Geo-Spatially Informed Models for Geocoding Unstructured Addresses

Uddeshya Singh*
Indian Institute of Technology Bombay
Mumbai, Maharashtra, India
ud.uddeshya16@gmail.com

Ravi Shankar Devanapalli
Flipkart
Bangalore, Karnataka, India
devanapalli.ravi@flipkart.com

Gowtham Bellala
Flipkart
Bangalore, Karnataka, India
b.gowtham@flipkart.com

Vikas Goel
Flipkart
Bangalore, Karnataka, India
vikas.goel@flipkart.com

Abstract

Geocoding customer addresses and determining precise locations is a crucial component for any e-commerce company. Shipment delivery costs make up a significant portion of overall expenses, and having exact customer locations not only improves operational efficiency but also reduces costs and enhances the customer experience. While state-of-the-art geocoding systems are well-suited for developed countries with structured city layouts and high-quality reference corpora, they are less effective in developing countries like India, where addresses are highly unstructured and reliable reference data is scarce. Recent research has focused on creating geocoding systems tailored for developing nations such as India. In this work, we propose a method to geocode addresses in such environments. We explored various approaches to incorporate geo-spatial relationships using an LLM backbone, which provided insights into how the model learns these relationships both explicitly and implicitly. Our proposed approach outperforms the current state-of-the-art system by 20% in drift accuracy within 100 meters, and the state-of-the-art commercial system by 54%. This has a potential to reduce the incorrect delivery hub assignments by 8% which leads to significant customer experience improvements and business savings.

1 Introduction

Accurate customer location is a critical component for an e-commerce company for efficient delivery of the shipments. It plays a key role in delivering the shipments on time while optimizing for the shipping cost. Some of the key applications in a e-commerce company are the delivery hub assignment and fake attempt prevention. Delivery Hub (DH) is the last mile hub in a shipment's journey from where the shipment is delivered to the customer by a Delivery partner (DP). Every DH has a geo-serviceable boundary, and the customer geo-location is used to determine the DH that the shipment must be assigned to. An incorrect geo-location will lead to the assignment of the shipment to the wrong DH resulting in a shipment misroute. Misrouted shipments will require re-routing leading to promise time breaches, poor customer experience and additional shipping cost. Having accurate geo-coordinates is hence very critical for this application.

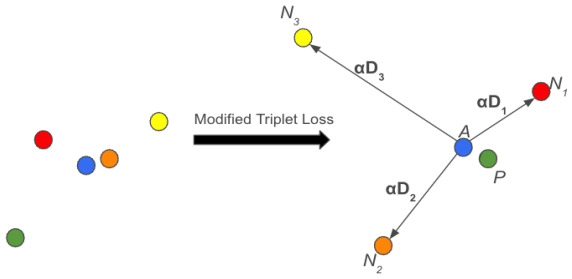


Figure 1: Illustration of the Modified Triplet Loss effect on the latent space: Before training (left), samples are dispersed without structure. After training (right), negative samples are separated from the anchor point proportionally to their distance, scaled by αD_i.

Another major application is DP fake delivery attempt prevention. DPs can mark a shipment as undelivered if the customer is unavailable at the time of delivery. However, at times, DPs abuse this feature and mark the shipments as undelivered without making a genuine delivery attempt. Having an accurate customer geo-location can help us detect and prevent such fake attempts that often lead to a poor customer experience. These are a few examples that highlights the importance of having precise customer geo-coordinates.

In developing countries such as India, customer location information is typically provided as an address text, which poses challenges for direct use in supply chain operations. To overcome this, geo-

*Work done during internship at Flipkart Internet Pvt. Ltd.

Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, pages 236–242
January 19–24, 2025.

coordinates (latitude & longitude) are extracted from the address text, a process known as Geocoding. While various geocoding systems have been developed, most assume structured addresses and are tailored for developed countries. These systems are less effective in developing countries like India, where addresses are often unstructured, unordered, prone to missing or incorrect tokens, along with numerous spelling errors. Some of these challenges are discussed in detail in (Kothari and Sohoney, 2022) and (Srivastava et al., 2020).

