



REPUBLIC OF BENIN
MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH OF BENIN

UNIVERSITY OF ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE RECHERCHE EN
INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88
<https://www.ifri-uac.bj> Email : contact@ifri.uac.bj



THESIS

for

Master's degree in computer science

Option : Software Engineering

Presented by :

Dave Lionel Sourou **DASSI**

Chatbot for requesting public administration services in Benin.

Supervised by :

Dr Msc John **AOGA**

Examination committee :

Pr Eugène C. EZIN	Full Professor	IFRI,IMSP/UAC	President
Dr. Msc. Ratheil HOUNDI	Senior Lecturer	IFRI/UAC	Examiner
Dr. Msc. John AOGA	Teacher-Researcher	IFRI/UAC	Secretary

Academic year : **2022 - 2023**

Summary

Dedication	ii
Acknowledgements	iii
Résumé	iv
Abstract	v
List of Figures	vi
List of Tables	viii
List of Acronyms	ix
General Introduction	1
1 Related works	3
2 Material and Methods	21
3 Results and discussions	37
Conclusion & Perspectives	51
Bibliography	52
Appendix	56
Contents	64

Dedication

TO MY DEAR PARENTS, BROTHERS AND SISTERS

Acknowledgements

I would like to thank :

- Almighty God for giving me the courage, strength and patience to complete this modest work;
- Professor Eugène C. Ezin, Director of the Institut de Formation et de Recherche en Informatique (IFRI);
- Dr John Aoga, supervisor of this thesis, for his patience, availability and, above all, his sound advice, which contributed to my thinking and his encouragement during the completion of this dissertation;
- Mr Arnould Adjovi, for his invaluable contribution and support throughout the preparation of this dissertation. His generosity in sharing his expertise was of vital importance to the progress of this work;
- each of the professors who supervised us throughout our training at IFRI, for having given us the necessary lessons to enable us to better launch ourselves into the professional and research world;
- all our fellow students at IFRI, with whom we have spent two years sharing our experiences in a convivial atmosphere;

Résumé

Le Bénin, comme d'autres pays africains, s'est engagé dans la digitalisation de son administration. Plus de 250 services sont désormais disponibles sur le portail des services publics. Cependant, il est difficile de s'y retrouver car il n'existe aucun outil pour aider les citoyens à exprimer leurs besoins et à trouver les services correspondants. C'est ainsi que nous avons mis en place CHAT BJ, notre chatbot conçu pour aider les citoyens dans l'identification des services en fonction de leurs besoins et leur fournir les informations nécessaires pour en bénéficier. CHAT BJ utilise une architecture de type "Retrieval-Augmented Generation" (RAG). Cela signifie qu'il peut d'abord extraire des informations pertinentes à partir d'une base de données de connaissances, puis générer une réponse précise et factuelle à l'aide d'un modèle de langage (LLM). Pour ce travail nous avons utilisé Mistral-7b-instruct-v0.1 comme modèle génératif. Dans le but d'améliorer la pertinence des informations extraites, nous avons utilisé une approche de recherche hybride combinant la recherche sémantique et lexicale, puis nous avons intégré l'utilisation d'un modèle re-ranker. Ce modèle re-classe les informations pertinentes extraites en fonction de leur pertinence par rapport à la requête de l'utilisateur avant que le modèle génératif ne génère une réponse. Les résultats ont montré que CHAT BJ est plus performant que d'autres chatbots populaires tels que ChatGPT et Bard dans la génération des réponses liées aux services de l'administration béninoise. Il génère des réponses plus pertinentes et de meilleure qualité, et atteint même les performances d'un gpt3.5-turbo dans les mêmes conditions.

Mots clés : Administration, Chatbot, services publics, LLM, RAG.

Abstract

Following the digitalization trend in Africa, Benin boasts over 250 services on its public services portal. However, navigating these options is challenging due to the lack of a tool that assists citizens in expressing their needs and finding the relevant services. To bridge this gap, we present CHAT BJ, a chatbot designed to identify services based on user needs and provide the information required to utilize them. CHAT BJ leverages a Retrieval-Augmented Generation (RAG) architecture, allowing it to extract pertinent information from a knowledge base before generating accurate and factual responses using a generative model. In this work we used Mistral-7b-instruct-v0.1 as a generative model. In order to improve the relevance of extracted information, we used a hybrid search approach combining semantic and lexical search, then integrated the use of a re-ranker model. This model re-ranks the relevant information extracted according to its relevance to the user's query, before the generative model generates an answer. The results showed that CHAT BJ outperformed other popular chatbots such as ChatGPT and Bard in generating responses related to Beninese government services. CHAT BJ generates superior quality and relevant responses, even rivaling the performance of a gpt3.5-turbo under similar conditions.

Key words: Digitalization, chatbot, public services, LLM, RAG.

List of Figures

1.1	Investment in electronic communications networks networks (in billions of CFA francs)[2].	5
1.2	Architecture of a Transformer model [5].	8
1.3	Illustration of how a T5 model works [7].	10
1.4	Representation of a word embedding of dimension 4 [10].	11
1.5	Conversation with ELIZA [10].	12
1.6	Dr. Sbaitso's interface [15].	13
1.7	Example of a discussion with ALICE [11].	13
1.8	Diagram illustrating the different approaches to chatbots [18].	15
1.9	Diagram illustrating the different chatbots techniques base on the approaches[18].	17
1.10	ChatGPT user interface.	18
1.11	Bard user interface.	18
1.12	Conversation with ChatGPT about the process of applying for a passport in Benin	20
1.13	Conversation with Bard about the process of applying for a passport in Benin	20
1.14	Conversation with Bard about the process of applying for a passport in Benin - 2	20
2.1	Cycle of our chatbot development	22
2.2	Functional architecture of the system	23
2.3	Benchmarking of Dense Passage Retriever (DPR) and BM25 on entity questions and Natural Questions (NQ) [27].	25
2.4	illustration of web scrapping [30].	26
2.5	Steps for using a vector store [33].	29
2.6	Representation of an adapter in transformer [34].	30
2.7	Representation of LoRA in transformer [34].	30
2.8	Representation of Prompt tuning in transformer [34].	31
2.9	ROUGE-1 formula [38].	33
2.10	ROUGE-2 formula [38].	33
2.11	ROUGE-L formula [38].	33
2.12	BERTScore formula [40].	34
3.1	Document size distribution before chunking	38
3.2	Document size distribution after chunking	38
3.3	Details of our dataset after processing	38
3.4	Sample of acronyms questions	40
3.5	Mixed dataset for hybrid search evaluation	40

3.6 Sample of SQuADfr dataset	41
3.7 The components of our training set	41
3.8 Fine-Tuning Results	44
3.9 Our system's response to the request "How do I apply for a CIP ?"	46
3.10 ChatGPT's response to the request "How do I apply for a CIP ?"	46
3.11 Bard's response to the request "How do I apply for a CIP ?" - 1	47
3.12 Bard's response to the request "How do I apply for a CIP ?" - 2	47
3.13 Our system's response to the request "How do I apply for my IFU ?" - 1	48
3.14 Our system's response to the request "How do I apply for my IFU ?" - 2	48
3.15 ChatGPT's response to the request "How do I apply for my IFU ?"	49
3.16 Bard's response to the request "How do I apply for my IFU ?" - 1	49
3.17 Bard's response to the request "How do I apply for my IFU ?" - 2	50
3.18 Illustration of deep learning and machine learning performance[41].	56
3.19 Architecture of an artificial neuron [42].	57
3.20 Deep neural network topology [42].	57

