```
# ----------------------------
# 1. CORE IMPORTS
# ----------------------------
import pandas as pd
import numpy as np
import duckdb
import gdown
from tqdm import tqdm

# HuggingFace translation
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM, pipeline

# Display settings
pd.set_option("display.max_colwidth", None)

print("Setup complete.")
```

```
Setup complete.
```

```
import gdown

file_id = "1RxeikLQHjYEJCewMtxpbNVlQ1FcQcC9s"
url = f"https://drive.google.com/uc?id={file_id}"

output = "/content/farmer_questions.csv"

gdown.download(url, output, quiet=False)

print("Download complete:", output)
```

```
Downloading...
From (original): https://drive.google.com/uc?id=1RxeikLQHjYEJCewMtxpbNVlQ1FcQcC9s
From (redirected): https://drive.google.com/uc?id=1RxeikLQHjYEJCewMtxpbNVlQ1FcQcC9s&confirm=t&uuid=a79554
To: /content/farmer_questions.csv
100%|██████████| 7.25G/7.25G [01:55<00:00, 62.9MB/s]Download complete: /content/farmer_questions.csv
```

```
import duckdb

con = duckdb.connect("/content/producers_direct.duckdb")

con.execute("""
    CREATE OR REPLACE VIEW raw_questions AS
    SELECT * FROM read_csv_auto('/content/farmer_questions.csv');
""")

print("View ready.")
```

```
View ready.
```

```
con.execute("SHOW TABLES").df()
```

	name
0	raw_questions

```
con.execute("PRAGMA table_info('raw_questions')").df()['name'].tolist()
```

```
['question_id',
 'question_user_id',
 'question_language',
 'question_content',
 'question_topic',
 'question_sent',
 'response_id',
 'response_user_id',
 'response_language',
 'response_content',
 'response_topic',
 'response_sent',
 'question_user_type',
 'question_user_status',
 'question_user_country_code',
 'question_user_gender',
 'question_user_dob',
 'question_user_created_at',
 'response_user_type',
 'response_user_status',
 'response_user_country_code',
 'response_user_gender',
 'response_user_dob',
 'response_user_created_at']
```

```
con.execute("SELECT * FROM raw_questions LIMIT 5").df()
```

	question_id	question_user_id	question_language	question_content	question_topic	question_sent	res
0	3849056	519124	nyn	E ABA WEFARM OFFICES ZABO NIZISHANGWA NKAHI?	None	2017-11-22 12:25:03+00:00	
1	3849061	521327	eng	Q this goes to wefarm. is it possible to get for us market for our product. thax	None	2017-11-22 12:25:05+00:00	
2	3849077	307821	nyn	E ENTE YANJE EZAIRE ENYENA YASHOBERA. \nOBWIRE BWOKUZARA BUBAIRE BWAHIKIRE EZAIRE AKANYENA KAKYE KAMARAHO ENDAKIKA ITANO KAFA .KANDI NKAZINJE .OBWO NIBURWIREKI??	cattle	2017-11-22 12:25:08+00:00	
3	3849077	307821	nyn	E ENTE YANJE EZAIRE ENYENA YASHOBERA. \nOBWIRE BWOKUZARA BUBAIRE BWAHIKIRE EZAIRE AKANYENA KAKYE KAMARAHO ENDAKIKA ITANO KAFA .KANDI NKAZINJE .OBWO NIBURWIREKI??	cattle	2017-11-22 12:25:08+00:00	
4	3849077	307821	nyn	E ENTE YANJE EZAIRE ENYENA YASHOBERA. \nOBWIRE BWOKUZARA BUBAIRE BWAHIKIRE EZAIRE AKANYENA KAKYE KAMARAHO ENDAKIKA ITANO KAFA .KANDI NKAZINJE .OBWO NIBURWIREKI??	cat	2017-11-22 12:25:08+00:00	