Recent work has focused on building geocoding systems specifically for developing countries. SAGEL (Chatterjee et al., 2016) and GeoCloud (Srivastava et al., 2020) are few such systems which are discussed in detail in Section 2. A recent work by (Kothari and Sohoney, 2022) introduced a triplet loss-based approach using RoBERTa for geocoding in a similar geographical context as ours, which is currently considered state-of-the-art for developing countries. We replicated this method but found that it under performed compared to our existing production system—a simple text classification model using fastText. The (Kothari and Sohoney, 2022) approach relies on coordinates recorded by delivery partners and uses a weakly supervised framework based on triplet loss. This raises the question: why approach geocoding as a weakly supervised task when a fully supervised framework might be more effective? To address this, we explored fully supervised techniques for geocoding.

In addition, while most classification problems assume independent target labels, geocoding inherently involves geo-spatial relationships between the labels (H3, 2020). We leveraged these relationships to enhance both weakly supervised and fully supervised approaches.

In summary, our main contributions are: 1) the exploration of fully supervised techniques for geocoding, and 2) the incorporation of geo-spatial relationships between target labels. The remainder of the paper is organized as follows: Section 2 reviews relevant literature, Section 3 discusses the data and Section 4 details the existing production system. In Section 5, we present our approaches while in Section 6 we discuss the experiments and the results and we conclude in Section 7.

2 Related Work

Berkhin et al. (2015) present an approach called Bing GC for geocoding. They frame the geocoding task as an information retrieval problem. They split the entire Earth's surface into overlapping rectangular tiles and leverage traditional web search technologies to retrieve matching tiles with the geocoding query. They use geo-entities associated with each tile to match with the query. Our approach is similar in dividing the region into tiles, but we do not presume access to tile's actual geo-entities.

Chatterjee et al. (2016) present a geocoding engine called SAGEL for geocoding Indian addresses. They use high quality structured address corpus (from a commercial map data provider) as their address database. They pre-process the address query and retrieve matching address documents from the address corpus. The candidates are ranked using graph techniques and the geo-coordinates of the top ranked document is returned. However, structured high quality address corpus is limited and expensive as well. We use SAGEL as one of our baselines.

Srivastava et al. (2020) propose a method called GeoCloud for geocoding unstructured addresses. They parse the entire address corpus and create a geo-polygon for each address chunk using the historical delivered data. However, they use heavy domain knowledge in designing heuristics for parsing the address into chunks and creating a geo polygon, which is not a scalable approach and limits model re-training capabilities.

Kothari and Sohoney (2022) propose a framework to resolve the addresses to a shallower granularity. They propose a weakly supervised deep metric learning model to encode the geospatial semantics in address embeddings and then search for top-k nearest neighbours and retrieves the geo-coordinates from them. This is currently the state-of-the-art system and we modify this approach to further improve the performance.

3 Data Description

Available Data: During order placement, customers provide a shipment delivery address which contains the following fields: (i) Customer Address (a free-text field entered by the customer primarily consisting of granular information like building name, sub-locality, locality), (ii) Pincode, (iii) City, (iv) State. As mentioned in Section 1, there are various challenges associated with this address text. In addition, for every historical shipment, we have the DP (Delivery Partner) captured geo-coordinates at the time of delivery. However, there could be

noise in the DP captured location due to manual errors, GPS errors, network issues, etc. In spite of the noise in this data, it serves as a critical piece of information for our modeling.

Dataset Generation: We have millions of data from our historical deliveries. Since there is noise in the delivered data, we cannot straight away use them. For every address, we chose the mediod of its deliveries as a single geo-coordinate for that address. We split the dataset into train, validation and test as below. To have high confidence on the test set, we chose the addresses that have at least 20 historical deliveries. The intuition is that, if we have high number of deliveries, then most of them would be around the actual location and thus the mediod will be very close to the actual location. We split our dataset into training, validation, and test sets based on delivery frequency: the training set includes addresses with fewer than 15 deliveries, the validation set includes addresses with 15 to 20 deliveries, and the test set includes addresses with more than 20 deliveries.