List of Tables

1.1	Limits of existing chatbots on public services requests in Beninese administration	19
2.1	Characteristics of embedding models [31]	26
2.2	Characteristics of re-ranking models	27
2.3	Characteristics of LLMs [32]	27
2.4	Retriever's metrics	32
2.5	Confusion matrix for retrieval evaluation	32
2.6	Tools description	35
3.1	corpus after collecting data	37
3.2	The statistics for our training and test sets	41
3.3	Embeddings models evaluation on naturals questions	42
3.4	Cost of model use	42
3.5	Performance of the hybrid approach compared to the semantic approach of our retriever	43
3.6	Performance of re-rankers on our mixed dataset of questions.	43
3.7	Comparison of the performance of our system without and with the re-ranker	43
3.8	Evaluation of some generative models on the SQuADfr dataset	44
3.9	Comparative Evaluation of Mistral-7b-instruct Models on test set	44
3.10	Comparison of Mistral-7b-instruct and GPT3.5 on test set	45
3.11	Comparison of Mistral-7b-instruct and GPT3.5 Latency	45

List of Acronyms

ABSU-CEP :

Agence Béninoise de Service Universel des Communications Électroniques et de la Poste

ACID :

Atomicity, Consistency, Isolation, Durability

ADN :

Agence pour le Développement du Numérique

AI :

Artificial Intelligence

AIM :

AOL Instant Messenger

AIML :

Artificial Intelligence Markup Language

ANIP :

Agence Nationale d'Identification des Personnes

ANN :

Artificial Neural Networks

ANSSI :

Agence Nationale de Sécurité des Systèmes d'Information

ARCEP :

Autorité de Régulation des Communications Électroniques et de la Poste

ASIN :

Agence des Systèmes d'Information et du Numérique

ASSI :

Agence des Services et Systèmes d'Information

BART :

Bidirectional and Auto-Regressive Transformers

BERT :

Bidirectional Encoder Representations from Transformers

BLEU :

Bilingual Evaluation Understudy

BM25 :

Best Matching 25

CI/CD :

Continuous Integration/Continuous Deployment

CNHU-HKM :

Centre Hospitalier et Universitaire Hubert Koutoukou Manga

CNN :

Convolutional Neural Networks

DPR :

Dense Passage Retriever

FFN :

Feed Forward Network

GB :

Giga Bytes

GPT :

Generative Pre-trained Transformer

ICTs :

Information and Communication Technologies

IDE :

Integrated Development Environment

LDA :

Latent Dirichlet Allocation

LLM :

Large Language Model

LoRA :

Low-Rank Adaptation

MEF :

Ministry of the Economy and Finance

METEOR :

Metric for Evaluation of Translation with Explicit ORdering

ML :

Machine Learning

MND :

Ministère du Numérique et de la Digitalisation

MTEB :

Massive Text Embedding Benchmark

NQ :

Natural Questions

OIV :

Vital Importance Operators

PEFT :

Parameter-Efficient Fine Tuning

QLoRA :

Quantized Low-Rank Adaptatio

RAG :

Retrieval-Augmented Generation

ReLU :

Rectified Linear Unit

RLHF :

Reinforcement Learning from Human Feedback

RNN :

Recurrent Neural Networks

RoBERTa :

Robustly optimised BERT approach

ROUGE :

(Recall-Oriented Understudy for Gisting Evaluation

SBIN :

Société Béninoise des Infrastructures Numériques

SENIA :

Salon de l'Entrepreneuriat Numérique et de l'Intelligence Artificielle

SNIAM :

Stratégie Nationale d'Intelligence Artificielle et des Mégadonnées

T5 :

Text-To-Text Transfer Transformer

XML :

Extensible Markup Language

General Introduction

1. Context

The rapid evolution of Benin towards the digitization of administrative services represents a significant advance in the socio-economic landscape. With over 250 services now accessible online, the Beninese government has succeeded in simplifying interactions between citizens and the administration, thereby reducing the constraints associated with travel and waiting lines. However, Despite the availability of over 250 digitally accessible services, citizens often find themselves faced with a crucial challenge.

The diversity of services, the multiplicity of administrative structures, and the constant evolution of regulatory requirements create a complex landscape for citizens seeking to navigate this new digital environment. The risk of errors in the procedures and the extended delays resulting from insufficient understanding of the procedures underline the urgent need to find innovative solutions.

2. Problem statement

Finding your way around all these new services is a real challenge, because there is no help tool to make it easier to find the links to these services or to get the needed very quickly. The main issue of this research is therefore how a specialised chatbot can be designed to respond effectively and adaptively to the information needs of Beninese citizens. How can this chatbot overcome the obstacles associated with the complexity of administrative procedures, offer relevant advice and help to simplify access to online public services? Our motivation is to optimise the citizen experience and consolidate the foundations of a more inclusive and efficient digital administration in Benin.

3. Objectives

The main objective of this research is to develop a conversational agent (chatbot) designed for administrative service requests within the Beninese public administration. The goal is to implement an innovative tool that facilitates access to clear and precise information, thus contributing to optimizing the process of informing citizens about online administrative procedures.

Specific objectives are:

- **Analyze citizen needs:** Understand the expectations and gaps in information about administrative procedures in order to design a chatbot that can adaptively respond to users' specific requests.
- **Implement mechanisms for collecting and processing administrative data:** Establish efficient processes for collecting, updating, and processing administrative information in order to ensure the relevance and accuracy of the responses provided by the chatbot.
- **Design a chatbot architecture:** Develop a chatbot structure that takes into account the specificities of the Beninese public administration, integrating a contextual understanding of the services offered and the procedures to follow.
- **Evaluate chatbot performance:** Measure the chatbot's effectiveness in terms of the accuracy of responses, speed of execution, and user satisfaction, using appropriate metrics to ensure an objective evaluation.

4. Thesis Organisation

This document will be structured around three main parts that aim to explore exhaustively the challenges and solutions related to the implementation of a chatbot for administrative services in Benin.

- The first chapter will be dedicated to the state of the art, exploring the foundations of transformers and chatbots, while providing a detailed overview of the Beninese public administration.
- The second chapter will focus on methods and solutions, exposing the technological choices, data collection and processing processes, as well as the chatbot's logical architecture.
- Finally, the third chapter will focus on the analysis of the results, including the evaluation of the models and some screenshots of our system's answers about administration public services in Benin.

Related works

Introduction

This section is devoted to explore Benin’s public administration, identifying its structure, organisation, missions, and services offered to citizens. We will also examine the state of the art in the field of chatbots, highlighting significant advances in the areas of Deep Learning, Transformers and chatbot applications. In addition, we will review related work in the field, highlighting current successes and gaps, in order to provide a comprehensive framework for our research aimed at developing an innovative service request chatbot for public administration in Benin.

1.1 Digitalization in Benin

Since 2016, Benin has opted for a digital transformation of its administration, placing Information and Communication Technologies (ICTs) at its heart. With the use of ICT tools, Benin’s public administration has been modernised and public services digitalised. In this chapter, we will describe the reforms and investments made in the digital field, and then describe the state of the art of Digitalised Public Services.

1.1.1 Legal and institutional reforms of the digital economy in Benin

Since 2016, Benin has been committed to making digital technology and the digitization of public services an engine for economic growth and social inclusion, materialized by the adoption of the digital code through Law no. 2017-20 of April 20, 2018^[1]. This code aims to regulate the digital sector and facilitate the adoption of digital services by all.

