Building an Amharic E-commerce Data Extractor: Fueling FinTech with Telegram Insights for EthioMart

Prepared by: Abenezer Solomon
Date: June 24, 2025

Executive Summary: Transforming Messy Telegram Posts into a Smart FinTech Engine

In today's dynamic digital economy, informal e-commerce activities conducted through platforms like Telegram are burgeoning, particularly within Ethiopia. However, the inherent decentralization of these operations poses significant challenges for both vendors and customers, complicating product discovery, order placement, and effective communication. EthioMart envisions a strategic solution: establishing a single, centralized platform that consolidates real-time data from these disparate Telegram channels into a unified hub.

This project serves as the foundational blueprint for EthioMart's vision. Our core objective is to develop an **Amharic Named Entity Recognition (NER) system** capable of automatically extracting crucial business entities—such as product names, prices, and locations—from the rich, unstructured text, images, and documents disseminated across these Telegram channels. This extracted, structured data will be instrumental in populating EthioMart's centralized database, thereby transforming it into a comprehensive e-commerce hub. Ultimately, this structured data will power a sophisticated FinTech engine designed to identify and evaluate promising vendors for micro-lending opportunities based on their observed business activity and engagement.

This report meticulously outlines our comprehensive development workflow, encompassing rigorous data ingestion and preprocessing, meticulous dataset labeling, cutting-edge LLM fine-tuning, insightful model comparison, and a conceptual demonstration of its application in vendor analysis. The aim is to demonstrate the end-to-end capabilities required to meet EthioMart's ambitious business goals.

1. The Data Journey: From Raw Posts to Labeled Gold

The foundation of any intelligent system is its data. Our ambitious goal of building a smart FinTech engine begins with a systematic approach to data acquisition and preparation.

1.1. Data Ingestion: Tapping into the Telegram Stream

To adequately fuel our NER model and the EthioMart platform, a robust data ingestion system was meticulously designed and implemented.

- **Sources:** We strategically identified and established connections to **8 active Ethiopian-based e-commerce Telegram channels** for data extraction. These channels, including prominent examples like @Shegeronlinestore, @qnashcom, and @marakibrand, serve as our primary data sources, offering a rich stream of textual descriptions, product images, and various shared documents.
- **Dual Ingestion Strategy:** Our system employs a two-pronged approach for data collection to ensure both breadth and freshness:
    - **Historical Data Scraping:** An initial, large-scale scrape was conducted, successfully collecting over **30,000 messages** from the selected channels. This substantial historical dataset provides the foundational volume of data essential for the initial fine-tuning of our Large Language Model (LLM).
    - **Real-Time Message Ingestion:** Crucially, a real-time message ingestion system has been implemented. This event-driven component continuously monitors the designated Telegram channels, capturing new posts as they are published. This real-time capability is vital for fulfilling EthioMart's core business need of consolidating *live* data, thereby ensuring the centralized platform's content remains current and provides timely insights for dynamic e-commerce operations.
- **Tools & Storage:** The ingestion system is custom-built using the telethon Python library, with sensitive API credentials (api_id, api_hash, phone) securely managed via python-dotenv. All raw collected data—text content stored in CSV format and associated media files in a dedicated 'photos' directory—is persistently saved to Google Drive. This strategy guarantees data integrity and accessibility across different computing sessions. Furthermore, essential metadata such as Channel Title, Username, Message ID, and Date are systematically extracted and separated from the primary message content during the ingestion phase.

1.2. Preprocessing the Amharic Gold: Cleaning for Clarity

Raw, unstructured data from informal platforms like Telegram is inherently noisy. To prepare this valuable information for effective machine learning, the extracted text (specifically the Message column) undergoes a meticulous preprocessing pipeline to ensure consistency and quality.

- **Handling Missing Data:** Any message entries lacking content (represented as NaN values) are robustly converted to empty strings, thereby preventing potential errors in subsequent text processing steps.