5 rows × 24 columns

```
con.sql("""
    SELECT
        COUNT(*) AS total_rows,

        -- core fields
        SUM(question_id IS NULL) AS missing_question_id
        SUM(question_id IS NULL) AS missing_question_id,
        SUM(question_content IS NULL) AS missing_question_content,
        SUM(question_language IS NULL) AS missing_question_language,
        SUM(question_topic IS NULL) AS missing_question_topic,
        SUM(question_user_id IS NULL) AS missing_question_user_id,

        -- challenge 4 fields
        SUM(question_user_country_code IS NULL) AS missing_country_code,
        SUM(question_sent IS NULL) AS missing_question_sent

    FROM raw_questions;
""").df()
```

```
   total_rows  missing_question_id  missing_question_content  missing_question_language  missing_question
0    20304843                  0.0                       0.0                        0.0               35
```

```
con.sql("""
    CREATE OR REPLACE VIEW question_level AS
    SELECT DISTINCT
        question_id,
        question_content,
        question_language,
        question_topic,
        question_user_country_code,
        question_sent
    FROM raw_questions;
""")
```

```
con.sql("""
    SELECT COUNT(*) AS n_question_level_rows
    FROM question_level;
""").df()
```

```
   n_question_level_rows
0                6627409
```

```
con.sql("""
    SELECT
        COUNT(*) AS total_questions,
        SUM(question_topic IS NULL) AS missing_topics
    FROM question_level;
""").df()
```

```
   total_questions  missing_topics
0          6627409       1672009.0
```

```
con.sql("""
    SELECT question_topic, COUNT(*) AS n
    FROM question_level
    GROUP BY question_topic
    ORDER BY n DESC;
""").df()
```

	question_topic	n
0	None	1672009
1	maize	595779
2	chicken	493791
3	cattle	462713
4	tomato	353851
...	...	...
144	blackberry	27
145	setaria	25
146	mulberry	22
147	purple-vetch	12
148	cranberry	4

149 rows × 2 columns

```
con.sql("""
    SELECT DISTINCT question_topic
    FROM question_level
    ORDER BY 1;
""").df()
```

	question_topic
0	acacia
1	african-nightshade
2	amaranth
3	animal
4	apple
...	...
144	vetch
145	watermelon
146	wheat
147	yam
148	None

149 rows × 1 columns

```
con.sql("""
    SELECT DISTINCT question_language
    FROM question_level;
""").df()
```

```
  question_language
0               nyn
1               eng
2               swa
3               lug
```

```
con.sql("""
    SELECT question_id, COUNT(*)
    FROM question_level
    GROUP BY question_id
    HAVING COUNT(*) > 1
    ORDER BY COUNT(*) DESC
    LIMIT 20;
""").df()
```

```
    question_id  count_star()
0      33636032            15
1      44092721            15
2       8811829            14
3      36617024            13
4      36717802            13
5      34835109            13
6      24015389            12
7      31138296            12
8      42214400            12
9      43450645            12
10      5233828            11
11     34764414            11
12     26529316            11
13     20433220            11
14     19636378            11
15     23316811            11
16     19636223            11
17     19635914            11
18     27272879            11
19     22928035            11
```

```
con.sql("""
    SELECT SUM(occurrences - 1) AS total_duplicate_question_rows
    FROM (
        SELECT question_id, COUNT(*) AS occurrences
        FROM question_level
        GROUP BY question_id
        HAVING COUNT(*) > 1
    );
""").df()
```

```
   total_duplicate_question_rows
0                       761590.0
```

```
con.sql("""
    CREATE OR REPLACE VIEW question_level AS
    SELECT DISTINCT
        question_id,
        question_content,
        question_language,
        question_topic,
        question_user_country_code,
        question_sent
    FROM raw_questions;
""")
```

```
con.sql("""
    SELECT COUNT(*) AS n_question_level_rows
    FROM question_level;
""").df()
```

```
   n_question_level_rows
0                6627409
```

```
con.sql("""
    SELECT
        COUNT(*) AS total_rows,
        SUM(question_content IS NULL) AS missing_question_content,
        SUM(question_language IS NULL) AS missing_question_language,
        SUM(question_topic IS NULL) AS missing_question_topic,
        SUM(question_user_country_code IS NULL) AS missing_country_code
    FROM question_level;
""").df()
```

```
   total_rows  missing_question_content  missing_question_language  missing_question_topic  missing_count
0     6627409                       0.0                        0.0               1672009.0
```