At institutional level, the Beninese government has set up bodies dedicated to the development and implementation of the digital strategy. The Ministère du Numérique et de la Digitalisation (MND) has been created to oversee policy in this area, while entities such as

¹https://apdp.bj/wp-content/uploads/2019/04/CODE-DU-NUMERIQUE-DU-BENIN_2018-version-APDP.pdf

the Autorité de Régulation des Communications Électroniques et de la Poste (ARCEP) and the Agence Béninoise de Service Universel des Communications Électroniques et de la Poste (ABSU-CEP) are responsible for regulating electronic communications and promoting digital access throughout the country. Outside the Ministry, the Agence pour le Développement du Numérique (ADN), the Agence des Services et Systèmes d'Information (ASSI) and the Agence Nationale de Sécurité des Systèmes d'Information (ANSSI) have been created and placed under the authority of the Presidency of the Republic.

For reasons of rationality, efficiency and consistency of public action, Decree N° 2022-324 of June 1, 2022 merging the ADN, ASSI, ANSSI and ABSU-CEP agencies to form the Agence des Systèmes d'Information et du Numérique (ASIN) was adopted by the Council of Ministers on June 1, 2022. ASIN is placed under the dual authority of the Ministry of the Digital Economy and Digitization (MND) and the Ministry of the Economy and Finance (MEF).

The Agence Nationale d'Identification des Personnes (ANIP), which reports to the President of the Republic, is responsible for managing the digital and biometric identity of individuals throughout the country, and for producing documents and integrated systems relating to identity.

Société Béninoise des Infrastructures Numériques (SBIN) has been created to continue the internet access activities of Bénin Télécoms Services. In parallel, the Autorité de Protection des Données Personnelles (APDP) is responsible for guaranteeing the protection of personal information. It monitors the compliance of personal data processing with current legal requirements.

1.1.2 Development indicators for the digital economy to the end of 2022

The digital economy is booming, and Benin is no exception. In 2022, the country saw strong growth in its digital development indicators, particularly in terms of Internet access, use of digital technologies by businesses and individuals, and job creation in the digital sector.

Benin's mobile operators have increased their investment in electronic communications networks in 2022, from 66.40 billion CFA francs to 119.38 billion CFA francs. This 79.8% increase reflects the operators' determination to meet their quality of service and network coverage obligations. This volume of investment is divided between mobile and fixed electronic communications networks, with respective rates of 52.3% and 47.7%^[2].

The graph below illustrates the evolution of investment in electronic communications communications networks in Benin from 2019 to 2022.



Figure 1.1: Investment in electronic communications networks networks (in billions of CFA francs)[2].

By the end of 2022, the number of active SIM cards connected to mobile electronic communications networks in Benin had increased by 14.28% compared with the previous year. However, the number of subscribers (after elimination of multi-SIMs) to mobile telephony is estimated at 6,811,527 on December 31, 2022, representing a mobile penetration rate of 52.7%citearcep.

The growth of the mobile market in Benin in 2022 compared with 2021 is reflected in increased traffic on all services:

- Voice traffic increased by 2.8% to 6.488 billion call minutes.
- The volume of SMS messages exchanged rose by 11.6%, to 601.877 million messages.
- Mobile Internet traffic rose by 46.5% to 203.070 million GB.

By 2022, the number of mobile Internet SIM cards connected to the networks of electronic communications operators in Benin has reached 9,727,180, an increase of 15.46% compared with 2021. The number of mobile Internet subscribers is estimated at 5,435,257, representing a mobile Internet penetration rate of 42.1%. This growth is mainly due to the increase in the number of 3G and 4G users. Indeed, the proportion of users of these technologies has risen from 51.47% in 2021 to 57.87% in 2022[2].

1.1.3 Digitalization of public services in Benin: Smart Gouv

The digitization of public services is a global trend that aims to improve the efficiency and accessibility of services offered by the state to citizens and businesses. In Benin, the government launched the Smart Gouv program in 2016, which aims to modernize public administration and make public services more accessible and efficient.

The Beninese government has set up a national platform for online public services ² with the aim of providing simplified, efficient access for citizens, while improving governance.

²<https://service-public.bj/public/services/e-services>

The platform offers over 1140 public services, 261 of which are digitized, enabling online application, payment and access. For some services, online application and payment are possible, but final access requires a physical presence. Other services are only available for online consultation, requiring a physical request.

These public services cover various fields, including civil status, water, energy, finance, civil service, education and health. For civil status, the platform offers applications for birth certificates, digital identity cards, passports, electronic visas and more. It also facilitates connection and bill payment requests for water and electricity services.

In the financial and civil service sectors, the platform simplifies tax declarations, tax payments and access to pay slips and career records for civil servants. This digitization makes it easier for individuals and companies to pay their taxes, and for government employees to manage their retirement files without major constraints.

As for education, the government aims to improve the quality of education by connecting research centers and universities, offering online training, and equipping primary and secondary schools, in particular by providing digital classrooms in high schools for girls. Although not all objectives have yet been achieved, several universities are already connected and the results of national examinations (CEP, BEPC and BAC) can be consulted online. BAC) can be consulted online.

In the healthcare sector, digitalization is still at an early stage, but some services such as appointment booking at the Centre Hospitalier et Universitaire Hubert Koutoukou Manga (CNHU-HKM) and Covid-19 tests (appointment booking, payment and results collection) are available online. This national e-services platform thus represents a major step forward in the modernization of public services in Benin, enabling greater accessibility for citizens and greater efficiency in administrative management.

1.1.4 National Strategy for Artificial Intelligence and Megadata

Adopted in January 2023, the National Strategy for Artificial Intelligence and Megadata (SNIAM)³ aims to make Benin a major AI player in West Africa by 2027. One of the key objectives of this strategy is to develop sectoral applications of AI in key areas such as public administration.

The development of chatbots for the Beninese administration fits perfectly into the SNIAM framework. Chatbots can help to improve the efficiency and quality of public services, reduce costs and improve citizen satisfaction. There is considerable potential for the use of chatbots in the Beninese administration:

- Citizen information: Chatbots can provide information and answer questions from citizens on a variety of subjects, such as administrative procedures, available services, steps to take, etc.
- Task automation: Chatbots can automate simple and repetitive tasks, such as making appointments, requesting documents, paying bills, etc.

³<https://numerique.gouv.bj/assets/documents/strategie-nationale-d'intelligence-artificielle-et-des-megadonnees-2023-2027.pdf>

- **Improved communication:** Chatbots can act as a two-way communication channel between the administration and citizens, enabling opinions and suggestions to be collected, information to be disseminated and citizens' concerns to be addressed.

1.2 NLP and Transformers in AI's Pinnacle

Natural Language Processing (NLP) explores the use of computers to understand and generate human languages also called natural languages, with the aim of executing practical tasks.

NLP involves the development of algorithms and models that enable computers to interpret, analyze, and generate human-like text. This includes tasks such as speech recognition, machine translation, sentiment analysis, chatbot etc. By bridging the gap between human communication and machine understanding, NLP contributes to the creation of intelligent systems capable of interacting with users in a way that mimics human language comprehension.

The field of NLP is experiencing a surge in popularity, propelling it to the forefront of Artificial Intelligence (AI). This rapid advancement can be attributed to the power of deep learning, particularly the effectiveness of the "Transformer" architecture that underpins most NLP models.

Neural networks [3.3](#) are very effective statistical models for analysing complex data with variable formats [\[3\]](#). While models such as CNNs emerged for visual data processing, for text processing, the neural network architectures that were used were RNNs, and more specifically LSTMs.

LSTM networks, a type of recurrent neural network (RNN), have been specifically designed to solve the gradient vanishing problem found in standard RNNs. They have the ability to learn and retain long-range dependencies and are often used for sequence-to-sequence tasks such as translation[\[4\]](#).

On the other hand, transformers have gained in popularity in recent years thanks to their parallelization capabilities and the introduction of the attention mechanism, which enables them to efficiently process large, complex sequences more quickly without getting bogged down in sequential data processing [\[4\]](#).