- **Emoji Removal:** Telegram messages are often rich with emojis. These non-linguistic elements are comprehensively removed from the message text using the `emoji` Python library. This crucial step cleans the input and eliminates noise irrelevant to Named Entity Recognition.
- **Amharic Punctuation Normalization:** Amharic employs its own distinct set of punctuation marks (e.g., ።, ፣, ፤, ፥). These are standardized to their widely recognized Latin equivalents, promoting uniformity across the dataset. Additionally, any instances of excess whitespace are reduced to single spaces, and leading/trailing spaces are meticulously removed.
- **The Heart of Text Preparation: Amharic-Aware Tokenization:**
    - For Transformer-based LLMs, tokenization is a pivotal step. We segment the cleaned text into subword units, a technique particularly effective for handling the complex morphology and agglutinative nature of the Amharic language.
    - We specifically utilized the `AutoTokenizer` from Hugging Face's `transformers` library. Our chosen tokenizer, `Davlan/bert-base-multilingual-cased-finetuned-amharic`, proved to be instrumental in this regard. Unlike more generic multilingual tokenizers that frequently produced numerous `[UNK]` (unknown) tokens for Amharic words, this specialized tokenizer, having been pre-trained and fine-tuned on Amharic text, accurately segments Amharic words into meaningful subword units. This precise tokenization is critical as it ensures our model can effectively learn from the actual Amharic content, minimizing information loss.
- **Output:** The preprocessed data, comprising cleaned message text and its corresponding `raw_tokens` (lists of subword units), is then saved into a new CSV file (`processed_telegram_messages.csv`). This structured output serves as the prepared input for the subsequent NER labeling phase.

1.3. Crafting the Ground Truth: NER Labeling (CoNLL Format)

To effectively teach our LLM what constitutes a "Product," "Price," or "Location" within the context of Amharic e-commerce, we require high-quality labeled examples. This manual labeling process establishes the "ground truth" for our model.

- **Objective:** To manually label a subset of our preprocessed messages following the standard **CoNLL format**. This format precisely specifies the entity boundaries for each token.
- **Entity Types:** Our labeling scheme adheres to the project's defined entity types:
    - **Core Entities:** `Product` (`B-Product`, `I-Product`), `Price` (`B-PRICE`, `I-PRICE`), and `Location` (`B-LOC`, `I-LOC`).
    - **Optional Entities:** `CONTACT_INFO` (for phone numbers, Telegram usernames) and `DELIVERY_FEE` are also included where applicable.

    - **Outside Entities:** `O` is used for tokens not belonging to any defined entity.
- **Methodology:** An interactive Python script was developed to facilitate this manual annotation process. The script:
    - Loads the `processed_telegram_messages.csv` file, converting the string representation of token lists back into actual Python lists for processing.
    - Selects a subset of messages for labeling (we aimed for an initial 30-50 messages, which resulted in **38 unique samples** for training/evaluation after internal processing).
    - Crucially, it **reconstructs and displays the full message text** from its tokens to provide essential context for accurate human understanding and labeling, mitigating the difficulty of labeling isolated subword tokens.
    - Prompts the annotator for a CoNLL label for each individual token, ensuring adherence to the BIO (Beginning, Inside, Outside) tagging scheme.
    - Includes control commands (`skip`, `exit`, `restart_message`) to manage the interactive labeling workflow efficiently.
- **Output:** The script generates a plain text file (`my_labeled_data_conll.txt`), meticulously formatted according to the strict CoNLL standard (token `\t` label, with a single blank line separating each message).
- **Dataset Size Note:** While 38 labeled samples serve as a critical initial ground truth, it is imperative to acknowledge that LLMs typically require **hundreds to thousands of labeled examples** for robust, production-level NER accuracy. This current sample size is a primary factor influencing the model's current performance, and expanding this dataset is a crucial next step.

2. Bringing it to Life: Fine-Tuning the Amharic NER Model

With our meticulously prepared and labeled dataset, the next pivotal phase is to adapt a pre-trained Large Language Model for our highly specialized Amharic NER task.