table	column	issue	row_count	magnitude	solvable?	
question_level	question_topic	missing topic labels	1,672,009	25.23%	N	leave as is – no way to
question_level	question_content	no missing values	0	0.00%	N/A	no action needed
question_level	question_language	no missing values	0	0.00%	N/A	no action needed
question_level	question_user_country_code	no missing values	0	0.00%	N/A	no action needed
raw_questions	question_id	duplicates collapse into fewer rows	14,439,024 dupes	71.1%	Y	resolved by creating qu

N — Note & Document

In this final step, I documented every decision made during the data cleaning process. I completed the issues log with row counts and percentages, explained how each issue was handled, and recorded which issues could not be fixed. I also kept a clear paper trail by noting the code used to remove duplicates, the columns kept for analysis, and the reasoning behind leaving missing topics unchanged. This documentation provides transparency, shows how the dataset was transformed, and ensures that anyone reviewing the project can follow the cleaning process from beginning to end.

Below is the final issues log summarizing all identified problems, their magnitude, and the actions taken:

```
ISSUES LOG


table           column                        issue                                   row_count     magnitu
--------------- ----------------------------- --------------------------------------- ------------- ------
question_level  question_topic                missing topic labels                     1,672,009      25.2
question_level  question_content              no missing values                        0              0.00
question_level  question_language             no missing values                        0              0.00
question_level  question_user_country_code    no missing values                        0              0.00
raw_questions   question_id                   duplicate question rows in raw data      14,439,024     ~71%
```

These notes complete the data cleaning documentation. All cleaning steps and decisions were recorded to maintain clarity, support reproducibility, and provide a transparent audit trail for downstream translation and topic analysis.

```
# Row count: should be ~6.6M unique questions
con.sql("""
    SELECT COUNT(*) AS n_question_level_rows
    FROM question_level;
""").df()

# Quick sanity check on columns and values
con.sql("""
    SELECT *
    FROM question_level
    LIMIT 5;
""").df()
```

	question_id	question_content	question_language	question_topic	question_user_country_code	question
0	3849077	E ENTE YANJE EZAIRE ENYENA YASHOBERA. \nOBWIRE BWOKUZARA BUBAIRE BWAHIKIRE EZAIRE AKANYENA	nyn	cattle	ug	2017 12:25:08

```
sample = con.sql("""
    SELECT
        question_id,
        question_language,
        question_content
    FROM question_level
    WHERE question_language != 'eng'
    LIMIT 5;
""").df()

sample
```

	question_id	question_language	question_content
0	3937171	swa	sungura wangu wako na upele niwape ndawa gani?
1	3937182	swa	S;nmefuka samaki na nmekoxa soko.Where wll I get?.
2	3937212	nyn	E# nimbuza ngu ebihimba hati biri arizingahi?.
3	3937230	swa	s napaswa kula nyama ya ngombe ambaye alikufa usiku.?
4	3937270	swa	S Niko Na Maragwe Gunia Tanu Na Ta Futa Shoko Kama Unataka Napati Kana

```
from transformers import pipeline

translator = pipeline(
    "translation",
    model="facebook/nllb-200-distilled-600M",
    device_map="auto"
)

def translate_row(text, src_lang_code):
    return translator(
        text,
        src_lang=src_lang_code,     # e.g. "nyn_Latn", "swh_Latn"
        tgt_lang="eng_Latn",
        max_length=400
    )[0]["translation_text"]

sample["translated_en"] = sample.apply(
    lambda r: translate_row(r["question_content"], r["question_language"]),
    axis=1
)

sample
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/se
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
```

```
config.json:  100%                                          846/846  [00:00<00:00,  45.4kB/s]

pytorch_model.bin:  100%                                         2.46G/2.46G  [01:28<00:00,  73.0MB/s]

model.safetensors:  100%                                         2.46G/2.46G  [01:06<00:00,  69.5MB/s]

generation_config.json:  100%                                       189/189  [00:00<00:00,  11.4kB/s]

tokenizer_config.json:  100%                                        564/564  [00:00<00:00,  57.6kB/s]

sentencepiece.bpe.model:  100%                                        4.85M/4.85M  [00:01<00:00,  4.19MB/s]

tokenizer.json:  100%                                           17.3M/17.3M  [00:00<00:00,  21.8MB/s]

special_tokens_map.json:     3.55k/?  [00:00<00:00,  232kB/s]
```

```
Device set to use cuda:0
```