1.2.1 Transformer architecture

Transformers are an architecture of neural networks proposed in 2017 by Google researchers in collaboration with the University of Toronto, in a research paper that has become legendary: "Attention is All You Need" [\[5\]](#).

Transformer architecture is based on an encoder-decoder structure.

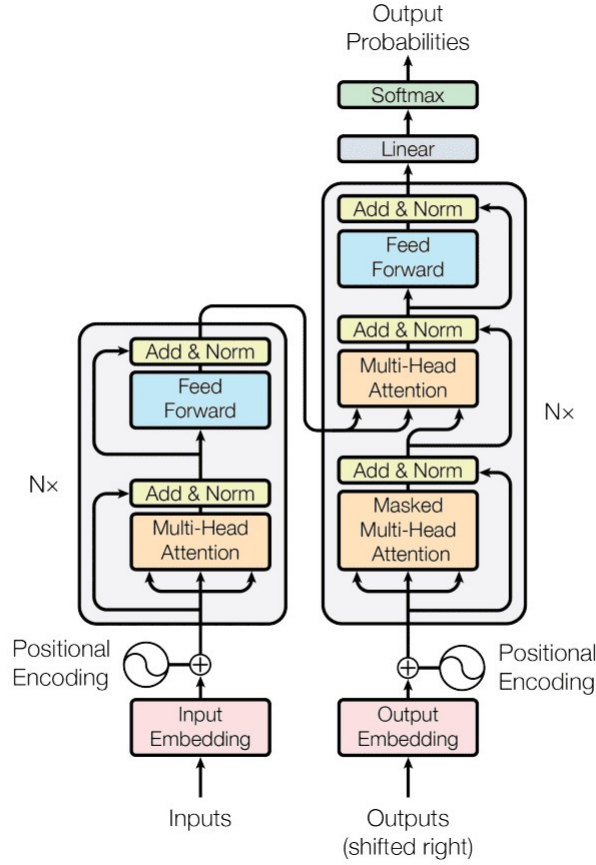


Figure 1.2: Architecture of a Transformer model [5].

- **Encoder:** Comprising a stack of N identical layers, each layer is composed of two sub-layers. The first sub-layer employs multi-head attention, while the second utilizes a fully connected simple network known as the Feed Forward Network (FFN). A residual connection surrounds each of the two sub-layers, followed by layer normalization. In essence, the output of each sub-layer can be expressed as follows: $\text{LayerNorm}(x + \text{Sublayer}(x))$, where $\text{Sublayer}(x)$ represents the function implemented by the sub-layer itself[5].
- **Decoder:** it is also composed of a stack of $N = 6$ identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, we employ residual connections around each of the sub-layers, followed by layer normalization [5].

1.2.2 Attention

In transformers, the "attention mechanism" plays a crucial role in identifying the key words within a sequence of text. This mechanism assigns weights to each word, reflecting their relative significance in predicting the next word or completing a given task. By focusing on the high-weighted words, the transformer gains a deeper understanding of the context and generates more accurate outputs[3]. It is calculated using this formula:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V} \text{ [5] ; where}$$

- Q, K and V are the query, key and value matrices respectively.
- d_k : represents the dimension of the K .

Multi-headed attention allows the model to learn several independent decision factors, instead of averaging everything into one attention. In practice, attention is calculated several times:

$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O$ [5] ; Où

- Q, K and V are the query, key and value matrices respectively.
- $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$
- W^O is the projection weight matrix applied to the concatenation of attention head outputs to obtain the final output.

The transformer uses multi-head attention in three (03) different ways [5]:

- **Self-attention in the encoder** : In the context of an encoder, self-attention allows each word in the input sequence to attend to other words in the same sequence, learning about their relationships and interdependencies. This mechanism allows the encoder to understand the context in which each word appears and to gain a more nuanced understanding of the overall meaning of the input.

Models based solely on self-attention in the encoder:

Notable examples include BERT (Bidirectional Encoder Representations from Transformers) and RoBERTa (Robustly optimised BERT approach)[6].

- **Self-attention in the decoder** : In the context of a decoder, self-attention plays a similar role to that of the encoder, allowing the decoder to capture contextual information in the output sequence. However, there is one crucial difference in the way self-attention is applied in the decoder compared to the encoder: **masking**.

Masking is a technique used in the decoder to prevent the model from "cheating" by looking at future positions in the output sequence. This technique is essential to ensure that the decoder generates the output sequence sequentially, without relying on future information. To implement masking, the decoder's self-attention mechanism is modified by setting the attention scores for future positions to a very negative value, thereby suppressing their contribution to the context vector. This ensures that the decoder is only interested in previously generated words, forcing it to rely on its current understanding of the context to predict the next word.

Models based solely on self-attention in the decoder :

The most notable models are those in the GPT (Generative Pre-trained Transformer) family developed by OpenAI, which have achieved remarkable results on a variety of text generation tasks. The most recent is GPT4. It is capable of generating human-quality text, translating languages, writing various types of creative content and answering questions in an informative manner.

- **The encoder-decoder attention** : Attention between encoder and decoder is a mechanism used in neural network architectures such as the Transformer to improve communication between encoder and decoder components. It allows the decoder to selectively focus on specific parts of the encoded input sequence when generating the output sequence, improving the model's ability to capture contextual information and produce accurate translations or summaries.

Models based on encoder-decoder attention :

The T5 (Text-To-Text Transfer Transformer) model [7] is based on the Transformer's encoder-decoder architecture. Designed by Google AI, its distinctive feature is that it formulates all NLP tasks in the form of a text-to-text task. This means that whatever the nature of the task (translation, summary, question-answer, etc.), it is reformulated into a sequence of input text and output text, offering a simple and coherent approach to solving various tasks.

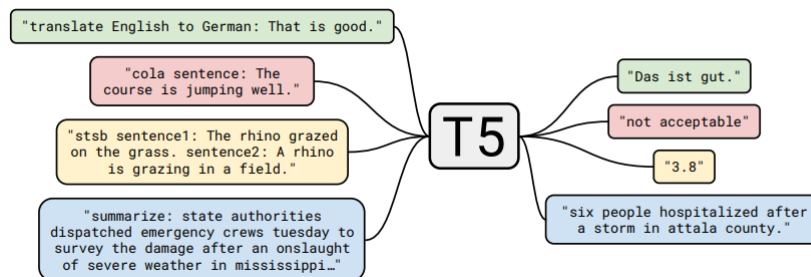


Figure 1.3: Illustration of how a T5 model works [7].

The BART model (Bidirectional and Auto-Regressive Transformers) [8] is another system based on the Transformer's encoder-decoder architecture. Developed by Facebook AI, BART combines bidirectional and auto-regressive components to improve text generation and language comprehension.

1.2.3 The Embeddings

Natural Language Processing (NLP) aims to enable machines to understand, interpret and generate text in a similar way to humans. The major challenges in NLP include accurately representing words and modelling the semantic relationships between them. In natural language processing, vectors are often used to represent words and sequences of words. A vector can be represented as an ordered list of real numbers or binary values.

Word embedding is a technique used in Natural Language Processing (NLP) to represent words as numerical vectors in a continuous vector space. It is a method of capturing the semantic and syntactic relationships between words, by assigning dense vector representations to each word in the vocabulary.

In word embedding, each word is represented by a vector of real numbers of fixed dimension. These vectors are learnt from large amounts of unlabelled text using machine learning techniques such as neural networks, using models such as Word2Vec, GloVe or FastText

(appendix 3.3). The fundamental idea behind word embedding is that words with similar contexts tend to have similar representations in vector space [9]. Consequently, words that appear frequently in similar contexts will have similar vectors. For example, the words "cat" and "dog" will have similar vector representations because they are often used in similar contexts related to pets.