2.1. Why PEFT (LoRA) is Our Best Friend

Traditional "full fine-tuning" of LLMs involves updating every single one of their millions or even billions of parameters. While powerful, this approach is prohibitively computationally expensive, demands significant GPU memory, and results in unwieldy large model checkpoints. To circumvent these challenges, we opted for **Parameter-Efficient Fine-Tuning (PEFT)**, specifically implementing the **LoRA (Low-Rank Adaptation)** methodology.

- **Mechanism:** LoRA operates by "freezing" the vast majority of the original pre-trained LLM's parameters. Instead, it strategically injects and trains only a tiny set of new, low-rank matrices (adapters) into select layers of the Transformer

architecture. This dramatically reduces the number of trainable parameters.

- **Benefits for Our Project:** This approach is ideally suited for our project due to several key advantages:
  - **Resource Efficiency:** It drastically lowers the computational power and GPU memory required for training, making fine-tuning highly feasible on platforms like Google Colab's free tier or more modest local GPU setups.
  - **Mitigates Catastrophic Forgetting:** By preserving most of the original model's weights, LoRA effectively prevents "catastrophic forgetting," ensuring the model retains its broad Amharic linguistic understanding while specializing in NER.
  - **Effectiveness with Limited Data:** LoRA often proves more effective than full fine-tuning in scenarios where labeled data is relatively scarce, such as our initial dataset.
  - **Portability:** The resulting fine-tuned model checkpoints (containing only the LoRA adapters) are exceedingly small (typically in MBs, not GBs), facilitating easy storage, sharing, and deployment.
- **Tools:** Our implementation is built upon the robust `transformers` and `peft` libraries provided by Hugging Face.

2.2. Model Comparison & Selection: Identifying the Amharic NER Champion

To identify the most suitable model for EthioMart's specific NER requirements, we fine-tuned and critically compared several Transformer-based models, ranging from general multilingual architectures to those with specialized Amharic/African language pre-training.

- **Candidate Models:**
  - **XLM-RoBERTa (**`xlm-roberta-base`**):** A large, general-purpose multilingual model.
  - **BERT (**`bert-base-multilingual-cased`**):** A foundational multilingual BERT model.
  - **AFROMXLM (**`masakhane/afroxlmr-large-ner-masakhaner-1.0_2.0`**):** An XLM-RoBERTa Large variant explicitly fine-tuned for a Named Entity Recognition (NER) task on the MasakhaNER dataset, which includes African languages.
- **Evaluation Metrics:** During fine-tuning, model performance was rigorously evaluated using standard NER metrics: **F1-score** (our primary metric, balancing precision and recall), Precision, Recall, and Accuracy, computed per epoch on a dedicated validation set.
- **Results & Selection:** The comparative fine-tuning process yielded a distinct champion:

Consolidated Model Comparison (5 Epochs)

Model	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
AFROMXLM	**0.2053**	**0.190956**	**0.509786**	**0.599372**	**0.550962**	**0.936960**
XLM-RoBERTa	0.4569	0.417882	0.000000	0.000000	0.000000	0.890887
BERT	0.6983	0.644878	0.000000	0.000000	0.000000	0.890887

- The results clearly demonstrate **AFROMXLM (**masakhane/afroxlmr-large-ner-masakhaner-1.0_2.0**)** as the superior model for this task. It achieved a notable **F1-score of approximately 55.1%** on the validation set, alongside a precision of ~51% and recall of ~60%. In stark contrast, both the generic XLM-RoBERTa and BERT models effectively failed to learn the NER task, exhibiting 0% F1-scores. This profound difference underscores the critical importance of selecting a pre-trained model that possesses strong linguistic understanding in the target language (Amharic) and has prior exposure to NER tasks within similar domains. AFROMXLM's specialized training on African languages for NER was a decisive factor in its strong initial performance.
  Note on Current Performance: While an F1-score of 55.1% represents a strong foundational achievement, it indicates that there is significant room for improvement to reach production-level accuracy (typically 80%+ F1). This current performance is largely attributable to the **limited size of our labeled dataset (2500 samples)**. Large Language Models require hundreds, or ideally thousands, of high-quality labeled examples to fully generalize and achieve state-of-the-art results for complex tasks like NER. Increasing the volume of labeled data is the most critical next step for enhancing model performance.