	question_id	question_language	question_content	translated_en
0	3937171	swa	sungura wangu wako na upele niwape ndawa gani?	What medication should I give my baby?
1	3937182	swa	S;nmefuka samaki na nmekoxa soko.Where wll I get?.	S;nmefuka fish and nmekoxa soko.Where will I get?.
2	3937212	nyn	E# nimbuza ngu ebihimba hati biri arizingahi?.	E# nimbuza by ebihimba that are arizingahi?.

```
batch = con.sql("""
    SELECT *
    FROM question_level
    WHERE question_language != 'eng'
    ORDER BY question_id
    LIMIT 5000
    OFFSET 0
""").df()
```

```
def translate_row(text, src_lang_code):
    return translator(
        text,
        src_lang=src_lang_code,
        tgt_lang="eng_Latn",
        max_length=400,
    )[0]["translation_text"]
```

```
from tqdm.auto import tqdm
tqdm.pandas()  # lets us use progress_apply on DataFrames
```

```
batch["translated_en"] = batch.progress_apply(
    lambda r: translate_row(r["question_content"], r["question_language"]),
    axis=1
)
```

```
100%                                          5000/5000  [57:39<00:00,    1.72it/s]
```

```
batch[["question_id", "question_language", "question_content", "translated_en"]].head()
```

	question_id	question_language	question_content	translated_en
0	3849056	nyn	E ABA WEFARM OFFICES ZABO NIZISHANGWA NKAHI?	Where are the WEFARM offices used?
1	3849077	nyn	E ENTE YANJE EZAIRE ENYENA YASHOBERA. \nOBWIRE BWOKUZARA BUBAIRE BWAHIKIRE EZAIRE AKANYENA KAKYE KAMARAHO ENDAKIKA ITANO KAFA .KANDI NKAZINJE .OBWO NIBURWIREKI??	My wife, Ezaire, is a good-for-nothing woman, and I'm a good-for-nothing woman, but I'm a good-for-nothing woman, and I'm a good-for-nothing woman, and I'm a good-for-nothing woman, and I'm a good-for-nothing woman, and I'm a good-for-nothing woman, and I'm a good-for-nothing woman, and I'm good-for-nothing, and I'm good-for-nothing, and I'm good-for-nothing, and I'm good-for-nothing, and I'm good-for-nothing, and I'm good-for-nothing, and I'm good-for-nothing, and I'm good-for-nothing, and

```
# Any empty translations?
batch[batch["translated_en"].isna() | (batch["translated_en"].str.strip() == "")]
```

	question_id	question_content	question_language	question_topic	question_user_country_code	quest
1555	3886291	s muembe uzaa	swa	mango	ke	2

```
# How many rows per language in this batch
batch["question_language"].value_counts()
```

	count
question_language	
swa	2877
nyn	1927
lug	196

dtype: int64

```
# language distribution
con.sql("""
    SELECT
        question_language,
        COUNT(*) AS n_rows
    FROM question_level
    GROUP BY question_language
    ORDER BY n_rows DESC;
""").df()

# total non-English rows
con.sql("""
    SELECT
        COUNT(*) AS non_english_rows
    FROM question_level
    WHERE question_language != 'eng';
""").df()
```

	non_english_rows
0	3242518

```
batch.to_csv("translated_batch_000.csv", index=False)

from google.colab import files
files.download("translated_batch_000.csv")
```

```
con.sql("""
    CREATE TABLE IF NOT EXISTS translated_questions AS
    SELECT *
    FROM read_csv_auto('translated_batch_000.csv');
""")
```

```
con.sql("""
    SELECT *
    FROM translated_questions
    LIMIT 3;
""").df()
```