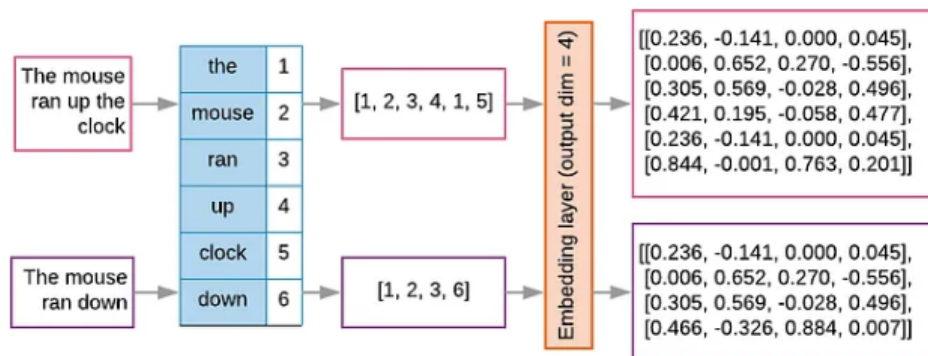


Figure 1.4: Representation of a word embedding of dimension 4 [10].

1.3 Chatbots

Chatbots are a form of artificial intelligence that allows users to interact with computer systems through a conversational interface. In the scientific literature, Chatbots are more formally referred to as conversational agents. They are practical options when it comes to consumer services, offering assistance and automated responses to a range of common questions or problems.

1.3.1 History of Chatbots

Chatbots, or conversational agents, have come a long way from their humble beginnings in the computer labs of the 1960s to the sophisticated language models such as ChatGPT[11] that are now redefining the way we interact with machines. This remarkable evolution reflects decades of research and development aimed at creating programmes capable of understanding and generating responses in natural language.

ELIZA :

ELIZA [12] was the very first chatbot mentioned above. It was created by Joseph Weizenbaum in 1966 and uses a matching and substitution methodology to simulate a conversation. The program was designed to mimic human conversation. Chatbot ELIZA works by transmitting words typed by users to a computer, then associating them with a list of possible scripted responses. It uses a script that simulates a psychotherapist. The script has had a significant impact on natural language processing and non-natural intelligence, and copies and variants have sprung up in academies across the country.

```

Welcome to
          EEEEE LL   IIII ZZZZZZ AAAAA
          EE    LL   II   ZZ   AA  AA
          EEEEE LL   II   ZZ   AAAAAA
          EE    LL   II   ZZ   AA  AA
          EEEEE LLLLL IIII ZZZZZZ AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:

```

Figure 1.5: Conversation with ELIZA [10].

PARRY :

In 1972, the American psychiatrist Kenneth Colby created PARRY [13], a program designed to mimic a patient experiencing schizophrenia, aiming to replicate the symptoms of the illness. Operating as a natural language program, PARRY mirrors the cognitive processes of an individual, utilizing a sophisticated system of assumptions, attributions, and 'emotional responses.' The program's functionality involves adjusting the weights assigned to verbal inputs to generate varied outputs.

Jabberwacky :

Developed by Rollo Carpenter in 1988[14], the chatbot was designed to replicate human conversation in an enjoyable manner. Jabberwacky has sparked additional technological advancements and is utilized for academic research, facilitated by its web presence since its inception. The chatbot employs an artificial intelligence technique known as "contextual pattern matching."

Dr. Sbaitso :

Dr. Sbaitso is a chatbot created by Creative Labs for MS-Dos in 1992[15]. It was one of the first attempts to integrate A.I. into a chatbot and is renowned for its entirely voice-driven chat program. The programme conversed with the user as if he were a psychologist. Most of its responses were along the lines of "Why do you feel that way?" rather than a complicated interaction.

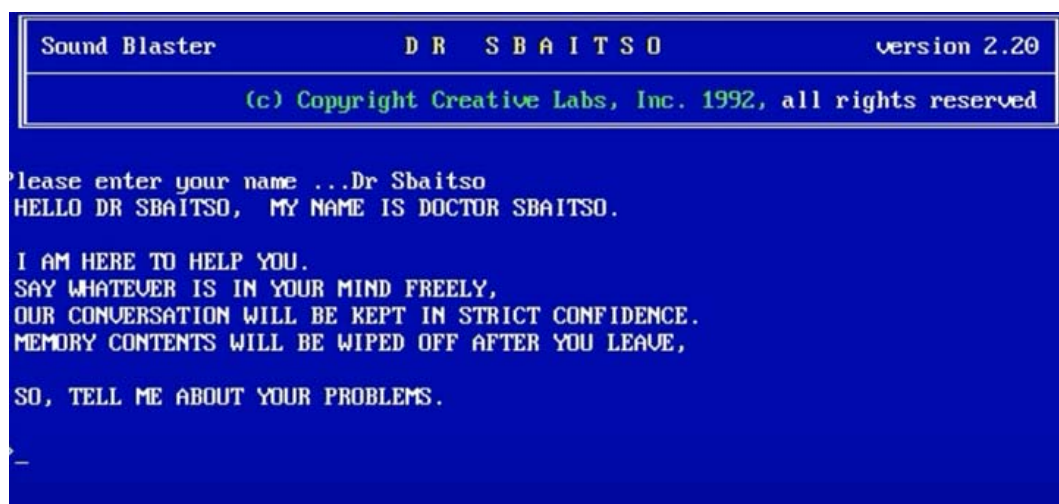


Figure 1.6: Dr. Sbaitso's interface [15].

A.L.I.C.E. :

A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) is a versatile chatbot for language processing that engages in conversations using heuristic models. In 1995, Richard Wallace pioneered the creation of ALICE, originally referred to as Alicebot due to its initial implementation on a computer named Alice[16].

The software operates based on the Artificial Intelligence Markup Language (AIML) XML schema, which defines conversation rules. In 1998, the program underwent editing in Java, and by 2001, Wallace published an AIML specification. Since then, various developers have created free and open-source versions of ALICE in multiple programming languages and diverse foreign languages.

The program emulates a conversation with a real person on the Internet, featuring Alice, a virtual character in the guise of a young woman. She discloses information about her age, hobbies, and other intriguing details while addressing user queries.



Figure 1.7: Example of a discussion with ALICE [11].

SmarterChild :

SmarterChild was a conversational agent developed by ActiveBuddy in 2001[17]. It was

mainly accessible via the AOL Instant Messenger (AIM) platform, as well as on other messaging platforms.

This chatbot offered a range of features such as the ability to answer simple questions, provide information on the weather, news, sports, etc., as well as being able to send and receive messages. SmarterChild was also able to interact with users by providing jokes, fun facts and various other information.

Siri :

Siri was created by Apple for iOS in 2010 and is an intelligent personal assistant and learning browser that uses a natural language user interface. It paved the way for all subsequent AI personal assistants and bots[11].

Google Now :

Google Now was launched at Google Inc in 2012. It answers questions, performs actions through requests to a range of web services and makes recommendations[11].

It was part of a package of updates and changes to the user interface for mobile search, which included a female-voiced mobile assistant to compete with Apple's Siri.

Google Now began as a way to get contextually relevant information based on location and time of day. It has evolved to become much more complex and elaborate, with a wide range of content categories delivered on cards.

Google Now was replaced by Google Assistant in 2017. Today, Assistant is part of a more aggressive growth strategy for Google Search. The idea is simple, Google wants to deliver information in an easy-to-read format before you even know you need it.

Cortana :

Cortana was first showcased at Microsoft's Build 2014 developer conference, and has become directly integrated into Windows phone devices and Windows 10 PCs [11].

This program uses voice recognition and relevant algorithms to obtain and respond to voice commands.

