout enabled. For all the geographies, we use Adam optimizer [23] with an initial learning rate of $3e$-5, dropout of 0.15 and a batch size of 32 for 12 epochs and as required we do resort to early stopping at times to prevent overfitting. The value d for successful historical deliveries is set to 10.

5.4 Evaluation on pairwise data

We split the CGT data in 70-10-20 for training, validation and testing. We compare all the models trained on entire training dataset and evaluated on same test dataset. The validation set is used only to tune the hyperparameters and the test set is held out during both training and validation. To align with downstream applications, a high precision (atleast 95% precision of match class) model is required. We evaluate the model across three metric, namely — Recall@90% precision (Recall@90P), Recall@95% precision (Recall@95P), and Precision-Recall area-under-curve (PR-AUC). Table 2 reports the performance of all the models across multiple metrics on CGT test dataset. The precision-recall numbers are corresponding to the match class (represents same physical building) to align the model performance with the delivery point geocode learning use-case. From Table 2, we observe that our approach

Table 3: Ablation study to demonstrate the impact of different stages of model fine-tuning.

Model	Recall@95P				Recall@90P				PR-AUC			
	ES	BR	KSA	EG	ES	BR	KSA	EG	ES	BR	KSA	EG
Baseline Model	88.57	90.8	53.85	53.17	92.41	95.22	75.0	74.5	97.35	98.06	90.57	90.14
Baseline Model (with WS)	90.78	90.0	56.56	55.42	93.86	94.74	73.33	75.51	97.53	97.84	91.14	90.98
Our approach w/o both stages	6.8	7.9	3.4	2.9	9.89	10.1	5.7	4.89	32.13	50.58	28.84	16.67
Our approach w/o stage-1	88.69	94.05	59.1	70.45	92.6	97.31	76.18	80.7	96.39	97.67	91.53	90.74
Our approach w/o stage-2	76.89	91.58	55.78	66.96	91.6	97.95	71.11	76.23	97.33	97.54	67.17	55.57
Our Approach	**91.97**	**97.16**	**65.41**	**76.45**	**96.07**	**98.4**	**80.41**	**85.2**	**97.92**	**98.96**	**93.18**	**94.46**

outperforms the baseline significantly. Overall on an average, our approach shows an improvement of *15.57%* on Recall@95P, *6.84%* on Recall@90P and *2.23%* increase in PR-AUC across four geographies. We further observe improvement against existing state-of-the-art baselines, for example, Ditto.

6 Real world application of Customer Address Matching

Address matching is a fundamental problem to be solved in the domain of delivery planning and route optimization. Real-time matching and organizing existing addresses into a hierarchy are two workflows which can help build a knowledge graph to define relationships between places in a geography. Accurate address matching is crucial to enable both of them. Multiple learning-based systems depend on these matching solutions. In this paper, we limit the scope of pairwise address matching for delivery point geocoding, which we describe below.

6.1 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. For this paper, we limit the scope to DP geocode learning for cold-start addresses. Following are some of the key metrics used to measure the quality of DP geocodes. **Delivery Precision** is the percentage of total shipments for which the actual delivery happened within a threshold distance $\mathcal{Z}$ from the planned delivery location. **Delivery Defects** is the percentage of total shipments for which the actual delivery happened outside of the threshold distance $\mathcal{Y}$ from the planned delivery location. Hence, lower the value of outliers, better the metric. For business reasons, we cannot reveal the actual values of $\mathcal{Z}$ and $\mathcal{Y}$.

Having sufficient number of deliveries to an address allows us to learn reasonable quality geocodes by aggregating the past delivery locations [15]. 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 then aggregate the geocodes of all matched addresses to learn a single DP geocode using Kernel Density Estimation (KDE) [37]. The equation 1 below formulates the kernel density estimator P over the retrieved matched addresses M where $K(x;h)$ is a Gaussian kernel with haversine distance metric. The bandwidth h works as a smoothing parameter, and we chose h as 25 meters after performing validations manually validation over a range of values.