	question_id	question_content	question_language	question_topic	question_user_country_code	question
0	3849056	E ABA WEFARM OFFICES ZABO NIZISHANGWA NKAHI?	nyn	None	ug	2017 12:25:03

```
con.sql("""
    SELECT COUNT(*) AS n_translated
    FROM translated_questions;
""").df()
```

	n_translated
0	5000

```
con.sql("""
    CREATE OR REPLACE TABLE translated_ids AS
    SELECT DISTINCT question_id
    FROM translated_questions;
""")
```

We have created a helper table of translated ids. This lets use skip anything we already translated.

```
batch_size = 5000  # change if you want

batch = con.sql(f"""
    SELECT
        q.question_id,
        q.question_language,
        q.question_content,
        q.question_topic,
        q.question_user_country_code,
        q.question_sent
    FROM question_level q
    LEFT JOIN translated_ids t
        USING (question_id)
    WHERE q.question_language != 'eng'
      AND t.question_id IS NULL   -- skip already translated (this comment is inside SQL, so it's fine)
    ORDER BY q.question_id
    LIMIT {batch_size};
""").df()

batch.head()
```

	question_id	question_language	question_content	question_topic	question_user_country_code	question
0	3969507	swa	S NITAPATA WAPI MBOLEA YA KUTENGENEZWA (ORGANICALY) NA NIPESA NGAPI?	None	ke	2017 19:04:11+
1	3969513	swa	s:mnawezapatiana mifugo ili kufugiwa	livestock	ke	2017 19:04:32+

Next steps: Generate code with `batch` New interactive sheet

```
con = duckdb.connect("/content/producers_direct.duckdb")  # or your path
```

```
con = duckdb.connect()   # creates an in-memory DB
```

```
con.execute("""
CREATE OR REPLACE TABLE translated_questions AS
SELECT *
FROM read_csv_auto('/content/translated_batch_000.csv');
""")
```

```
<duckdb.duckdb.DuckDBPyConnection at 0x7e47855b37f0>
```

```
con.execute("SELECT * FROM translated_questions LIMIT 5").df()
```

	question_id	question_content	question_language	question_topic	question_user_country_code	question
0	3849056	E ABA WEFARM OFFICES ZABO NIZISHANGWA NKAHI?	nyn	None	ug	2017 12:25:03+

```
con.execute("""
CREATE OR REPLACE VIEW local_text AS
SELECT
    question_id,
    question_language,
    question_content,
    -- lowercase + strip punctuation
    REGEXP_REPLACE(LOWER(question_content), '[^[:alnum:] ]', ' ') AS text_clean,
    translated_en
FROM translated_questions
WHERE question_language IN ('swa', 'lug', 'nyn');
""")

con.execute("SELECT * FROM local_text LIMIT 5").df()
```

	question_id	question_language	question_content	text_clean	translated_en
0	3849056	nyn	E ABA WEFARM OFFICES ZABO NIZISHANGWA NKAHI?	e aba wefarm offices zabo nizishangwa nkahi	Where are the WEFARM offices used?
1	3849077	nyn	E ENTE YANJE EZAIRE ENYENA YASHOBERA. \nOBWIRE BWOKUZARA BUBAIRE BWAHIKIRE EZAIRE AKANYENA KAKYE	e ente yanje ezaire enyena yashobera \nobwire bwokuzara bubaire bwahikire ezaire akanyena kakye	My wife, Ezaire, is a good-for-nothing woman, and I'm a good-for-nothing woman, but I'm a good-for-nothing woman, and I'm a good-for-nothing woman, and I'm a good-for-nothing woman, and I'm a good-for-nothing woman, and I'm a good-for-nothing woman, and I'm a good-for-nothing woman, and I'm good-for-nothing, and I'm good-for-

```
term_unigrams = con.execute("""
WITH tokens AS (
    SELECT
        question_id,
        question_language,
        translated_en,
        UNNEST(STRING_SPLIT(text_clean, ' ')) AS term
    FROM local_text
),
clean_tokens AS (
    SELECT
        question_id,
        question_language,
        translated_en,
        term
    FROM tokens
    WHERE term IS NOT NULL
      AND term <> ''
      AND LENGTH(term) > 2
)
SELECT
    question_language,
    term,
    COUNT(*) AS term_count,
    COUNT(DISTINCT question_id) AS n_questions
FROM clean_tokens
GROUP BY question_language, term
HAVING COUNT(*) >= 5
ORDER BY term_count DESC;
""").df()

term_unigrams.head(20)
```