To get started, a person needs to type a question into the search box, or select the microphone and speak to Cortana. If a person isn't quite sure what to say, they'll see suggestions on the lock screen, as well as on Cortana's home page by selecting the search box in the taskbar.

Cortana can perform tasks such as reminders based on time, place or person, send emails and texts, create and manage lists, chat and play games, find facts, files, places and information, among others.

Alexa :

Alexa is an intelligent personal assistant developed by Amazon. It was introduced in 2014 and is now integrated into devices such as the Amazon Echo, Echo Dot, Echo Show and many others[11]. There is also an Alexa application and other third-party devices that integrate Alexa.

Just say "Alexa, put on some music" or "Alexa, find me an Italian restaurant" and she'll help you.

Using only the sound of your voice, you can search the web, listen to music, create to-do or shopping lists, set alarms, play podcasts, read audio books, get news or weather reports, control your smart home products and much more.

ChatGPT :

ChatGPT, developed by OpenAI in 2021 [11], is a substantial language model engineered to assist users in producing text resembling human language based on provided inputs. Its applications span various tasks, such as crafting conversations and translating languages. The model undergoes extensive training with a vast dataset, empowering it to produce text that frequently mirrors human-authored content. Appreciation has been expressed for ChatGPT's proficiency in generating text that sounds natural, as well as its potential utility across diverse fields.

1.3.2 Chatbot approaches

Chatbots can generate responses based on three (03) different approaches: the rule-based approach, the corpus-based approach and the hybrid approach[18].

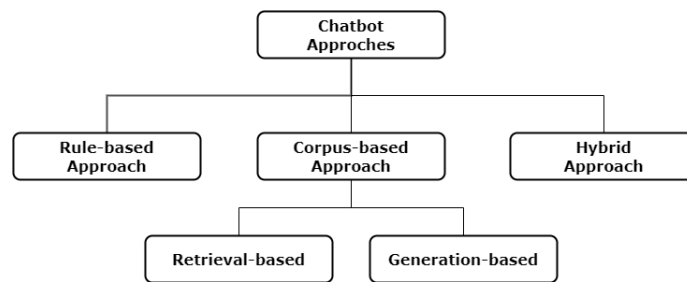


Figure 1.8: Diagram illustrating the different approaches to chatbots [18].

- **Rules-based approach** : The rules-based method involves utilizing a predetermined set of rules created by humans within the hierarchy. These rules are applied to transform user input into output by breaking down the input into a sequence of keywords to identify patterns and produce a response, rather than generating a completely new answer. While this approach may be perceived as simpler to implement, it confines responses to the boundaries of the predefined rules, potentially resulting in inaccurate answers and an overall unsatisfactory user experience[18].
- **Corpus-based approach** : The corpus-based approach uses a knowledge base containing a statistical approach to language to select appropriate responses instead of using predefined rules. Most chatbots using this approach produce their responses using two approaches:
 - Retrieval based approach: instead of generating a new answer, this method utilizes information retrieval and heuristic techniques to extract a potential response from the corpus. It considers both the input and context by pinpointing keywords to suggest the most suitable response from predefined answers.

- **Generation based approach:** the generation-based method creates responses by considering the ongoing conversation context without relying on pre-established replies. The chatbot aims to produce a fresh response, considering the user's current and past interactions. Acquiring a substantial training set for this method can be challenging. Nevertheless, there is a notable risk of response errors due to real-time generation.
- **Hybrid approach :** the hybrid method leverages the strengths of both the retrieval-based and generation-based approaches, resulting in more precise responses that capture the informative features inherent in the patterns.

1.3.3 Chatbot Techniques

The development of chatbots is based on a range of techniques designed to provide relevant responses to users. These include:

- **Parsing :** Parsing is the process of extracting pertinent information from a text by simplifying words through lexical analysis, making them more manageable for storage and manipulation. This technique is designed to establish the grammatical structure of a sentence. In its initial stages, the ELIZA chatbot employed a basic syntactic approach to identify keywords in incoming sentences. It then compared these keywords with its document corpus to generate an appropriate response for the user. A more sophisticated approach, semantic analysis, is responsible for converting the user's original sentence into a machine-understandable representation of its meaning. [19].
- **Pattern matching :** In this method, chatbots generate responses using predefined combinations of questions and answers, a complex and non-automatic process[19]. Although this speeds up response time, responses can become predictable and repetitive, leading to monotonous interactions that lack spontaneity.
- **AIML (Artificial Intelligence Mark-up Language) :** AIML is a language based on XML that helps create chatbots capable of engaging in natural conversations. It achieves this by using data structures called AIML objects. These objects, also known as AIML elements, are essentially building blocks containing two main parts: subjects and categories. Subjects are optional and have a unique "name" attribute, along with a group of related categories. Each category, considered the core unit of knowledge within an AIML file, holds at least two essential elements: a "pattern" that matches user input and a "template" that defines the corresponding response [19]. AIML grants significant flexibility and control in crafting chatbot conversation flows, but it does require some knowledge of natural language processing and programming.
- **Artificial neural network models :** Powerful chatbots are emerging thanks to recent breakthroughs in machine learning, particularly with artificial neural networks (ANNs). These ANN-powered chatbots can craft responses using either search or generation methods[19], with the latter being the current focus of research. What sets them apart from traditional rule-based chatbots is their ability to learn and adapt thanks to built-in learning algorithms.

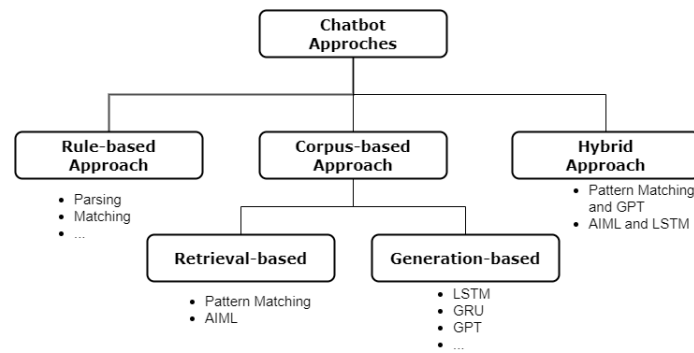


Figure 1.9: Diagram illustrating the different chatbots techniques base on the approaches[18].

1.4 Literature review about chatbot on public services in Beninese Administration

In the context of public administration in Benin, the integration of chatbots for service requests represents a significant innovation. However, this initiative is part of a wider, global evolution of chatbots around the world.

By examining previous experiences, successes and challenges encountered, this study seeks to illuminate opportunities, good practices and lessons learned to guide the effective realization of our chatbot.

1.4.1 Related works

In the field of chatbots, several initiatives around the world have demonstrated their usefulness in offering innovative solutions to improve the accessibility and quality of information for Beninese citizens. Here are a few relevant examples of related work in this field:

1.4.1.1 ChatGPT

Developed by OpenAI, ChatGPT is a sophisticated chatbot that advances the field of conversational agents. It is based on the Transformer language model, which uses deep learning to produce high-quality text. It can understand the context of a conversation, generate appropriate responses and even show some creativity in its answers, making it relevant to a variety of interaction scenarios[20].

In addition, to ensure quality interaction, OpenAI has implemented a moderation system to prevent ChatGPT from generating inappropriate or offensive responses. This system is based on reinforcement learning from human conversation data[21].

In terms of performance, ChatGPT demonstrated an impressive ability to generate accurate and contextually appropriate responses.