4 Existing Production System

The existing production system uses the customer address text and its corresponding delivered coordinates to build a geocoding model. A geographical region is divided into hexagonal grids of resolution 10 having an edge length of 75m using the H3 library. H3 (2020) is a open source library built by Uber that divides the entire earth into hexagonal grids at various resolutions. For an address, we retrieve a grid ID using its delivered geo-coordinates. Thus we generate the <address text, grid ID> mapping data using the historical delivered data. A supervised fastText model is trained with address text as input and grid ID as target. At the inference time, the model predicts a grid ID for the given address and return its centroid coordinates as the predicted coordinates.

For the production system model, fastText (Joulin et al., 2017) is chosen because of the following advantages. The training duration to be orders of magnitude faster than the other methods. It learns embeddings at sub-word level which helps with spell errors. Also, since in our production system, one model is trained per pincode and as there are large number of pincodes, it needed a model which not only trains fast but also requires less memory. FastText has a compression module (Joulin et al., 2016) that allows us to reduce model sizes with minimal impact on performance.

However, fastText generates static embeddings and does not account for context unlike the recent state-of-the-art approaches such as BERT. Hence one focus area of our work is to explore more sophisticated embedding architectures. Also, in a typical classification approach, the target classes are fairly independent. However, in our task, the target labels have a geo-spatial relation. Some of the grids are nearby and some are far-away. In the current system, the only geo-spatial information that is used is in the design choice of model by limiting it to a pincode. We wanted to embed this geo-spatial relation as part of the model training as well. The work in (Kothari and Sohoney, 2022) does something similar through contrastive learning approach. We begin by expanding this work further, which we discuss in detail in next section.

5 Methodology

In this work, we initially attempted to improve the existing state-of-the-art (SOTA) method from (Kothari and Sohoney, 2022), which uses a triplet loss-based approach for geocoding. Our initial focus was on enhancing the model's ability to incorporate geo-spatial relationships more effectively, starting with improvements to the loss function. Following that, we explored alternative methods, moving beyond weakly supervised contrastive learning, by experimenting with fully supervised frameworks. These methods not only demonstrated better performance but also provided insights into how large language models (LLMs) capture geospatial relationships when explicitly guided, compared to relying on implicit learning.

5.1 RoBERTa Address

We began by pre-training the RoBERTa model (Sanh et al., 2019) on an address-specific corpus using the masked language model (MLM) objective similar to (Kothari and Sohoney, 2022) approach. Given that address structures differ significantly from general English, we also retrained the tokenizer to better capture the nuances of the address data. This pre-trained model serves as the common base for all subsequent approaches discussed in further sections.

5.2 Weakly Supervised Contrastive Learning

The original triplet loss-based approach from (Kothari and Sohoney, 2022) samples T negative

addresses from the ring of 1-skip neighboring grids at the parent level $(L-1)$. Triplets are generated by varying the grid resolution ($L \in \{11, 10, 9\}$) for both positive and negative samples. However, we hypothesized that this approach does not fully capture the geo-spatial relationships between samples for two key reasons:

1. Anchor-positive pairs in one resolution (e.g., resolution 9) may be treated as anchor-negative pairs in another resolution (e.g., resolution 11), potentially confusing the model.

2. The original approach treats all negative samples equally within a given resolution, without considering their varying distances from the anchor. This limits the model's ability to effectively differentiate between geospatially close and distant negatives.

To address these issues, we modified the sampling strategy by selecting D_k negative samples from grids up to Parent's K-skip neighboring grids away from the anchor, rather than relying solely on the immediate parent level's neighboring grids. This adjustment ensures that multiple negative samples are drawn from varying spatial distances as shown in Figure 2b.

We then modified the triplet loss function to incorporate spatial information by scaling the margin α based on the relative distance of each negative sample from the anchor. This ensures that negative samples farther from the anchor are pushed away more aggressively in the latent space, while allowing relatively closer negative samples (like N_1) to remain closer in comparison to N_2 and N_3 as shown in Figure 1. The relationship is formalized in the modified loss function, as shown in Equation 1:

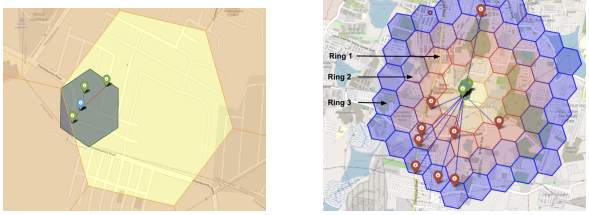
$$\mathcal{L}(A, P, N, D) = \sum_{i=1}^{N} \Big[\|f(A_i) - f(P_i)\|_2^2 - \|f(A_i) - f(N_i)\|_2^2 + \alpha \cdot D_i \Big]_+ \quad (1)$$

