A Congolese Swahili Task-Oriented Dialogue System for Addressing Humanitarian Crises

Ussen Kimanuka
Institute for Basic Sciences, Technology and Innovation, Pan African University, Kenya
abre.ussen@students.jkuat.ac.ke (corresponding author)

Ciira wa Maina
Dedan Kimathi University of Technology, Kenya | Center for Data Science and Artificial Intelligence (DSAIL), Kenya
ciira.maina@dkut.ac.ke

Osman Buyuk
Izmir Demokrasi University, Turkiye
osman.buyuk@idu.edu.tr

Masika Kassay Godelive
Benevolencija, Congo
jojokassay@gmail.com

Received: 27 May 2025 | Revised: 17 July 2025 and 28 July 2025 | Accepted: 1 August 2025

 DOI: https://doi.org/10.48084/etasr.12403

ABSTRACT

As Artificial Intelligence (AI) advances, conversational agents are increasingly used across sectors, including humanitarian response. However, current systems and datasets mainly support high-resource languages and open-domain tasks, resulting in significant limitations in addressing low-resource, domain-specific needs. This study addresses this gap by focusing on a Congolese Swahili corpus collected from Short Message Service (SMS) messages and call-center humanitarian questions to develop an effective conversational agent for low-resource languages that supports communication during humanitarian crises. The goal of this research is to develop an effective Task-Oriented Dialogue System (ToDS) to assist displaced persons seeking humanitarian information in Congolese Swahili. We built a pipeline-based ToDS that converts natural language into SPARQL by utilizing a trained Named Entity Recognition (NER) model and a Dual Intent and Entity Transformer (DIET) classifier. This ToDS includes a humanitarian-specific ontology and dynamically queries a local triple store with data derived from the Humanitarian Data Exchange (HDX). The preliminary results indicate high accuracy in entity recognition and intent classification, which enables precise and timely information responses. The agent effectively provides context-relevant answers to humanitarian questions in crisis interactions. The findings demonstrate that applying Natural Language Understanding (NLU) methods in a low-resource, crisis-based context is viable and impactful. This ToDS offers a scalable solution for improving information accessibility in humanitarian emergencies and during forced internal displacements.

Keywords-conversational Artificial Intelligence (AI); Task-Oriented Dialogue System (ToDS); natural language query formalization; SPARQL; resource data framework; ontology; low-resource languages; humanitarian crisis scenarios

I. INTRODUCTION

Task-Oriented Dialogue Systems (ToDS) are Natural Language Processing (NLP) applications or agents that users interact with to obtain specialized responses. ToDS start by determining the user's goal, termed intent, together with correlated arguments called slots, before processing the request effectively. Task-oriented dialogue structures are intended to enable users to perform a particular task, such as hotel, taxi, and restaurant reservations; information provision; and customer support, among others, whereas non-task-based dialogue frameworks are designed to offer reasonable answers to users, entertain users, and have open domains that are not particularly limited in scope.

The progression of deep neural networks, mainly the practical usage of large pre-trained models, has further driven

significant development in ToDS research in recent years [1]. The main tasks in a ToDS are intent recognition and slot filling, which can be carried out separately [2] or jointly [3] as a unified task. This architecture, typified by the system division into different modules performing distinct functions, is a modular architecture, in contrast to the end-to-end method that uses a neural network for all tasks [4]. Therefore, this makes them highly data-intensive because they simulate numerous tasks using training data. The pipeline of a ToDS comprises four modules: a Natural Language Understanding (NLU) module that extracts the intent and key user slots [5]; a Dialogue State Tracker (DST) that tracks the user's belief state based on dialogue history [6]; a Dialogue Policy Learning (DPL/POL) module that determines the next course of action [7], and a Natural Language Generation (NLG) module that generates dialogue system answers [8]. Even though earlier pipeline ToDS approaches for high-resource languages produced impressive results, they still had significant flaws, including a lack of knowledge shared across all modules and errors carried over from one module to the next [9]. Some works have proposed end-to-end frameworks to address these problems [10-12].

According to the authors in [13, 14], recent years have seen tremendous achievements in End-to-End Task-Oriented Dialogue (EToD), whose approaches can be divided into two categories based on whether intermediate supervision is necessary and whether knowledge-based retrieval is differentiable: (1) modular EToD, which preserves some degree of modularity while sharing parameters; and (2) fully EToD, which employs a single model for the dialogue process. These end-to-end frameworks facilitate discourse beyond domain-specific corpora and increase the expressiveness of the state space. Due to the advantages these systems offer users and businesses, a growing range of application domains and personal services are now being designed to incorporate system support. One key application is in the humanitarian sector, where ToDS can support the assessment of risks and uncertainties related to resilience and previously unidentified humanitarian actions. This, in turn, can enhance the coordination and delivery of humanitarian aid.

Despite the dialogue systems' popularity, most of the development has only been performed on a small cluster of languages, such as French, English, and Chinese, referred to as "resource-rich." Most of the approximately 7,000 remaining languages [15], mainly African ones, are categorized as "low-resource", facing challenges of having adequate corpora and NLP tools [16]. As a result, they fall behind in most deep learning-powered dialogue systems, which achieve state-of-the-art results in high-resource languages but straggle to reproduce similar performance in African languages. It remains uncertain how this method applies to ToDS for low-resource languages.

For example, using machine translation, low-resource languages can capitalize on existing intent response datasets to project annotation onto the low-resource language [17]. However, not all low-resource languages have access to cutting-edge machine translation software, such as Congolese Swahili. In this line of work, few researchers have studied and developed initial ToDS for sub-Saharan African languages in the humanitarian landscape. Authors in [18] implemented and collected proprietary ToDS resources for Congolese languages, including Congolese Swahili and Lingala, to respond to questions related to COVID-19 and Ebola. Other languages, such as Hausa and Kanuri, have witnessed the creation of linguistic resources for ToDS applications through the same initiative. Authors in [19] developed and collected a proprietary ToDS dataset for a chatbot in Congolese Swahili that helps women access legal information to assert their rights. The majority of other ToDS resources were created with the Eastern African Swahili (Swahili sanifu) dialect. Authors in [20] developed a ToDS that interacts with members of a commercial entity and gives voice replies in Kiswahili (Kenyan) via Short Message Service (SMS) messages and WhatsApp. Through this system, members can interact with the bot for various functions, including checking their balance, requesting loans, and receiving transaction statements. Authors in [21, 22] developed an offline ToDS in collaboration with rural smallholder women farmers as an alternative source for agricultural information.

Findings from Swahili-related literature indicate that accessing ToDS datasets is difficult because they are not publicly available, and the methods used to create them are not shared. This slows down progress in research on ToDS for humanitarian purposes. Most of the related work utilizes the traditional pipeline architecture due to limited resources regarding dataset availability for intent detection, slot filling, and response generation.

Table I highlights sub-Saharan ToDS initiatives [23-28]. However, these works have significant limitations. They still fall short of real human interaction scenarios in solving social problems because their current datasets and knowledge bases are still in their infancy, creating a major constraint for chatbot development. Additionally, most existing studies have not evaluated their methods using EToD approaches. Hence, real-world cases need to be applied directly, and both traditional pipeline and modular end-to-end frameworks in low-resource African languages need to be evaluated.

Currently, mobile phones are viewed as a steadfast friend to users. The extensive usage of mobile phones, mainly for SMS communication, has become a critical element of contemporary life. SMS services are a significant contributor to the Gross National Income (GNI) of developing nations [29]. Millions of people use this resource daily due to its convenience, accessibility, rapid delivery, and affordable cost compared to phone calls. Humanitarian crisis management organizations use SMSs and calls to interact with affected communities [30]. Using these communication tools, vulnerable or affected communities can ask the appropriate organization for precise guidance regarding their uncertainties. Thus, the data collected by these humanitarian organizations can serve as a training dataset for ToDS or conversational Artificial Intelligence (AI) agents.