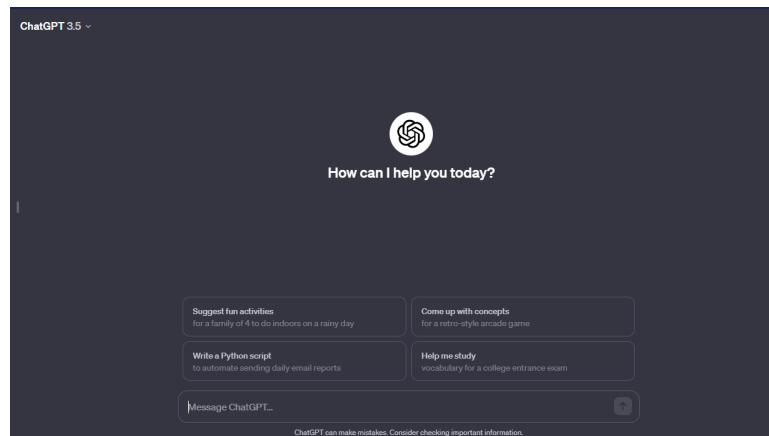


Figure 1.10: ChatGPT user interface.

1.4.1.2 Bard

Bard is an innovative chatbot, developed by Google, which has been trained on a wide range of internet data. This chatbot is able to generate creative and relevant responses using a pre-trained language model known as Transformer[5].

Bard has been designed to respond more naturally and spontaneously than most traditional chatbots. It has been programmed to understand and respond appropriately to a wide variety of topics. What sets it apart is its ability to generate detailed responses and provide accurate information.

Crucially, Bard was trained using a method called Reinforcement Learning from Human Feedback (RLHF)[21], which allows him to learn from previous interactions and continually improve his responses. This enabled Bard to develop a thorough understanding of the context of the conversation and to respond appropriately.



Figure 1.11: Bard user interface.

1.4.1.3 GPT BJ

GPT BJ is a chatbot developed by Benin's Agence des Systèmes d'Information et du Numérique (ASIN) in collaboration with the iSheero association. Its alpha version was presented to the public at the Salon de l'Entrepreneuriat Numérique et de l'Intelligence Artificielle (SENIA)

2023.

GPT BJ is designed to answer questions relating to Benin's General Tax Code, Digital Code, Labor Code and Penal Code. It can also provide general information on the Beninese government and public services[22].

1.4.2 Critique of the existing

Previous work on chatbots has made significant contributions to this emerging discipline, but there are also limitations and gaps that require in-depth analysis to guide future developments.

Table 1.1: Limits of existing chatbots on public services requests in Beninese administration

Chatbots	Limits	Justification
ChatGPT	Obsolete knowledge	ChatGPT is trained on data up to a certain date. This date used to be September 2021, but is now January 2022 for ChatGPT and April 2023 for ChatGPT Plus[23]. This means that ChatGPT cannot answer questions about events occurring after this date.
	Privacy and data security	As the number of users increases, privacy and data security become increasingly important. In fact, ChatGPT was banned in Italy at the beginning of April for privacy reasons[24]. This is all the more important as models collect a lot of personal information and preferences during interactions
	Bias and Discrimination	Due to the influence of pre-training data, the models present biases in political, ideological and other areas. The application of LLMs in public domains, such as education and advertising, should be approached with extreme caution.
Bard	Lack of in-depth understanding	Bard has limitations in his understanding of specific and complex issues. His ability to provide precise, detailed answers may be restricted when confronted with highly specific, poorly researched or constantly evolving subjects.
	Limited generative potential	Google Bard is faced with the limitation of its generative potential, unlike ChatGPT which has the ability to produce content with long text.
	Context limitation	Google Bard does not have access to the full context of a conversation or situation[25], leading to misunderstandings and incorrect responses.
GPT BJ	Limited knowledge	GPT BJ in its alpha version only deals with issues relating to Benin's General Tax Code, Digital Code, Labor Code and Penal Code.



Figure 1.12: Conversation with ChatGPT about the process of applying for a passport in Benin

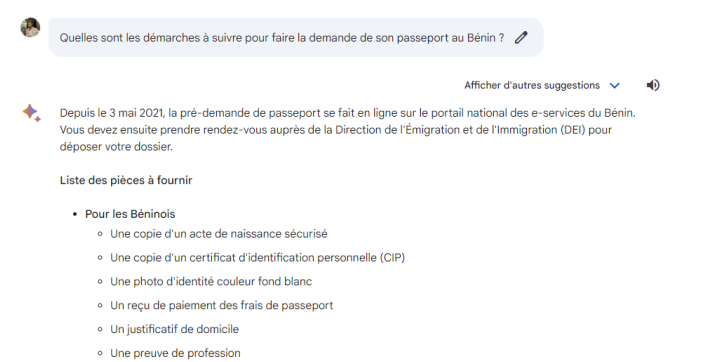


Figure 1.13: Conversation with Bard about the process of applying for a passport in Benin

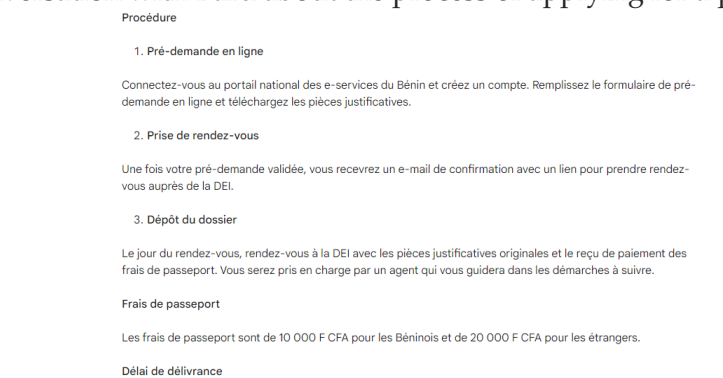


Figure 1.14: Conversation with Bard about the process of applying for a passport in Benin - 2

Conclusion

In the light of recent advances in Deep Learning, the transformations brought about by the use of Transformer models, and the emergence of Chatbots, it is clear that the application of these technologies offers promising prospects for optimising public services. In the specific context of Benin's public administration, an in-depth analysis of its structure and missions will help determine how these innovations can be judiciously integrated to meet citizens' needs in an efficient and personalised way.

Material and Methods

Introduction

In this chapter, we delve into the foundational aspects of our research methodology and the proposed solutions aimed at developing an efficient service request chatbot for the public administration domain in Benin. Our exploration encompasses a multifaceted approach, beginning with a comprehensive elucidation of the methodologies employed. We describe in detail the methods and techniques used to create our chatbot. In addition, we present a detailed overview of the technologies and execution environment of our system. This chapter seeks to provide a thorough understanding of the strategies and technical underpinnings that constitute the groundwork for the implementation of a proficient chatbot tailored for public service requests within the unique context of Benin’s administrative framework.

2.1 Methods

The creation of this conversational agent followed several key steps, from data collection to model deployment. The diagram below illustrates each crucial step of this process in a concise manner.