	question_language	term	term_count	n_questions
0	swa	gani	632	537
1	swa	kuku	578	375
2	swa	dawa	514	452
3	swa	kwa	484	376
4	swa	nataka	215	110
5	swa	mahindi	188	144
6	swa	wapi	183	155
7	swa	nini	180	143
8	nyn	kandi	179	153
9	swa	kienyeji	174	78
10	nyn	ndi	173	157
11	swa	wangu	171	150
12	nyn	ninyenda	171	153
13	nyn	ente	160	138
14	swa	niko	154	124
15	swa	mbegu	147	136
16	swa	ngapi	146	113
17	swa	nyanya	145	114
18	nyn	nimbuza	141	136
19	swa	sungura	131	108

Next steps: Generate code with term_unigrams New interactive sheet

```
con.execute("""
CREATE OR REPLACE TABLE glossary_auto AS
WITH tokens AS (
    SELECT
        q.question_id,
        q.question_language,
        q.translated_en,
        UNNEST(STRING_SPLIT(q.text_clean, ' ')) AS term
    FROM local_text q
),
clean_tokens AS (
    SELECT
        question_id,
        question_language,
        translated_en,
        term
    FROM tokens
    WHERE term IS NOT NULL
      AND term <> ''
      AND LENGTH(term) > 2
)
SELECT
    question_language,
    term,
    LIST(DISTINCT translated_en)[1] AS sample_english,
    COUNT(*) AS term_occurrences,
    COUNT(DISTINCT question_id) AS n_questions
FROM clean_tokens
GROUP BY question_language, term
HAVING COUNT(*) >= 5
ORDER BY term_occurrences DESC;
""")

glossary_df = con.execute("SELECT * FROM glossary_auto").df()
glossary_df.head(15)
```

	question_language	term	sample_english	term_occurrences	n_questions
0	swa	gani	I am a farmer of Indian origin but if I have tuna, pingi and corn I can use it.	632	537
1	swa	kuku	It's a viroboto vaccine for chicken.	578	375
2	swa	dawa	It's a viroboto vaccine for chicken.	514	452
3	swa	kwa	It's a viroboto vaccine for chicken.	484	376
4	swa	nataka	S,I'd like to grow a 100 degree chicken pie and I'll build your house with chicken.I'll have a few feet per square foot.	215	110
5	swa	mahindi	If I plant corn pila fertilizer, and then I use a mixture of D A P and urea, will it grow?	188	144
6	swa	wapi	I CAN get it from wherever I want it.	183	155
7	swa	nini	S napier is what makes a cow drink more milk.	180	143
8	nyn	kandi	How do I know why I'm here and why I'm here?	179	153
9	swa	kienyeji	Q,What kind of treatment have we tried using chicken Upantewa Homa?	174	78
			E#HUMAN WORKING IN THE WORLD OF GOD AYOHEREZA'S WITNESS		

Next steps: Generate code with `glossary_df` New interactive sheet

```
glossary_df.to_csv("prep_challenge_glossary_batch000.csv", index=False)
```

```
keywords_df = con.execute("""
SELECT
    question_language AS language,
    term,
    term_occurrences,
    n_questions
FROM glossary_auto
WHERE term_occurrences >= 10
ORDER BY language, term_occurrences DESC;
""").df()

keywords_df.head(30)
```

1 to 30 of 30 entries Filter

index	language	term	term_occurrences	n_questions
0	lug	kasoli	33	24
1	lug	nga	26	20