To this end, SMSs and calls from affected persons can be used as data for training ToDS. This inspiration comes from the fact that user-provided questions in a conversation are typically relatively brief, much like SMSs. As a result, the dialogue

system must first ascertain the intent of the question. However, the query's information might be incomplete or implied. If the dialogue system does not have the knowledge or context required to respond to the question, several rounds of dialogue might be needed to resolve these problems and validate the user's intentions. In this line of work, for sub-Saharan Africa, authors in [31, 32] integrated SMS-based data gathering to crowdsource crisis data during emergencies. Through SMS, users could report incidents, which the systems combined and displayed for emergency responders on maps. Author in [33] explained the role of digital humanitarianism by emphasizing how participatory methods centered on SMS data have changed crisis mapping in developing nations. However, this research concentrated mainly on crisis mapping and did not tackle conversational agent development. Therefore, our task is to demonstrate emergency communication capabilities.

TABLE I. SUMMARY OF TASK-ORIENTED DIALOGUE SYSTEMS FOR SUB-SAHARAN AFRICAN LANGUAGES

Ref.	Language	Domain	Platform	Data access	ToDS type	Evaluation
[18]	Congolese Swahili, Lingala, Hausa, Kanuri	Covid-19, Ebola	Telegram, SMS	Proprietary	Pipeline	N/A
[19]	Congolese Swahili	Legal information	Mobile	Proprietary	Pipeline	N/A
[20]	Kiswahili (Kenyan)	Finance	SMS, WhatsApp	Proprietary	ND	N/A
[21, 22]	Kiswahili (Tanzanian)	Agriculture	Mobile	Proprietary	ND	N/A
[23]	Kiswahili (Kenyan)	Climate / weather	Mobile	Synthetic / proprietary	ND	N/A
[7]	Wolof	Multi-domain	CLI	Proprietary	Pipeline	Limited
[24]	Swahili	Mental health	Mobile	Proprietary	ND	N/A
[25]	Kinyarwanda	Covid-19	Mobile, USSD	Proprietary	ND	Limited
[26, 27]	Multiple African	Healthcare	Mobile, chat	Proprietary	ND	N/A
[28]	Swahili, others	Crisis response	Web, mobile	Proprietary	ND	N/A

a. ND: Not Disclosed; Proprietary: Dataset not publicly accessible; Synthetic: Created through machine translation; N/A: Not Available; Limited: Minimal evaluation details.

b. USSD: Unstructured Supplementary Service Data.

c. CLI: Console Line Interface.

This paper addresses two key challenges: the lack of human-annotated datasets and the limited evaluation of conversational AI for the Congolese Swahili language in a humanitarian crisis. We created a large ToDS dataset focused on Congolese Swahili humanitarian emergencies. Congolese Swahili is widely spoken in the Democratic Republic of Congo (DRC). Our dataset targets the emergency response domain and supports tasks such as intent classification and slot filling [34]. We provide several baseline models using a traditional pipeline task-oriented framework and a modular end-to-end framework. Furthermore, we analyze different options for full fine-tuning of ToDS that are suitable for zero-shot and few-shot learning (e.g., five examples per label), including cross-lingual parameter-efficient fine-tuning [35]. Our contributions are the following:

1. We assemble a novel dataset: the first open-source ToDS dataset for Swahili-speaking Congolese in humanitarian crises, supporting intent classification, NLG, and slot filling.
2. We provide an extensive baseline for the dataset: Starting with the traditional pipeline ToDS, we perform a comparative study against modular end-to-end task-oriented dialogue strategies and various alternatives to full fine-tuning. To the best of our knowledge, this is the first attempt to establish baselines and evaluate Congolese Swahili in ToDS.

II. DATA

A. Congolese Swahili

Belonging to the Bantu branch of the Niger-Congo language family, Congolese Swahili is a regional dialect of Kiswahili, primarily spoken in the eastern regions of the DRC. It is also spoken in the border regions of neighboring countries such as Uganda, Rwanda, and Burundi. While it is mutually intelligible with standard Swahili (Kiswahili), Congolese Swahili differs in vocabulary, pronunciation, and syntax due to influences from French (the DRC's official language) and various local Bantu languages.

Congolese Swahili is a non-tonal agglutinative language written in the Latin script, following the conventions of standard Swahili to some extent [36]. However, as with many African languages, it is perceived as a low-resource language in the NLP context, particularly compared to high-resource languages such as English, Mandarin Chinese, or French. While Kiswahili has a relatively strong presence in East Africa and a growing body of digital resources, Congolese Swahili remains under-resourced, especially in its localized form. Although not rigidly defined, the term "low-resource language" has been interpreted by authors in [37] to describe languages lacking standardization, sufficient digital representation, linguistic resources, or NLP tools.

The linguistic diversity within the DRC makes developing NLP tools for Congolese Swahili particularly challenging. Despite its widespread use, the language lacks a fully standardized form across the country [38]. Although the Latin alphabet is used consistently, variation in spelling, grammar, and lexical choices is common. Moreover, code-switching between French and local languages complicates corpus development and language modeling.

As described by authors in [36], two distinct writing styles can be observed in Congolese Swahili:

- A standardized form, aligning with formal Kiswahili orthography, used in official documents, education, and some media.
- A vernacular form, which often disregards orthographic norms and is widespread in informal communication,

especially on social media, in text messages, and in everyday speech.

In this work, we focus on the standardized Latin-script form of Congolese Swahili due to its closer alignment with existing Swahili corpora and its suitability for NLP tasks.

B. Dataset

We collected a humanitarian domain-specific ToDS dataset in Congolese Swahili from the POLE FM radio call center, the SAUTI YA ENGILI radio call center, and a local Non-Governmental Organization (NGO) call center, Benevolencija.

We leveraged two community radio stations and one local NGO that stored users' SMSs and transcribed calls for emergency notification to higher official agencies. These stored SMSs and transcribed calls contained valuable information for creating a conversational agent dataset.

To extract the SMS and call data from the community radio stations, Excel sheets were used with five columns: name (radio station or local NGO), phone number (anonymized), date (message date), type (conversational type – SMS or call), and body (message content, either SMS or transcribed call).

Consequently, Python libraries such as Beautiful Soup, Requests, and regular expressions were applied to process the gathered Excel sheets and compile their content into a specific unified Excel sheet.

After collecting the corpus, the next step was preprocessing the gathered data to prepare them in a format suitable for various NLP applications. We used the following preprocessing steps:

- The compiled SMSs were converted into plain text, noisy characters and blank lines were cleaned up, and their encoding was changed to UTF-8 automatically to ensure readiness for training the system.
- All words were converted to lowercase to treat uppercase and lowercase forms as the same word.
- Duplicate questions were removed.
- Unwanted characters and words, including code-switched French words, were removed.

Moreover, after preprocessing the document, the words extracted from the questions and inquiries were mapped to their corresponding intents and entities. Table II shows the compiled corpus statistics for the Congolese Swahili dataset used in this study and its distribution in terms of size, intents, vocabulary, entities, tokens, and examples.

TABLE II. STATISTICS OF CONGOLESE SWAHILI NLU DATASET

Metric	Count
Number of unique intents	879
Number of unique entities	53
Number of responses	2,118
Number of questions	2,118
Number of tokens	15,019
Vocabulary size	1,958
Average tokens per question	7.09

Furthermore, the questions and inquiries required adequate responses. For this purpose, a humanitarian expert manually annotated all the 2,118 questions by providing humanitarian-specific responses. Figure 1 includes text samples from the structured ToDS dataset annotated with intents, entities, and human-provided responses. Table III presents the manually translated sample text for Figure 1, showing the Congolese Swahili dataset alongside its English translation.