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

6.2 Offline Evaluation and Impact

We perform offline evaluation and observe the impact on DP geocoding across actual set of deliveries done to all the cold-start addresses. The deliveries that happened from January 2021 to December 2022 across all delivery stations were considered for creating the reference set with known addresses. Deliveries across the span of first two weeks of April 2023 were used as test set. During the test period, a few hundred thousand deliveries were done on cold-start addresses for each geography against which we evaluated our approach. We observed significant improvement in both the DP geocoding metrics compared to the baseline model. We observe *6.57%* improvement in delivery precision and *10.82%* reduction in delivery defects compared to the model which was serving traffic in baseline. These improvements lead to better delivery planning, route optimization, significant reduction in operation costs and customer satisfaction.

6.3 Online Experiment

After observing significant improvements during offline evaluation, we launched an online A/B experiment on live traffic for EG and KSA geographies. We will adhere to the online experimentation setup that we followed to launch our current baseline model [28]. We are performing the model dial-up in a phased manner — 10%, 50%, *and* 100% delivery stations. We move to next stage only if statistically significant improvements are observed during one week of dial-up in each phase. During the A/B test period, our model has learned DP geocodes for a few hundred thousand shipments, where we observed 8.09% improvement in delivery precision and 11.78% reduction in delivery defects. Following the success in two geographies, we will launch an online experiment in other geographies as well.

Table 4: Demonstrates the quality of predictions of baseline model and our proposed approach against the ground truth.

Address1	Address2	Baseline Model	Our Approach	Ground Truth
Mario D Azzo Street , 12345 Tower , Ap 1234	Rua Mario D Azzo, 12345 , Apto 4321 Mariym Bolo	No Match	Match	Match
Estrada tre dacheisão qefueno 666 , casa 50 campo venti	Estrada Tre Dacheisão Qefueno 666, Casa 78 Campo Venti	Match	No Match	No Match
Kareem Holdings , Behind Omni Book Store Podosa, Navi Training & Education Center, Head Office - 7 Th Floor	Podosa Street, Backside Omni Book Store, Al - Navvi Training Building , 5 Th Floor	No Match	Match	Match
5561 Cultivated Area , Unit 9 , 12345 - 9867 , Villa 262 Psv Security Systems , Private Area	Villa 420 Psv security Systems , 5561 - Cultivated Area , Unit 9 , 12345 - 9867 , Private Area	Match	No Match	No Match
Paspal Maritino, 121, Entrance Reception Hotel, Palomao Area	Aventedot Gabrro Rocco 121, Hotel Melia Palma Marina Main Reception, Palomao	No Match	Match	Match

Figure 4: Comparison in quality of delivery point geocode predictions of models against actual delivery locations

7 Analysis

7.1 Quantitative Analysis

7.1.1 Ablation Studies. In order to study the importance of different elements of our approach, we present an ablation study to demonstrate the effectiveness of various components involved in our proposed framework. We aim to highlight the importance of transformer encoder and different phases of finetuning via this study. In Table 3, we show how removing each of these components impact the performance on CGT test data across multiple metrics on four geographies.

7.1.2 Multilingual Addresses. In the EG and SA geographies, a large proportion of addresses contain Arabic tokens. Since our approach is invariant to both English and Arabic addresses, we have used the pairwise address texts in CGT for our evaluation. However, for the baseline and other approaches shown in Table 2, we introduced an additional preprocessing step where Arabic addresses were translated to English using AWS Translate. This was done to facilitate a direct, head-to-head evaluation across all methods.

7.1.3 Single vs Separate models for EG and KSA. Due to a significant volume of Arabic addresses in both EG and SA, we used the same pre-trained model [24]. Considering the similarity in data distribution (English and Arabic), we aim to optimize the training process where instead of training separate models for EG and SA, we merge the individual training sets from both geographies and subsequently fine-tune the model using this consolidated training dataset. We observed improvements across pairwise matching as Recall@95Precision improved by 2.3%, on average, across EG and SA. Further, creating a single model that generalize for multiple geographies reduces deployment overhead as well.