Where:

- A_i: The anchor sample for the i-th triplet.
- P_i: The positive sample (within the same grid cell as the anchor) for the i-th triplet.
- N_i: The negative sample (outside the grid cell of the anchor) for the i-th triplet.
- D_i: The ring level distance of the negative sample N_i from the anchor A_i, calculated based on the i-skip parent neighbors in the H3 grid hierarchy.
- α: A scaling factor that adjusts the margin.
- $f(x)$: The embedding function that maps a sample x into a latent embedding space.

This modified loss function helps the model incorporate spatial hierarchy, improving its ability to distinguish between geo-spatially close and distant locations. The model is trained with this modified loss function as shown in Figure 3.

As demonstrated in Table 1, the original RoBERTa-Triplet approach (Kothari and Sohoney, 2022) shows significant performance improvements over the RoBERTa-Address model. Furthermore, our modified triplet loss function led to additional performance gains. The modified RoBERTa-Triplet model showed a clear improvement across all metrics, further validating the benefits of incorporating spatial hierarchy in the triplet loss. Despite these enhancements, the triplet loss-based method still underperformed when compared to the fully supervised framework, which we detail in the following sections.



(a) Positive Sample Generation

(b) Negative Sample Generation

Figure 2: Left: Positive sampling (blue anchor, green positives). Right: Negative sampling (red negatives). Red, purple, and blue rings denote 1, 2, and 3-skip parent's neighbors, respectively.

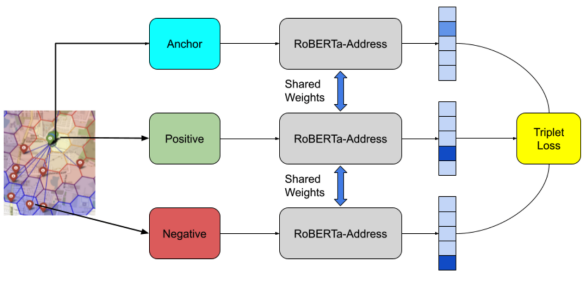


Figure 3: Contrastive Learning Model Architecture

5.3 Supervised Classification

In our initial exploration of the triplet loss-based approach, we found that while it forces the model to capture geo-spatial relationships as shown in fig 5, it did not perform satisfactorily in the downstream task of geocoding (refer to Table 1). This led us to question whether contrastive learning is the only way to embed geo-spatial relationships, or if alternative supervised approaches could capture also this spatial structure. In this section, we explore different supervised learning techniques.

5.3.1 Plain Classification Task

In this approach, we fine-tuned the pre-trained RoBERTa-Address model on a dataset of address-text and grid-ID pairs. The model was tasked with classifying an address to its corresponding grid ID, which are treated as independent and do not inherently share any geo-spatial relationships. As a result, the model learns geo-spatial relationships implicitly from the structured labels unlike the triplet loss approach, which explicitly embeds spatial relationships.

5.3.2 Multi-Head Classification

We trained a multi-head classification model with a shared RoBERTa base and separate classification heads for each of the selected N resolutions. In the H3 grid structure, each grid at resolution R is subdivided into 7 child grids at resolution $R + 1$. This hierarchical structure enables the shared layers to capture common address features, while each classification head learns geo-spatial relationships specific to its resolution. The model architecture is as shown in Figure 4. This approach offers several advantages:

- Separate classification heads allow the model to address both detailed and broader geo-spatial distinctions, making it suitable for tasks that require high precision for close distances and more generalized predictions for larger areas.
- A shared RoBERTa base across all resolutions facilitates learning of geo-spatial correspondences between different resolutions, enhancing the model's ability to generalize across varying levels of detail.

6 Experiments & Results

We evaluated both the contrastive and supervised approaches across several Indian states, using a single model per state rather than training separate models for each pincode, as is done in the production system. This approach reduces the maintenance overhead and is particularly advantageous in addressing issues related to incorrect pincodes, discussed further in Section 6.4.