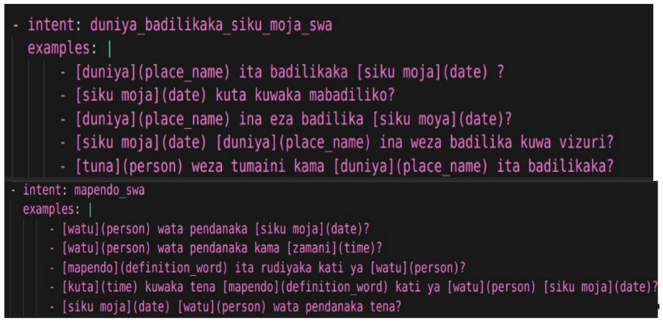

Fig. 1. Sample texts from the Congolese Swahili NLU dataset showing intent classification and entity annotations.

TABLE III. SAMPLE INTENTS, EXAMPLES, AND RESPONSES FROM THE CONGOLESE SWAHILI DATASET WITH ENGLISH TRANSLATIONS

Intent	Examples (Congolese Swahili)	Examples (English translation)	Response (Congolese Swahili)	Response (English translation)
duniya_badilika_siku_moja_swa / world_will_change_one_day	1. duniya ita badilika siku moja? 2. siku moja kuta kuwaka mabadiliko? 3. duniya ina eza badilika siku moya?	1. Will the world change one day? 2. Will there be changes one day? 3. Can the world change one day?	tunaimani kama dunia haita badilika ila wanadamu njo watabadilika	We believe the world will not change, but humans will change
mapendo_swa / love	1. watu wata pendanaka siku moja? 2. watu wata pendanaka kama zamani? 3. mapendo ita rudiyaka kati ya watu?	1. Will people love each other one day? 2. Will people love each other like in the past? 3. Will love return among people?	ndiyo kuko siku watapenda juu banaishi pamoja, na ni vema wapendane	Yes, there will be a day when they will love because they live together, and it is good to love one another

III. METHODOLOGY

In this section, we describe our experimental pipeline. First, we outline the data preprocessing steps. Next, we discuss the hyperparameters used in the experiments. We then present the experimental design of our ToDS model, followed by an in-depth discussion. We explain the three models trained on the dataset to select our baseline models. Afterwards, we demonstrate how an ontology dataset was utilized to improve the performance of our best baseline model. We also investigate Knowledge Base (KB) injection, where a prevalent formal query language, SPARQL, is utilized to query humanitarian data stored in Resource Description Framework (RDF) format, and we explore the final experimental method. Finally, we describe the evaluation metrics used to assess our ToDS models.

A. Data Preprocessing

Before training the ToDS models, we preprocessed the NLU and NLG datasets following procedures aligned with the Rasa framework [39]. Rasa is an open-source framework for conversational AI chatbot development. It offers tools and libraries for creating AI-based text and voice-driven chatbots capable of engaging in natural language conversations with users. The chatbot engine accepts one or more Excel files containing the bot's ontology (domain, dialogue data, and NLU). It then generates a functional Rasa chatbot from the files, as illustrated in Figure 2.

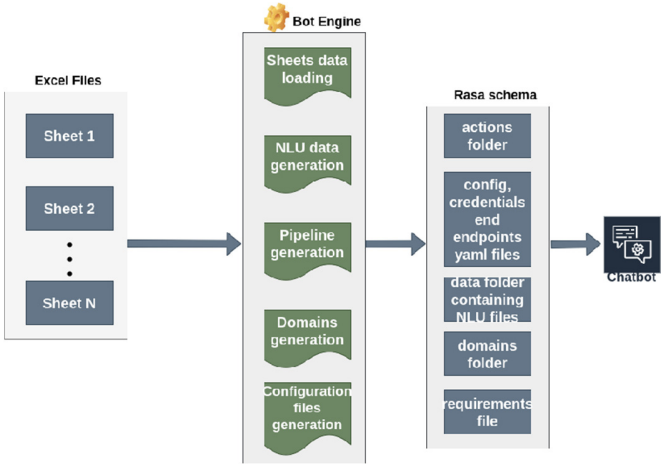

Fig. 2. Bot engine processing of Excel files to generate Rasa schema. Each Excel file represents a specific domain with multiple sheets corresponding to intents, and each sheet contains example data for the respective intent.

B. Hyperparameters

For our experiments, we selected hyperparameters based on the models used. We employed a sophisticated language-agonistic Bidirectional Encoder Representations from Transformers (BERT) variant pre-trained on 109 languages [40], making it suitable for low-resource language scenarios. The pipeline integrates Whitespace Tokenizer with several feature extraction methods, including RegexFeaturizer, lexicalSyntacticFeaturizer, and dual CountVectorsFeaturizers operating at both word and character n-gram levels (1-4 window sizes) to handle morphological complication efficiently.

For classification, it leverages a Dual Intent and Entity Transformer (DIET) model, a multi-task architecture that performs both intent classification and entity recognition by integrating pre-trained word and sentence embeddings with sparse features such as word-level and character-level n-grams. DIET then processes these features through feed-forward layers, two transformers, and a conditional random field layer. The model also applies masking, a regularization strategy that masks some tokens in the input sentence to improve generalization.

We trained for 100 epochs with adaptive batch sizes [16, 32] at a learning rate of 0.001 to balance stability and convergence speed. The dialogue management system combines memory-driven MemoizationPolicy, constraint-focused RulePolicy, and transformer-driven Transformer Embedding Dialogue Policy (TED), configured with a 5-turn historical window and 100 training epochs, developing a unified training approach across components. Because French is a resource-rich language, it is generally better represented in LaBSE than Swahili. This allows the model to make richer embeddings, effectively capitalizing on cross-lingual representation while adjusting to the distinctive traits of Congolese Swahili. This setup provides an optimal balance between modern deep learning methods and conventional NLP techniques for low-resource language processing. Figure 3 shows our architecture hyperparameters.

```
pipeline:
- name: WhitespaceTokenizer
- name: LanguageModelFeaturizer
  model_name: "bert"
  model_weights: "rasa/LaBSE"
- name: RegexFeaturizer
- name: LexicalSyntacticFeaturizer
- name: CountVectorsFeaturizer
- name: CountVectorsFeaturizer
  analyzer: "char_wb"
  min_ngram: 1
  max_ngram: 4
- name: DIETClassifier
  epochs: 100
  batch_size: [16, 32]
  learning_rate: 0.001
  learning_rate_decay: 0.1
  decay_type: "exponential"
- name: EntitySynonymMapper
- name: ResponseSelector
  epochs: 100
- name: FallbackClassifier
  threshold: 0.6
  ambiguity_threshold: 0.1

policies:
  Ctrl+L to chat, Ctrl+K to generate
- name: MemoizationPolicy
- name: RulePolicy
- name: TEDPolicy
  max_history: 5
  epochs: 100
assistant_id: 20250414-171321-grouchy-plywood
```

Fig. 3. Hyperparameters of the ToDS model.

C. Experimental Architecture

There are three models in our experimental architecture, namely the baseline, the knowledge-centered ontology, and the final ToDS models, as illustrated in Figure 4. The model's comprehensive description is explained in the subsections below. For training the baseline model, we utilized the accessible humanitarian domain-specific dataset of Section II, and for training the knowledge base ontology pipeline, we utilized a mix of humanitarian domain-specific datasets with SPARQL queries to the RDF data stored on Humanitarian Data Exchange (HDX). Finally, we fully fine-tuned the humanitarian ToDS dataset for rich evaluation.