7.2 Qualitative Analysis

7.2.1 Impact on pairwise matching. We observe the quality of predictions generated by our model against the baseline model for pairwise matching. The outcomes of the baseline model and our approach are shown in Table 4. It is evident from these examples that our approach handles false positives and false negatives more effectively. We could clearly observe that our model is able to pay attention to specific nuances, for example, same campus-different building, same building number-different road, colloquial road names, etc. We have highlighted the building level information in the address text for better understanding.

7.2.2 Impact on delivery point geocode learning. Further, we analysed the delivery point (DP) geocodes correctly predicted by our model against the actual delivery location which the baseline model failed to predict correctly. The quality of predictions is highlighted through the following real world scenario. *'Calle Tusa Lutenbergo, 30 T4 1º A, Segunda De, Granada'* is a newly created address. Figure 4 demonstrate that the Baseline model learns an incorrect DP geocode (blue marker), far away from the actual delivery location (black marker). As a consequence, this results in a delivery defect. Our model identifies nearby addresses (yellow points) by matching

the query address and learns an accurate DP (green marker) within the required threshold of the actual delivery location.

7.2.3 Latency. We assessed the latency of our approach with Baseline, GeoER, and Mistral models. To evaluate on a common ground, the interface 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, 2-times, 3-times and 8-times respectively, thus requiring significantly more hardware to reach the same TPS (transactions per second).

8 Conclusion

In this paper we proposed a novel end-to-end transformer based matching framework with weak supervision for customer address matching. Further, we leverage various geospatial signals to generate large volume of weakly labeled data that improved our models performance. Our experimental findings demonstrate the effectiveness and generic nature of our approach, as it consistently achieves superior matching performance compared to the baselines across four different geographies. We also observed significant improvement in delivery precision and reduction in delivery defects for geocode learning systems. These improvements lead to better delivery planning, significant decrease in operation costs, and customer satisfaction.

9 Future Work

As next steps, we plan to experiment with noise-aware loss functions to train accurate pairwise-matching models. Such a loss function will help us to train a robust model and mitigate the possible impact noise from weak labels. Further, a few recent studies have highlighted the benefits of using a generative model to learn weak labels instead of manually engineered rules based on domain knowledge. As a future research direction, we will experiment with such techniques for our problem space.

References

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

[2] Pasquale Balsebre, Dezhong Yao, Gao Cong, and Zhen Hai. 2022. Geospatial Entity Resolution. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3485447.3512026

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

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146. https://doi.org/10.1162/tacl_a_00051

[5] José Cañete, Sebastián Donoso, Felipe Bravo-Marquez, Andrés Carvallo, and Vladimir Araujo. 2022. ALBETO and DistilBETO: Lightweight Spanish Language Models. In *Proceedings of the 13th Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France.

[6] Riccardo Cappuzzo and Paolo Papotti. 2020. Creating Embeddings of Heterogeneous Relational Datasets for Data Integration Tasks. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 1335–1349.

[7] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[8] 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 (2013), 298–302.

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

[10] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look At? An Analysis of BERT's Attention. *CoRR* abs/1906.04341 (2019). arXiv:1906.04341 http://arxiv.org/abs/1906.04341

[11] 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 (2019), 334–348.

[12] 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 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[13] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment* 11, 11 (2018), 1454–1467.

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

[15] 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*. Springer, 403–419.

[16] Congcong Ge, Pengfei Wang, Lu Chen, Xiaoze Liu, Baihua Zheng, and Yunjun Gao. 2021. CollaborER: A Self-supervised Entity Resolution Framework Using Multi-features Collaboration. *CoRR* abs/2108.08090 (2021). arXiv:2108.08090 https://arxiv.org/abs/2108.08090

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

[18] 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*. Association for Computational Linguistics, Abu Dhabi, UAE, 322–331. https://aclanthology.org/2022.emnlp-industry.33

[19] Mauricio A Hernández and Salvatore J Stolfo. 1995. The merge/purge problem for large databases. *ACM Sigmod Record* 24, 2 (1995), 127–138.