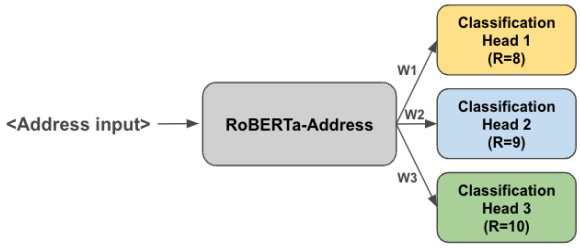


Figure 4: Multi-Head Model Architecture

6.1 Model Training

As described in Section 5, we initialized the model with the pre-trained RoBERTa-Address and trained it using triplet pairs generated per state. RoBERTa-Triplet (Original) model was trained following the approach of (Kothari and Sohoney, 2022), using triplet pairs across multiple resolutions $R = 8, 9, 10$. RoBERTa-Triplet (Modified) however focused specifically on resolution $R = 10$. For each state, millions of triplet pairs were created, selecting D_k negative samples from grids up to the parent's K-skip neighboring grids, where K ranges from 1 to 3. The triplet loss function was adjusted by scaling the margin $\alpha = 5$ based on the relative distance D_k of each negative sample. During inference for both models, approximate nearest neighbor (ANN) search was used to find the top-8 similar addresses, with the medoid of these neighbors serving as the predicted coordinates.

For the supervised classification tasks, including both the plain and multi-head models, each state provided millions of training data points. In the single-head setup, the model was trained with target labels at resolution $R = 10$. For the multi-head approach, the model utilized three classification heads, corresponding to resolutions $R = 8$, $R = 9$, and $R = 10$. These levels were chosen to balance computational efficiency and model performance. Using finer resolutions, such as $R = 11$, would seem like a natural extension. We also experimented at such finer resolutions; however, the performance has degraded. There could be two potential reasons for this, one is the GPS noise and the second one is the large number of target classes.

The hexagonal grid size at resolution 11 is around 28 meters, which is highly sensitive to GPS noise. There is inherent noise in the GPS signal, which is usually in a few 10's of meters. Hence even if the FE rightly captured the customer location, due to the GPS noise, it might get tagged as wrong grid-id. Added to this noise, the number of label classes also increases significantly (5x to 7x). Because of this large number of classes and noise in the grid-id labels, the model performance has degraded

6.2 Performance Comparison

The metric that we use for the comparisons is the "drift accuracy". Drift represents the great circle distance between the predicted and the actual coordinates. Drift accuracy at 100 meters represents, out of 100 given addresses, how many addresses have drift less than 100 meters. Table 1 summarizes the performance of various models, including baseline comparisons with SAGEL (Chatterjee et al., 2016), the Google Maps API (Google, 2020), and pre-trained models like RoBERTa-English and RoBERTa-Address. The RoBERTa-Address model, pre-trained on address-specific data, showed improvements over the generic RoBERTa-English due to its domain-specific pre-training.

For contrastive learning models, the RoBERTa-Triplet (Modified) model, which focused specifically on resolution $R = 10$ and incorporated a refined sampling strategy with distance-based margin adjustment, outperformed the RoBERTa-Triplet (Original) model that used triplet pairs across multiple resolutions ($R = 8, 9, 10$). The improvement in the modified version demonstrates the effectiveness of incorporating spatial information through adjusted negative sampling. However, despite these enhancements, the triplet-based methods still lagged behind the fully supervised approaches.

Among the supervised methods, the Plain Classification model trained at resolution $R = 10$ outperformed both the triplet-based models and the existing production system. The Multi-Head model provided further gains in accuracy, showcasing the benefits of capturing geo-spatial relationships at different levels of detail.

The best performing model improved the drift accuracy by 12.9% within 100m and 2.8% within 500m. This leads to 8% reduction in incorrect DH assignment and 7% reduction in fake delivery attempts.

Method	< 100m	<500m	<1000m	<2000m
Production	64.3%	88.4%	92.4%	94.8%
SAGEL	17.7%	38.9%	49.9%	68.8%
Google	23.8%	59.1%	73.1%	83.0%
RoBERTa-English	21.5%	45.0%	53.4%	61.0%
RoBERTa-Address	24.1%	51.1%	60.4%	67.0%
RoBERTa-Triplet (Original)	56.7%	73.4%	75.6%	76.9%
RoBERTa-Triplet (Modified)	65.7%	83.1%	85.1%	86.1%
Classification	72.4%	90.6%	93.1%	94.4%
Multi-Head	**77.2%**	**91.2%**	**93.3%**	**94.6%**

Table 1: Drift Accuracy Comparison of different models.