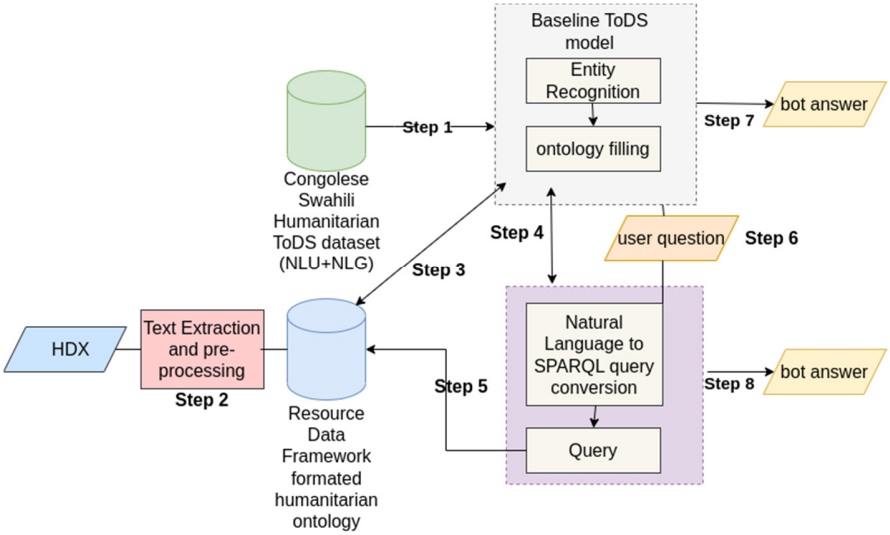

Fig. 4. Experimental architecture of the humanitarian ToDS model.

The procedure comprises eight steps:

- Step 1: Apply humanitarian domain-specific dataset and baseline model training.
- Steps 2 and 3: Produce the resource data framework and configure the humanitarian ontology data.
- Steps 4 and 5: Integrate the ontology knowledge base and the bot ontology filling via SPARQL query conversion for specialized answers.
- Step 6: User query either the baseline model or the specialized model.
- Steps 7 and 8: Bot answer generation and assessment on the test data.

1) Baseline Models

To determine the suitable model for subsequent experiments, we conducted three baseline experiments using the Congolese Swahili humanitarian ToDS dataset, as depicted in Step 1 of Figure 4. In our baseline experiments, we developed a Named Entity Recognition (NER) model to identify custom entities capable of decoding a humanitarian context and incorporating the humanitarian scope into the default bot ontology, leaving out HDX. We used the NLU pipeline and the DIET classifier, which functioned as its main element. The NER model's training set comprised Congolese Swahili humanitarian ToDS data sourced from community radio stations and local NGOs, with the model being trained on 53 custom entities. The NER model was consequently fine-tuned, utilizing more dialogue-related training examples, creating the system's NLU foundation via the Rasa incremental training method. Several NLU pipeline configurations, varying from simpler to more complex, were used and evaluated to ascertain the most appropriate alternative for our case. Therefore, these three baseline models were formed. Streamlit was used for comparing every configuration's outcome on the test set, using F1 score, recall, and precision as metrics. We tested diverse configurations in the featurization step, applying sparse and dense elements as inputs to the DIET classifier.

In Config_1, we combine sparse features with Language-agnostic BERT Sentence Embeddings (LaBSE) [40], demonstrating strong cross-lingual performance across 109+ languages, including several African languages. For Config_1_b, we utilized RoBERTa-base, a powerful transformer-based model that performs well for intent classification tasks. In Config_2, we relied solely on dense features from BERT multilingual-cased [41, 42], showing strong zero-shot cross-lingual transfer capabilities. Config_2_b employed SpaCy with the xx_ent_wiki_sm model, a lightweight multilingual model that supports Swahili and many other languages. This configuration uses SpaCy and sparse features for a balanced approach suitable for African languages. Finally, Config_3 utilized only sparse characteristics, making it the lightest and most deployment-friendly model, particularly ideal for low-resource environments commonly encountered in humanitarian settings. To enhance robustness, we trained all models for 100 epochs with masking activated in DIET's training. The configurations balance performance with practical deployment challenges in low-resource situations. Table IV summarizes the baseline configurations tested, including their use of sparse and dense features.

TABLE IV. HYPERPARAMETER CONFIGURATIONS FOR THE BASELINE MODELS

Configuration	Sparse features	Dense features
Config_1	Yes	Yes (LaBSE)
Config_1_b	Yes	Yes (RoBERTa-base)
Config_2	No	Yes (BERT multilingual-cased)
Config_2_b	Yes	Yes (SpaCy xx_ent_wiki_sm)
Config_3	Yes	No

2) Ontology Pipeline

This section delineates the primary elements of the pipeline that transforms a user question into a SPARQL query. An illustration of this transformation is presented in Figure 5.

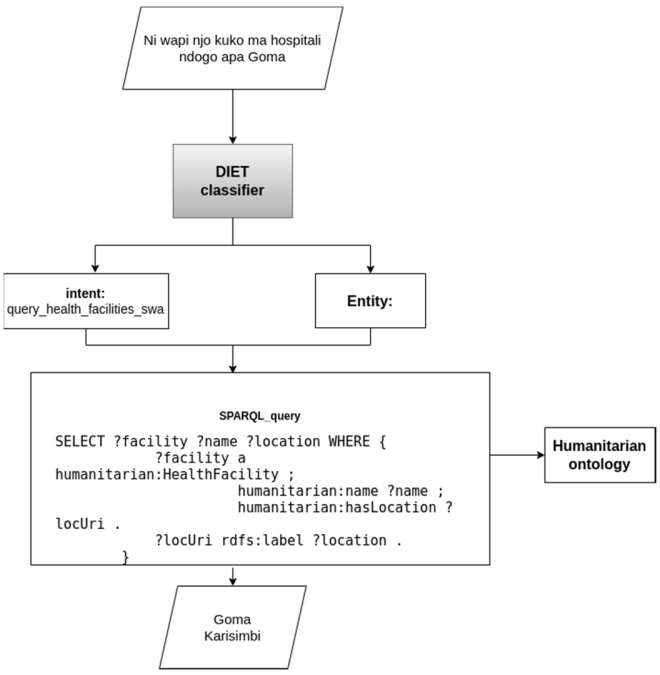

Fig. 5. Ontology pipeline.

a) Ontology Data Extraction and Preprocessing

At the start of the specialized dialogue, the bot acquires HDX datasets with a defined scope. From the Excel file obtained from HDX, Rasa extracts the humanitarian-related text and performs preliminary processing before sending the information to the entity recognition model for annotation. Text cleaning includes sentence-level tokenization, unit expansion, removal of unnecessary punctuation, and deletion of extra whitespace and tab characters. The system also preserves the designation of the humanitarian concept, which is from the directives.

b) Entity Recognition and Ontology Population

Following the preprocessing of the specialized humanitarian data, we import the trained NER model for the bespoke humanitarian-related entities. The humanitarian text, encompassing the details and directives, serves as input to the model. Upon the model annotating the text with the corresponding entities, these are utilized to populate the humanitarian ontology with the appropriate instances. The ontology is now finalized and prepared for querying.

c) SPARQL Query Transformation

When a user poses a question, the DIET classifier, which has been fine-tuned for entity recognition and intent classification, annotates the question with an intent and identifies any entities related to that intent to populate the corresponding Rasa slots. Upon identifying the intent, the corresponding Rasa action is activated, which subsequently relates to a SPARQL query. The query parameters are obtained from the filled slots. For instance, as illustrated in Figure 5, the user's question "ni wapi njo kuko ma hospitali ndogo apa Goma" is labeled by the DIET classifier with the intent query_health_facilities_swa and the entity request_type. The specified intent initiates the action action_query_health_facilities, responsible for locating health facilities in a designated area within the humanitarian ontology. The parameter for the query is obtained from the slot request_type, populated with the extracted entity. Prior to executing the query, the extracted entity string is compared against all facilities within the populated humanitarian ontology to identify the closest match. Upon identifying a match, the SPARQL query is executed. Upon retrieval of the results, we format the response and employ Rasa actions to convey it to the user in natural language.