2.3. Model Interpretability: Building Trust in AI Decisions (Conceptual)

Understanding *why* a complex machine learning model makes certain predictions is vital for building trust, debugging errors, and identifying potential biases. This is the domain of model interpretability.

- **Importance:** For our NER model, interpretability helps answer questions like: "Which tokens strongly influenced the model to classify a sequence as a 'Price'

entity?" or "Why did the model miss a specific location mention?".

- **Tools:** Techniques like **SHAP (SHapley Additive exPlanations)** and **LIME (Local Interpretable Model-agnostic Explanations)** are powerful for this purpose.
  - **SHAP** aims to explain individual predictions by showing how much each feature (token) contributed to the output.
  - **LIME** provides local, interpretable explanations for any model by approximating its behavior around a specific prediction.
- **Conceptual Approach:** While crucial for a production system, implementing and deriving deep insights from SHAP/LIME for a Transformer-based Token Classification model is a complex and time-consuming task. For this project, we've focused on understanding the conceptual setup. In a full implementation, we would expect SHAP/LIME to clearly highlight tokens such as 'ብር' (Birr), numerical digits (e.g., '2000'), and specific Amharic location names (e.g., 'አዲስ አበባ', 'መገናኛ') as highly influential features when the model predicts Price or Location entities.
- **Future Work:** Due to the project's time constraints, the practical implementation and in-depth analytical application of SHAP/LIME are deferred to future development phases. However, the conceptual understanding of their importance for debugging, understanding biases, and ensuring transparency in the model's NER outputs is firmly established.

3. The FinTech Nexus: Vendor Analysis & Lending Scorecard

The ultimate business objective of EthioMart is to empower micro-lending by identifying promising vendors. Our NER system, combined with available metadata, lays the groundwork for this FinTech application.

3.1. Vendor Analytics Engine: Processing Posts for Performance

The core of our FinTech solution is a vendor analytics engine. This script processes posts (leveraging our NER outputs and metadata) to calculate key performance metrics for each vendor. This provides a data-driven basis for assessing a vendor's business activity and customer engagement, which are crucial for loan assessment.

3.2. Key Vendor Metrics & Lending Score Calculation

For demonstration purposes, we analyzed all 8 of our scraped vendor channels. Due to the current F1-score of our NER model (55.1%) and the unavailability of certain raw metadata (like direct "Views" count from the initial Telegram scrape), the NER extractions and some metrics for this scorecard are **simulated (manually derived or conceptually defined)**. This allows us to demonstrate the *methodology* and *potential*

of the scorecard, rather than relying on currently imperfect automated outputs.

- **Calculated Metrics:**
  - **Activity & Consistency (Posting Frequency):** The average number of posts per week, calculated from message timestamps. This indicates how active a business is.
  - **Market Reach & Engagement (Average Views per Post):** (Simulated for this demo, as raw view counts were not initially scraped). In a production system, this would be derived from actual post view counts, serving as a direct indicator of customer exposure.
  - **Business Profile (Average Price Point):** (Simulated based on manually identified prices for demonstration). In a production system, this would be an average of prices automatically extracted by the NER model, helping to classify vendors (e.g., high-volume/low-margin vs. low-volume/high-margin sellers).
- **Lending Score Design:** A simple, weighted "Lending Score" was designed to combine these key metrics. Our conceptual formula weights Posting Frequency (0.4), Average Views per Post (0.3), and Average Price Point (0.3), after appropriate scaling, to prioritize active vendors with good reach and potentially higher-value products.
- **Consolidated Vendor Scorecard:** The following table summarizes the comparative analysis for all 8 vendors:

Consolidated Vendor Scorecard

Vendor	Posts/Week	Avg. Views/Post (Simulated)	Avg. Price (ETB, Simulated)	Lending Score
@modernshoppingcenter	37.47	880	2760.00	25.91
@marakibrand	22.62	950	4133.33	24.30
@MerttEka	36.80	780	890.00	19.73
@sinayelj	30.04	720	1090.00	17.45
@qnashcom	14.77	1000	2664.29	16.90

@ethio_brand_collection	8.17	900	2160.00	12.45
@Leyueqa	9.05	650	730.00	7.76
@Shegeronlinestore	2.93	1100	416.67	5.72

- This table clearly illustrates a ranking of vendors based on the defined metrics and lending score. For instance, @modernshoppingcenter and @marakibrand appear as top contenders based on this conceptual model, driven by higher posting frequency, strong simulated views, and competitive average price points.
  Note: This demonstration utilizes manually identified (simulated) NER extractions and conceptualized metrics where raw data was unavailable (e.g., 'Views'), due to project constraints. In a production system, these metrics would be calculated automatically using outputs from a highly accurate NER model and comprehensive metadata scraping.

Conclusion & Future Work

This project successfully establishes a foundational pipeline for building an Amharic E-commerce Data Extractor, directly addressing EthioMart's business need for centralized Telegram insights and FinTech vendor evaluation.

Key Achievements:

- **End-to-End Workflow:** A repeatable workflow has been developed, covering data ingestion (historical and real-time), preprocessing (Amharic-aware cleaning, tokenization), and CoNLL-formatted labeling.
- **Effective Model Selection & Fine-Tuning:** The project demonstrated successful PEFT (LoRA) fine-tuning of a Transformer-based model. Through rigorous comparison, **AFROMXLM (**masakhane/afroxlmr-large-ner-masakhaner-1.0_2.0**)** was identified as the champion model, showcasing promising NER capabilities for Amharic with an initial F1-score of ~55.1%, significantly outperforming generic multilingual models.
- **Conceptual Application:** The project provided a clear demonstration of how extracted NER data, combined with metadata, can be used to build a "Vendor Scorecard" and derive a "Lending Score," directly tying the technical solution to EthioMart's FinTech objectives.

Learning Outcomes Achieved:

This challenge successfully demonstrated the ability to programmatically collect and preprocess multi-modal data, apply standard NER labeling schemes, adapt LLMs for specialized tasks using the Hugging Face ecosystem, evaluate models based on metrics, conceptualize interpretability techniques, and articulate the connection between a technical solution and business objectives.

Future Work & Recommendations:

To transition this proof-of-concept into a robust, production-ready system, the following crucial steps are recommended:

1. **Extensive Data Labeling:** This is the single most critical next step. Achieving production-level NER accuracy (typically 80%+ F1-score) demands **significantly more high-quality labeled data (hundreds to thousands of messages)**. This will allow the AFROMXLM model to learn more complex and nuanced patterns.
2. **Iterative Fine-tuning:** Continuously fine-tune the AFROMXLM model on the expanding labeled dataset. Leverage the real-time ingestion system to constantly grow the dataset and periodically re-train, ensuring the model remains updated and accurate.
3. **Comprehensive Metadata Scraping:** Enhance the Telegram scraper to capture additional vital metadata, especially actual Views per post, to make the Vendor Scorecard truly data-driven without simulation.
4. **Full Implementation of Model Interpretability:** Integrate and analyze insights from SHAP and LIME tools. This will be invaluable for debugging model errors, understanding biases, and building strong trust in the automated NER outputs.
5. **Refining Vendor Scorecard:** Develop more sophisticated vendor scoring algorithms, potentially incorporating more granular NER outputs (e.g., specific product categories, material mentions, delivery fees) and linking them to financial health indicators.
6. **Deployment:** Plan for the deployment of the fine-tuned NER model and the vendor analytics engine within EthioMart's centralized platform.