6.3 Qualitative Analysis

Figure 5 shows t-SNE visualizations of embeddings from various models. In these plots, clusters of the same color represent addresses that fall within the same grid, with each point indicating an individual address. RoBERTa-Address forms more distinct clusters than RoBERTa-English, reflecting the advantages of pre-training on address data. The RoBERTa-Triplet model, trained with its contrastive approach, produces tighter clusters, effectively capturing geospatial relations. Interestingly, the Classification model, despite treating grid IDs as independent labels, achieves nearly comparable clustering, suggesting that it can infer spatial relationships even without explicit guidance.

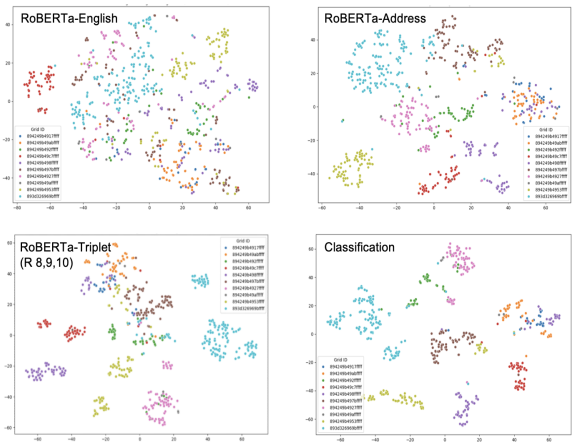


Figure 5: t-SNE visualization of embeddings from various models.

6.4 Handling Incorrect Pincodes

In real-world applications, particularly in India, users frequently provide incorrect pincodes, which can negatively impact geocoding accuracy and increase delivery delays. To evaluate this, we created a synthetic dataset by randomly altering pincodes to simulate real-world errors. Table 2 shows the performance comparison. The production system,

where each model is tied to a specific pincode, suffers from a significant drop in accuracy when incorrect pincodes are provided. In contrast, our approach, which does not rely on pincodes as input, remains robust in such scenarios.

		< 100m	<500m	<1000m	<2000m
Actual Pincode	Production	64.3%	88.4%	92.4%	94.8%
	Our Method	78.9%	92.1%	94.0%	95.2%
Incorrect Pincode	Production	46.7%	70.9%	76.4%	80.1%
	Our Method	78.9%	92.1%	94.0%	95.2%

Table 2: Performance comparison of models with incorrect pincodes.

7 Conclusion & Next Steps

To conclude, we began our experiments with a triplet loss-based approach and subsequently move towards a fully supervised framework, exploring different architectures to better incorporate geo-spatial relationships.

As part of our next steps, we plan to pre-train the RoBERTa model specific to each state before using it for subsequent experiments, anticipating that this localized pre-training will enhance model performance. Although the multi-head setup shows promise for capturing hierarchical geo-spatial structures and performs best for our use case, we plan to explore its effectiveness further in future experiments. We also intend to integrate contrastive learning into the multi-head learning framework for potentially greater improvements.

References

Pavel Berkhin, Michael R. Evans, Florin Teodorescu, Wei Wu, and Dragomir Yankov. 2015. A new approach to geocoding: Binggc. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '15, New York, NY, USA. Association for Computing Machinery.

Abhranil Chatterjee, Janit Anjaria, Sourav Roy, Arnab Ganguli, and Krishanu Seal. 2016. Sagel: Smart address geocoding engine for supply-chain logistics. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPACIAL '16, New York, NY, USA. Association for Computing Machinery.

Google. 2020. [link].

Uber H3. 2020. [link].

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Govind Kothari 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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Vishal Srivastava, Priyam Tejaswin, Lucky Dhakad, Mohit Kumar, and Amar Dani. 2020. *A Geocoding Framework Powered by Delivery Data*, page 568–577. Association for Computing Machinery, New York, NY, USA.