D. Evaluation

We evaluated our model's efficacy based on the subsequent metrics:

- Intent and entity recognition: We evaluated the system's capability to identify intents and entities correctly. At each iteration, the F1 score, precision, and recall were computed. These metrics comprise the final scores achieved by the model.
- Task performance: We investigated the speed and adeptness with which our assistant performs a task. Metrics such as task completion rate and task completion cost were applied. The former assesses the system's capability to perform a task, quantified as the ratio of successful events to the total number of attempted tasks. The latter examines the system's productivity by computing the average number of dialogue turns required to complete a task. The ideal scenario for our system is responding in a single exchange without the user needing to repeat inquiries; hence, the optimal task completion cost is 1.
- Edit distance: Finally, we evaluated whether the system effectively achieved its primary objective: performing humanitarian crisis response. We asked five users to interact with the system until they reached the end of the humanitarian response objective. Then, we represented each dialogue in sequence based on the number of turns. For instance, a ground truth dialogue with four turns would be represented as $< h_1\ h_2\ h_3\ h_4 >$. Instances where the system asks the user to repeat the query or fails to perform a task are denoted as $< h_1\ h_2\ x\ h_3\ h_4 >$, with x indicating a repetition or a failure. Subsequently, each dialogue was

compared with its corresponding ground truth dialogue by calculating the edit distance using the following metric:

$$\text{Error} = \frac{(\text{substitutions} + \text{deletions} + 0.4 * \text{insertions})}{\text{actual turns}} \quad (1)$$

Substitutions occur when the system fails to execute a task, deletions occur when it offers no response, and insertions occur when the user is prompted to repeat. To emphasize situations where the system asks the user to repeat, we apply a correction factor of 0.4.

IV. RESULTS AND EVALUATION

This section presents the findings of our experiments. We first report the performance of the baseline ToDS models, followed by the outcomes of the ontology-based ToDS models querying the RDF format stored in HDX.

A. Baseline Models

As outlined in the methodology, we trained and assessed three baseline models on the ToDS dataset to identify the most effective model. Table V presents the performance of the NER model configurations in terms of F1 score, precision, and recall. The DIET model, trained exclusively on sparse features with masking enabled, achieved the highest F1 score, recall, and precision. Given that the F1 score is the weighted average of precision and recall, Config_3 was identified as the optimal model. We prefer the F1 score over accuracy because it better reflects performance in scenarios with imbalanced class distributions, as in our case.

Notably, the model that outperformed all others was also the least complex, trained solely on sparse features. Models trained with pre-trained embeddings required more computational resources and longer inference times. The cross-lingual transfer of sentence embeddings utilizing French and Standard Swahili dense features was ineffective, likely because these embeddings were not exposed to Congolese Swahili during pretraining. These results demonstrate that DIET is a suitable architecture for our low-resource application.

TABLE V. PERFORMANCE OF NER MODEL CONFIGURATIONS: F1 SCORE, PRECISION, AND RECALL (BEST MODEL TRAINED ON SPARSE FEATURES WITH MASKING)

Configuration	Precision	Recall	F1 score	Support
Config_3	0.709422	0.786223	0.733571	421.000000
Config_2_b	0.699327	0.776722	0.723040	421.000000
Config_1	0.667656	0.755344	0.695249	421.000000
Config_1_b	0.663895	0.738717	0.685669	421.000000
Config_2	0.610333	0.693587	0.636263	421.000000

B. Automated Evaluations

1) Intent Classification and Entity Recognition

The first phase of the automated evaluation assessed the system's ability to identify intents and entities accurately. While we previously evaluated the NLU model on annotated Congolese Swahili humanitarian datasets, we now evaluate the model finetuned on dialogue-related data. Due to the limited data, we have opted for a 5-fold cross-validation instead of a singular train-test split. In each fold, the model was trained on four groups and evaluated on the remaining group. At each iteration, precision, recall, and F1 score were computed and the mean of these metrics represents the model's final score. Cross-validation provides a fairer and less biased assessment of our model since the whole dataset is used for evaluation. Table VI demonstrates that the NLU model achieves very high scores across all metrics for both intent and entity recognition.

TABLE VI. PERFORMANCE COMPARISON OF DIFFERENT MODEL CONFIGURATIONS ON INTENT RECOGNITION AND ENTITY EXTRACTION (5-FOLD CROSS-VALIDATION AVERAGES)

Configuration	Intent recognition			Entity extraction		
	Precision	Recall	F1 score	Precision	Recall	F1 score
Config_1	0.765	0.754	0.739	0.904	0.899	0.898
Config_1_b	0.749	0.732	0.723	0.893	0.905	0.895
Config_2	0.662	0.621	0.607	0.859	0.840	0.845
Config_2_b	0.717	0.689	0.675	0.883	0.871	0.873
Config_3	0.768	0.766	0.755	0.918	0.921	0.919

2) Effectiveness and Capability

In the second step of the automated evaluation, we analyzed how efficiently and quickly our assistance can perform a task. For this purpose, we used the task completion rate and task completion cost metrics. Since the ultimate objective of our assistant is to ensure that users receive essential humanitarian assistance, we created a simulated assessment environment. However, it is challenging to quantify whether users have completed the task or how quickly they have done so. The progression of the discussion depends on how each user interacts with the system. For example, one user may be familiar with the surroundings and ask only a few questions, whereas another may ask several questions until the process is finished. To better understand our system's effectiveness, we recognized that a humanitarian discourse may include several activities. These activities include:

- Supplying further contextual information, which may consist of offering pertinent details or achieving a specific objective. This is referred to as provide_information.
- Responding to a straightforward inquiry entails providing an accurate response with little elaboration. This is designated as answer_query.
- Clarifying the connections between things involves integrating information to address an inquiry. This is designated as explain_relationship.

- Supplying the particulars of the entities, which may include providing further information on entities referenced in the inquiry. This is designated as provide_entity_details.
- Enumerating the entities without further information. This is referred to as list_entities.
- Other: This may pertain to a particular issue that encompasses the selection of many tasks from the aforementioned list.

The conversations used to calculate the specified metrics were obtained from both our interactions with the system and those of other users. In each interaction, we sought to comprehensively diversify the questions regarding structure and purpose to assess the assistant's performance. We gathered 20 conversations and manually marked system responses as accurate or incorrect, depending on the specified humanitarian situation. Refer to Algorithm 1 for the stages of the evaluation method. The mean number of tasks per dialogue was 3.75. Examples of annotated conversations are shown in Table VII. The outcomes from the second phase of the automated assessment are shown in Figures 6, 7, and 8.

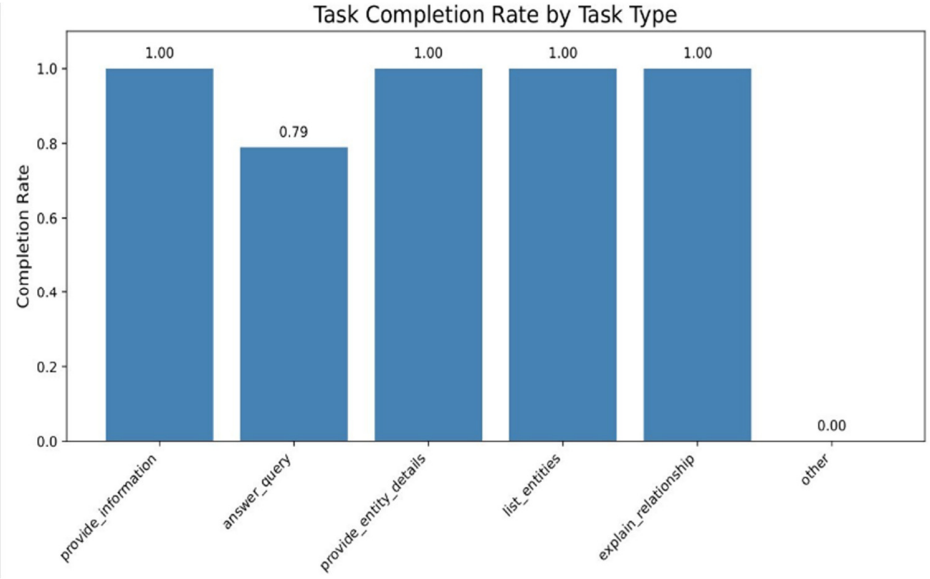

Fig. 6. Task completion rate by task type. Performance is perfect (1.0) for most task types, with only answer query showing lower performance (0.79) and other tasks not attempted (0.0).