[20] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. arXiv:2003.11080 [cs.CL]

[21] 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* (2023).

[22] 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* (2019).

[23] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[24] Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. GigaBERT: Zero-shot Transfer Learning from English to Arabic. In *Proceedings of The 2020 Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

[25] 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*.

[26] 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 (2020). arXiv:2004.00584 https://arxiv.org/abs/2004.00584

[27] 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*. 2209–2214.

[28] 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*. 457–463.

[29] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai 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*. 19–34.

[30] Stephen M Omohundro. 1989. Five balltree construction algorithms. Technical report.

[31] Ralph Peeters and Christian Bizer. 2022. Supervised contrastive learning for product matching. In *Companion Proceedings of the Web Conference 2022*. 248–251.

[32] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation, In Glove: Global Vectors for Word Representation. *EMNLP* 14, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[33] Vamsi Krishna Penumadu, Nitesh Methani, and Saurabh Sohoney. 2022. Learning geospatially aware place embeddings via weak-supervision. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*.

1–10.

[34] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[35] Kevin Sahr. 2019. Central place indexing: Hierarchical linear indexing systems for mixed-aperture hexagonal discrete global grid systems. *Cartographica: The International Journal for Geographic Information and Geovisualization* 54, 1 (2019), 16–29.

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

[37] David W Scott. 1992. Multivariate density estimation: Theory, practice and visualisation. John Willey and Sons. *Inc., New York* (1992).

[38] 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 (2017), 189–202.

[39] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2020. BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*.

[40] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. *CoRR* abs/1905.05950 (2019). arXiv:1905.05950 http://arxiv.org/abs/1905.05950

[41] 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 (2018). arXiv:1809.11084 http://arxiv.org/abs/1809.11084

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR* abs/1706.03762 (2017). arXiv:1706.03762 http://arxiv.org/abs/1706.03762

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

[44] Renzhi Wu, Sanya Chaba, Saurabh Sawlani, Xu Chu, and Saravanan Thirumuruganathan. 2019. AutoER: Automated Entity Resolution using Generative Modelling. *CoRR* abs/1908.06049 (2019). arXiv:1908.06049 http://arxiv.org/abs/1908.06049

[45] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. Unsupervised Data Augmentation. *CoRR* abs/1904.12848 (2019). arXiv:1904.12848 http://arxiv.org/abs/1904.12848

[46] Dongxiang Zhang, Dongsheng Li, Long Guo, and Kian-Lee Tan. 2022. Unsupervised Entity Resolution With Blocking and Graph Algorithms. *IEEE Transactions on Knowledge and Data Engineering* 34, 3 (2022), 1501–1515. https://doi.org/10.1109/TKDE.2020.2991063

[47] Sheng Zhang, Hao Cheng, Shikhar Vashishth, Cliff Wong, Jinfeng Xiao, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Knowledge-Rich Self-Supervised Entity Linking. *CoRR* abs/2112.07887 (2021). arXiv:2112.07887 https://arxiv.org/abs/2112.07887

[48] Yuan Zhang, Hongshen Chen, Yihong Zhao, Qun Liu, and Dawei Yin. 2018. Learning Tag Dependencies for Sequence Tagging.. In *IJCAI*. 4581–4587.

[49] Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware BERT for Language Understanding. arXiv:1909.02209 [cs.CL]

[50] Simiao Zuo, Qingru Zhang, Chen Liang, Pengcheng He, Tuo Zhao, and Weizhu Chen. 2022. Moebert: from bert to mixture-of-experts via importance-guided adaptation. *arXiv preprint arXiv:2204.07675* (2022).