import random

from pyspark.sql.functions import udf

from pyspark.sql.types import StringType

def introduce\_typos(word, error\_type):

"""Randomly introduce errors to simulate typos."""

if not word or len(word) < 2:

return word

word = list(word)

if error\_type == "same":

return "".join(word) # No change

elif error\_type == "similar":

idx = random.randint(0, len(word) - 2) # Pick a random index to swap

word[idx], word[idx + 1] = word[idx + 1], word[idx] # Swap adjacent letters

return "".join(word)

elif error\_type == "dissimilar":

return "RANDOM\_" + str(random.randint(100, 999)) # Completely change word

return "".join(word)

# Register as a PySpark UDF

introduce\_typos\_udf = udf(introduce\_typos, StringType())

from pyspark.sql.functions import col, when

# Create a new column with introduced typos

df = df.withColumn(

"Destination\_Modified",

when((col("Source").isNotNull()) & (col("variant\_classif") == "Alphanumeric"),

introduce\_typos\_udf(col("Destination"),

when(col("Source") % 3 == 0, "same") # 1/3 remain the same

.when(col("Source") % 3 == 1, "similar") # 1/3 slightly modified

.otherwise("dissimilar") # 1/3 completely changed

)

).otherwise(col("Destination")) # Leave unchanged if not alphanumeric

)

# Label similarity

df = df.withColumn(

"Similarity\_Label",

when(col("Destination\_Modified") == col("Source"), "Same")

.when(col("Destination\_Modified").contains("RANDOM\_"), "Dissimilar")

.otherwise("Similar")

)