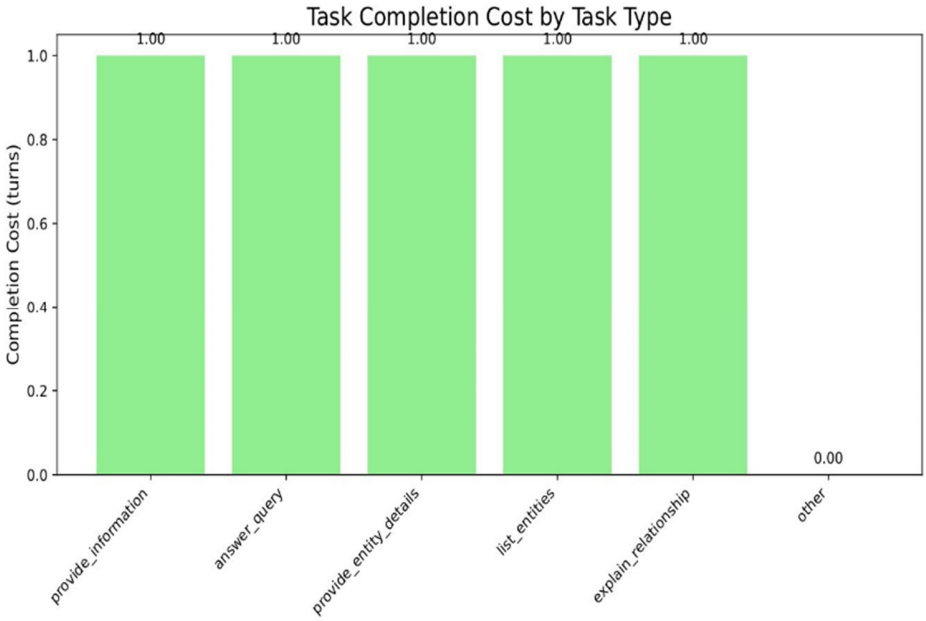

Fig. 7. Task completion cost by task type. All successfully completed tasks required exactly one turn to complete, indicating optimal efficiency across all task categories.

TABLE VII. DIALOGUE EVALUATION PROCESS EXAMPLES WITH TRANSLATIONS

Example	Evaluation step	Congolese Swahili	English
1	User turn	Kambi za wakimbizi wapi?	Where are the refugee camps?
	Bot turn	RUSAYO 2 (Rutshuru) - Uwezo: 2,280 KASHUGA (Rutshuru) - Uwezo: 10,976 MATER DEI (Rutshuru) - Uwezo: 460 MPATI (Rutshuru) - Uwezo: 10,976	RUSAYO 2 (Rutshuru) - Capacity: 2,280 KASHUGA (Rutshuru) - Capacity: 10,976 MATER DEI (Rutshuru) - Capacity: 460 MPATI (Rutshuru) - Capacity: 10,976
	Task identification	Kujibu swali Kutoa maelezo ya kitu Kutoa orodha ya vitu	Answer query Provide entity details List entities
	Task success	Yote yalifanikiwa (y/y/y)	All tasks were successful (y/y/y)
	Error type	0 - Hakuna kosa	0 - No error
	User turn	Naeza pata wapi chakula cha msaada?	Where can I get food aid?
2	Bot turn	PAM inapena chakula ya msaada kila wiki Kitchanga na Mweso.	WFP gives food aid every week in Kitchanga and Mweso.
	Task identification	Kujibu swali Kuelekeza eneo Kutaja shirika	Answer query Provide entity details List entities
	Task success	Yote yalifanikiwa (y/y/y)	All tasks were successful (y/y/y)
	Error type	0 - Hakuna kosa	0 - No error

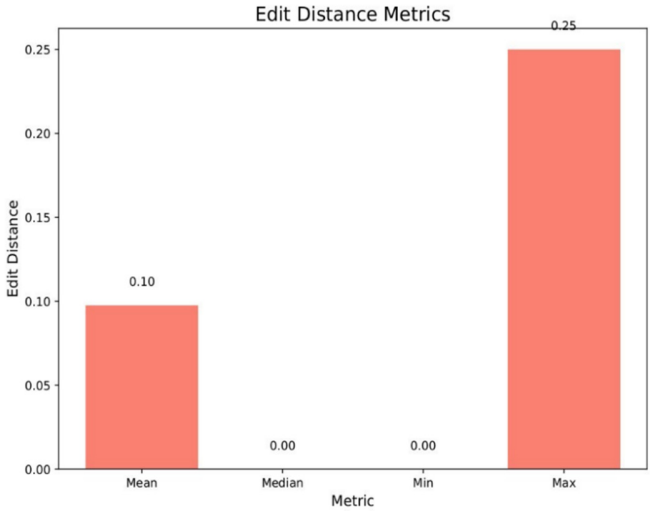

Fig. 8. Edit distance between actual and ideal dialogues. The mean edit distance of 0.10 indicates that 10% of turns required correction, with a maximum observed deviation of 0.25 in the most problematic dialogue.

```
Algorithm 1. Dialogue Evaluation
Procedure
Input  : A dialogue consisting of
  alternating User and Bot turns
Output : Annotated dialogue with task
  types, success status, error types, and
  ideal responses
1.  foreach TurnPair(UserTurn,BotTurn) in
    Dialogue do
2.     Identify all tasks in BotTurn:
3.      Task Types:
4.         1. Provide Information
5.         2. Answer Query
6.         3. Provide Entity Details
7.         4. List Entities
8.         5. Explain Relationship
9.         6. Other
10.    foreachIdentifiedTask in BotTurndo
11.       Annotator inputs task type (1-6)
12.       Annotator inputs task success
          (Yes/No)
13.    end
14.    Determine if there was an error in
       BotTurn:
15.     Error Types:
16.        0. None
17.        1. Substitution
18.        2. Deletion
19.        3. Insertion
20.    Annotator inputs error type
21. end
22. foreach BotTurn in Dialogue do
23.    Annotator decides: Does this turn need
       correction? (Yes / No)
24.    ifYes then
25.       Annotator writes the ideal
          corrected Bot response
26.    end
27. end
28. Store all annotations for the dialogue
```

Our system achieved a task completion rate of 79.8%, indicating that it successfully completed the majority of tasks. A significant finding during annotation was that the humanitarian ontology component contributed significantly to this high score. The NLU component of the assistant effectively addressed several erroneous outputs from the ontology segment while still providing a response. However, these responses were sometimes inaccurate. A common failure occurs when a user requests a precise answer; the system sometimes provides only a partial humanitarian response or produces false positives, likely due to the limited number of training instances for certain entities in the NLU model.

The system achieved a task completion cost score of 1.0, indicating high efficiency. This metric should be interpreted alongside Task Completion Rate, as it was calculated only from successfully completed tasks. The mean edit distance for all conversations was 0.1, demonstrating that user interactions

were highly efficient, with the majority of dialogue turns requiring no corrections.

V. DISCUSSION

We perform an error analysis of the module's performance and explore the research challenges and limitations. The lack of pretrained sentence embeddings in Congolese Swahili massively influenced intent classification more than entity recognition in the NLU module. In addition, the lack of a sentence embedder deters the precise understanding of sentence context in Congolese Swahili. This is because most intents are expressed as full sentences conveying meaning. Nevertheless, the usage of cross-lingual transfer shows modest improvement when leveraging pretrained standard Swahili (SpaCy features), compared with French cross-lingual transfer. Research demonstrates that Congolese Swahili embeddings are strongly linked with standard Swahili, despite frequent code-switching to French. Future work could explore integrating Masakhane embeddings [43], which are based on Niger-Congo B languages, into our ToDS pipeline.

The slot-filling module demonstrates accuracy across 53 entity types, with the top-performing entities being: population_type (1.0), rebel_name (1.0), and language_name (1.0), whereas currency_name (0.0) and measurement_type (0.0) are at the lowest end of the spectrum. The poor performance on currency_name and measurement_type is mainly due to the measurement units being incorrectly classified as different entities.

The superior performance of state-of-the-art systems is often linked to comprehensive training datasets and high-speed hardware like GPUs. We selected the DIET classifier-based method mainly due to the data volume and associated training costs. This work applies a conversational AI system in the humanitarian sector, utilizing a constrained dataset derived from SMS and phone communications, because of the challenges of obtaining large-scale humanitarian crisis data in Congolese Swahili. Therefore, a practical approach combines the DIET classifier model with a humanitarian ontology. Despite the dataset limitations, this represents an initial and promising outcome for agent modeling. Future research will focus on implementing prompting techniques and instructing large language models [44, 45] to improve the ToDS system's critical thinking and analysis capabilities.

Additionally, our ToDS technology's human evaluation is constrained by limitations. A practical, human-driven assessment is needed to evaluate the assistant's conversational quality through ratings provided by a specific number of human evaluators who interact with the system. This approach assesses discourse on fluency, relevance, informativeness, and appropriateness. Authors in [46] developed a recipe-centered ToDS with human review, where users were directed to choose a dish from a supported website and were urged to ask questions until they completed the entire recipe. To simulate realistic scenarios, they administered a questionnaire, evaluated the system's answers for this conversation type, verified whether the anticipated answers were provided, and assessed the dialogue's naturalness. In future studies, we aim to integrate additional human evaluation to strengthen real-world assessment metrics and improve the scalability of conversational systems.

VI. CONCLUSION AND FUTURE DIRECTIONS

We successfully implemented a humanitarian conversational Artificial Intelligence (AI) system designed to support users during humanitarian crises by addressing emergency-related questions. Our main focus for the system's design was the natural language to SPARQL conversation pipeline. To enable this query translation, we developed a humanitarian ontology from mined Humanitarian Data Exchange (HDX) datasets and trained a Named Entity Recognition (NER) model for specific humanitarian entities using a dataset of Congolese Swahili emergency crisis Short Message Service (SMS) messages and calls, employing Rasa's Natural Language Understanding (NLU) pipeline. We then fine-tuned this initial model with dialogue-related data to prepare the AI assistant for operation.

After creating our Task-Oriented Dialogue System (ToDS), we performed a thorough automated assessment. In this automated evaluation, we first assessed the effectiveness of the system in intent classification and entity extraction by calculating the F1 score, recall, and precision. We evaluated the assistant's ability to interact in efficient conversation by applying the task completion rate and task completion cost metrics, and compared our conversations with ground-truth conversations using the edit distance measure. The findings of this comprehensive evaluation indicated that the humanitarian ToDS can foster operational and efficient dialogue, thereby achieving satisfactory scores in dialogue quality.

As a preliminary measure for further studies, we find that it is advantageous to effectively use the dense features applicable to the Swahili language, namely the pre-training of Swahili sentence embeddings or the integration of Niger-Congo B pre-trained sentence embeddings from the MasaKhane community into our ToDS system.

The expansion of the ontology is an aspect that should be further considered for future growth. We could enlarge the ontology to incorporate more entities and complex interrelations among them. Accordingly, the system would accommodate more queries.

Eventually, we find it beneficial to conceptualize the entire process as a Question–Answer (QA) problem. A QA model might be refined using different humanitarian Congolese language conversation datasets obtained from emergency SMS messages and calls, utilizing large language model prompts or instructions. This would promote a wider range of humanitarian-related queries, mainly regarding the instruction element. However, this approach would impose a significant data collection burden because there is no existing annotated dataset for a large-scale humanitarian Congolese Swahili ToDS.

DATA AVAILABILITY STATEMENT

The Congolese Swahili task-oriented dialogue dataset introduced in this study was compiled from SMS and call center records provided by POLE FM, SAUTI YA ENGILI,

and the local NGO Benevolencija. To protect the privacy and security of affected individuals, personally identifiable information has been removed. The anonymized dataset, including intents, entities, and annotated dialogues in RDF format, is available for academic and non-commercial research purposes. It can be accessed at https://github.com/ussenuk/Congolese-swahili-TODS. Additionally, the experimental codebase can be accessed at https://github.com/ussenuk/Congolese-Swahili-TODS-ontology-pipeline. Researchers interested in using the dataset should cite this work.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the African Union (AU) for supporting this work. We also acknowledge the community radio stations and local NGOs that provided access to emergency call data, which were critical for this research.

REFERENCES

[1] L. Qin *et al*., "End-to-end Task-oriented Dialogue: A Survey of Tasks, Methods, and Future Directions," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore, 2023, pp. 5925–5941, https://doi.org/10.18653/v1/2023.emnlp-main.363.

[2] A. Arora, A. Shrivastava, M. Mohit, L. S.-M. Lecanda, and A. Aly, "Cross-lingual Transfer Learning for Intent Detection of Covid-19 Utterances." OpenReview, Aug. 12, 2020.

[3] S. Schuster, S. Gupta, R. Shah, and M. Lewis, "Cross-lingual Transfer Learning for Multilingual Task Oriented Dialog," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, MN, USA, 2019, pp. 3795–3805, https://doi.org/10.18653/v1/N19-1380.

[4] J. Ni, T. Young, V. Pandelea, F. Xue, and E. Cambria, "Recent advances in deep learning based dialogue systems: a systematic survey," *Artificial Intelligence Review*, vol. 56, no. 4, pp. 3055–3155, Apr. 2023, https://doi.org/10.1007/s10462-022-10248-8.

[5] L. Qin, X. Xu, W. Che, and T. Liu, "AGIF: An Adaptive Graph-Interactive Framework for Joint Multiple Intent Detection and Slot Filling," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online, 2020, pp. 1807–1816, https://doi.org/10.18653/v1/2020.findings-emnlp.163.

[6] L. Jacqmin, L. M. Rojas-Barahona, and B. Favre, "'Do you follow me?': A Survey of Recent Approaches in Dialogue State Tracking." arXiv, Jul. 29, 2022, https://doi.org/10.48550/arXiv.2207.14627.

[7] W.-C. Kwan, H.-R. Wang, H.-M. Wang, and K.-F. Wong, "A Survey on Recent Advances and Challenges in Reinforcement Learning Methods for Task-oriented Dialogue Policy Learning," *Machine Intelligence Research*, vol. 20, no. 3, pp. 318–334, Jun. 2023, https://doi.org/10.1007/s11633-022-1347-y.

[8] Y. Li, K. Yao, L. Qin, W. Che, X. Li, and T. Liu, "Slot-consistent NLG for Task-oriented Dialogue Systems with Iterative Rectification Network," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 97–106, https://doi.org/10.18653/v1/2020.acl-main.10.

[9] T. Zhao and M. Eskenazi, "Towards End-to-End Learning for Dialog State Tracking and Management using Deep Reinforcement Learning," in *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Los Angeles, CA, USA, 2016, pp. 1–10, https://doi.org/10.18653/v1/W16-3601.

[10] B. Peng, C. Li, J. Li, S. Shayandeh, L. Liden, and J. Gao, "Soloist: Building Task Bots at Scale with Transfer Learning and Machine Teaching," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 807–824, Aug. 2021, https://doi.org/10.1162/tacl_a_00399.

[11] Y. Yang, Y. Li, and X. Quan, "UBAR: Towards Fully End-to-End Task-Oriented Dialog System with GPT-2," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 16, pp. 14230–14238, May 2021, https://doi.org/10.1609/aaai.v35i16.17674.

[12] Y. Lee, "Improving End-to-End Task-Oriented Dialog System with A Simple Auxiliary Task," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic, 2021, pp. 1296–1303, https://doi.org/10.18653/v1/2021.findings-emnlp.112.

[13] H. Le, D. Sahoo, C. Liu, N. Chen, and S. C. H. Hoi, "UniConv: A Unified Conversational Neural Architecture for Multi-domain Task-oriented Dialogues," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online, 2020, pp. 1860–1877, https://doi.org/10.18653/v1/2020.emnlp-main.146.

[14] L. Qin, X. Xu, W. Che, Y. Zhang, and T. Liu, "Dynamic Fusion Network for Multi-Domain End-to-end Task-Oriented Dialog," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 6344–6354, https://doi.org/10.18653/v1/2020.acl-main.565.

[15] D. M. Eberhard, G. F. Simons, and C. Fennig, *Ethnologue: Languages of the World*, 22nd ed. Dallas, TX, USA: SIL International, 2019.

[16] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, and D. Klakow, "A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, 2021, pp. 2545–2568, https://doi.org/10.18653/v1/2021.naacl-main.201.

[17] D. Mbaye and M. Diallo, "Task-Oriented Dialog Systems for the Senegalese Wolof Language," in *Proceedings of the 31st International Conference on Computational Linguistics*, Abu Dhabi, UAE, 2025, pp. 4803–4812.

[18] "Chatbots against COVID-19: Using chatbots to answer questions on COVID-19 in the user's language." World Health Organization. https://www.who.int/news-room/feature-stories/detail/scicom-compilation-chatbot.

[19] A. Stopper. "Lifting Up Women Through Land Ownership." Mozilla Foundation. https://www.mozillafoundation.org/en/blog/lifting-up-women-through-land-ownership/.

[20] "Common Voice Kiswahili Awards." Mozilla Foundation. https://www.mozillafoundation.org/en/what-we-fund/programs/common-voice-kiswahili-awards/awards/.

[21] A. Stopper. "Growing Skills, Confidence, and 'Quality Potatoes' in Tanzania." Mozilla Foundation. https://www.mozillafoundation.org/en/blog/growing-skills-confidence-and-quality-potatoes-in-tanzania/.

[22] A. Stopper. "'Wezesha na Kabambe:' Building Tech With Smallholder Farmers in Western Kenya." Mozilla Foundation. https://www.mozillafoundation.org/en/blog/wezesha-na-kabambe-building-tech-with-smallholder-farmers-in-western-kenya/.

[23] A. Stopper. "Chatbots, Local Weather Reports, and a Boon for Kenya's Smallholder Farmers." Mozilla Foundation. https://www.mozillafoundation.org/en/blog/chatbots-local-weather-reports-and-a-boon-for-kenyas-smallholder-farmers/.

[24] "Back in the Early Days...." Woebot Health. https://woebothealth.com/back-in-the-early-days/.

[25] Digital-Umuganda. "Mbaza-chatbot: Kinyarwanda-chatbot." GitHub. https://github.com/Digital-Umuganda/Mbaza-chatbot/tree/master/Kinyarwanda-chatbot.

[26] "Nivi Platform." Nivi. https://www.nivi.io/platform.

[27] "Annual Report 2024." Reach Digital Health. https://www.reachdigitalhealth.org/annual-report-2024.

[28] "Crowdsourcing Solutions to Empower Communities." Ushahidi. https://www.ushahidi.com//.

[29] N. N. Amir Sjarif, N. F. Mohd Azmi, S. Chuprat, H. M. Sarkan, Y. Yahya, and S. M. Sam, "SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm," *Procedia Computer Science*, vol. 161, pp. 509–515, Jan. 2019, https://doi.org/10.1016/j.procs.2019.11.150.

[30] D. A. Oyeyemi and A. K. Ojo, "SMS Spam Detection and Classification to Combat Abuse in Telephone Networks Using Natural Language Processing," *Journal of Advances in Mathematics and Computer Science*, vol. 38, no. 10, pp. 144–156, Oct. 2023, https://doi.org/10.9734/jamcs/2023/v38i101832.

[31] O. Okolloh, "Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information," *Participatory learning and action*, vol. 59, pp. 65–70, 2009.

[32] L. Palen, S. R. Hiltz, and S. B. Liu, "Online forums supporting grassroots participation in emergency preparedness and response," *Communications of the ACM*, vol. 50, no. 3, pp. 54–58, Mar. 2007, https://doi.org/10.1145/1226736.1226766.

[33] P. Meier, *Digital Humanitarians: How Big Data Is Changing the Face of Humanitarian Response*, 1st ed. New York, NY, USA: Routledge, 2015.

[34] J. FitzGerald *et al.*, "MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada, 2023, pp. 4277–4302, https://doi.org/10.18653/v1/2023.acl-long.235.

[35] T. Bunk, D. Varshneya, V. Vlasov, and A. Nichol, "DIET: Lightweight Language Understanding for Dialogue Systems." arXiv, May 11, 2020, https://doi.org/10.48550/arXiv.2004.09936.

[36] A. Öktem, E. DeLuca, R. Bashizi, E. Paquin, and G. Tang, "Congolese Swahili Machine Translation for Humanitarian Response." arXiv, Mar. 19, 2021, https://doi.org/10.48550/arXiv.2103.10734.

[37] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication*, vol. 56, pp. 85–100, Jan. 2014, https://doi.org/10.1016/j.specom.2013.07.008.

[38] U. Kimanuka, C. wa Maina, and O. Büyük, "Speech recognition datasets for low-resource Congolese languages," *Data in Brief*, vol. 52, Feb. 2024, Art. no. 109796, https://doi.org/10.1016/j.dib.2023.109796.

[39] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open Source Language Understanding and Dialogue Management." arXiv, Dec. 15, 2017, https://doi.org/10.48550/arXiv.1712.05181.

[40] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic BERT Sentence Embedding," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland, 2022, pp. 878–891, https://doi.org/10.18653/v1/2022.acl-long.62.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, MN, USA, 2019, pp. 4171–4186, https://doi.org/10.18653/v1/N19-1423.

[42] A. D. Vincentio and S. Hansun, "A Fine-Tuned BART Pre-trained Language Model for the Indonesian Question-Answering Task," *Engineering, Technology & Applied Science Research*, vol. 15, no. 2, pp. 21398–21403, Apr. 2025, https://doi.org/10.48084/etasr.9828.

[43] D. I. Adelani *et al.*, "IrokoBench: A New Benchmark for African Languages in the Age of Large Language Models," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Albuquerque, NM, USA, 2025, pp. 2732–2757, https://doi.org/10.18653/v1/2025.naacl-long.139.

[44] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing," *ACM Computing Surveys*, vol. 55, no. 9, Jan. 2023, Art. no. 195, https://doi.org/10.1145/3560815.

[45] B. Xu *et al.*, "ExpertPrompting: Instructing Large Language Models to be Distinguished Experts." arXiv, Mar. 05, 2025, https://doi.org/10.48550/arXiv.2305.14688.

[46] E. P. Tsili, "A Voice Assistant for cooking based on a natural language to SPARQL transformation pipeline using Rasa," M.S. thesis, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens, Greece, 2023.