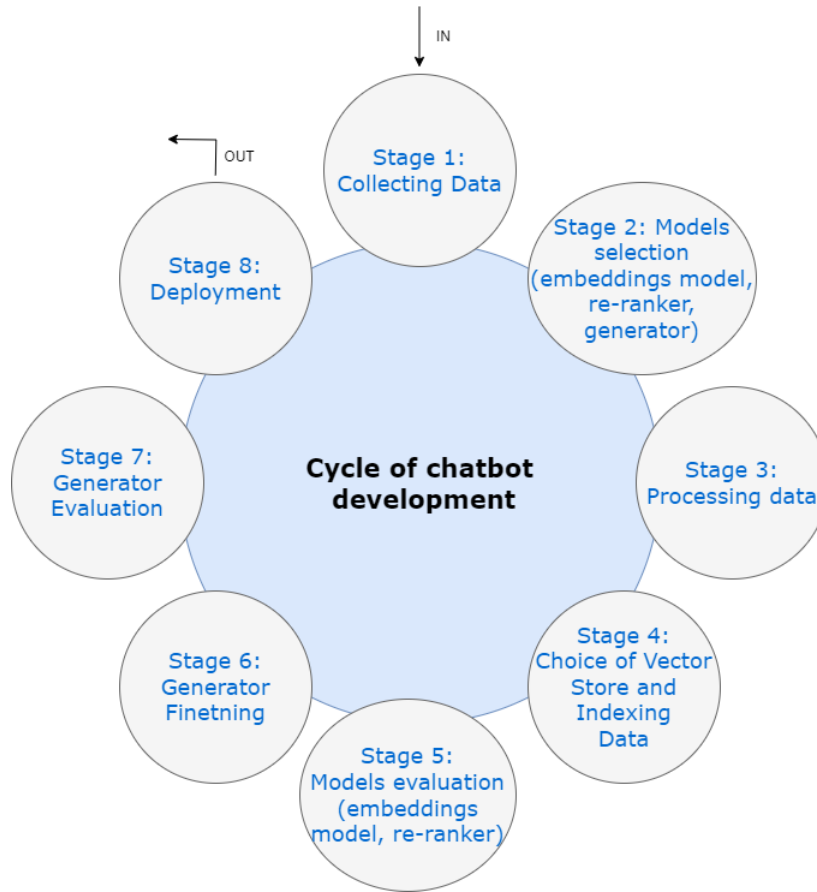


Figure 2.1: Cycle of our chatbot development

2.1.1 Functional architecture

The fundamental system architecture underlying our chatbot is RAG (Retrieval-Augmented Generation). The choice of this architecture stems from its ability to synergistically combine the advantages of information retrieval and text generation. The RAG consists of two main components: the Retriever and the Generator. The Retriever is responsible for searching and extracting relevant information from a knowledge database or text corpus. This stage identifies potential answers to users' questions. The Generator then takes these answers and transforms them into natural answers that users can understand. This architecture offers significant flexibility, as it allows the integration of various data sources, adapting the answers to the specific needs of each user and improving the overall quality of human-machine interaction. The Naive RAG follows a traditional process that includes indexing, retrieval, and generation [26] :

- **Indexing:** the indexing process is a crucial initial step that occurs offline and involves several stages. To accommodate the constraints imposed by the contextual limitations of language models, the text undergoes segmentation into smaller, more easily handled sections through a procedure referred to as chunking. Following this, an embedding model is employed to convert these chunks into vector representations, selected for its optimal combination of inference efficiency and model size. This process enables seamless similarity comparisons during the retrieval phase. Ultimately, an index is generated to store these segmented text portions and their corresponding vector em-

beddings as key-value pairs, facilitating efficient and scalable search capabilities.

- **Retrieval:** when a user query is received, the system utilizes the encoding model employed in the indexing phase to convert the input into a vector representation. Subsequently, it calculates similarity scores between the query vector and the vectorized chunks within the indexed corpus. The system then prioritizes and retrieves the top K chunks that exhibit the highest similarity to the query. These selected chunks are then utilized as an expanded contextual basis to fulfill the user's request.
- **Generation:** the posed question and chosen documents are combined to create a cohesive prompt, which is then given to a large language model for generating a response. The model's method of answering can adapt based on task-specific requirements, allowing it to utilize its inherent parametric knowledge or limit responses to the information found in the provided documents. In instances of continuous conversations, the model can incorporate existing conversational history into the prompt, facilitating effective engagement in multi-turn dialogue interactions.

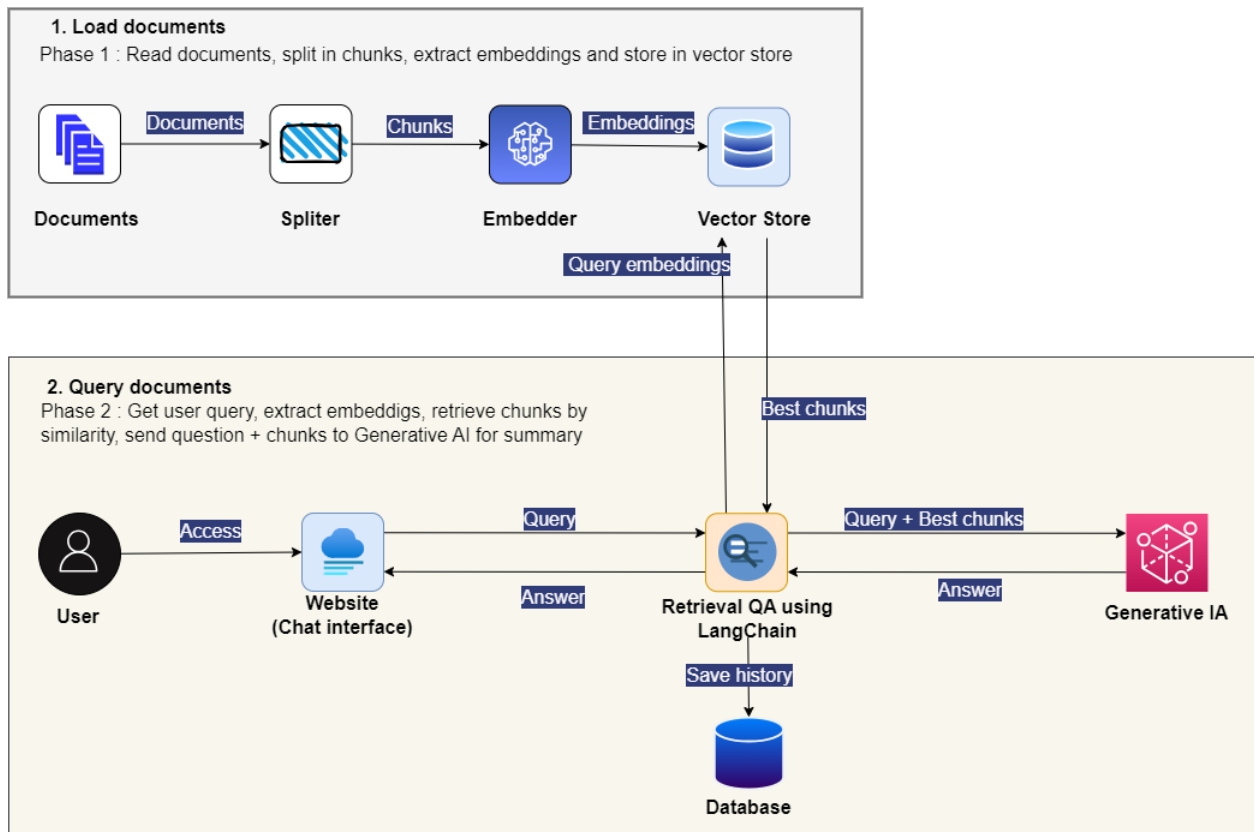


Figure 2.2: Functional architecture of the system

2.1.2 Advanced RAG

Advanced RAG was created with specific improvements to overcome the limitations of Naive RAG. In terms of the quality of retrieval, Advanced RAG implements pre-retrieval and post-retrieval strategies [26].

2.1.2.1 Pre-retrieval

The goal of optimizing data indexing is to enhance the quality of the content being indexed. This involves : enhancing data granularity, adding metadata, overlapping [26].

- **Enhancing data granularity:** improving data granularity seeks to enhance the standardization, consistency, factual accuracy, and contextual richness of text, thereby boosting the performance of the RAG system. This involves removing irrelevant information, dispelling ambiguity in entities and terms, verifying factual accuracy, preserving context, and revising outdated documents.
- **Adding metadata:** Incorporating metadata information entails integrating referenced metadata, such as dates, into chunks for filtering purposes. Additionally, it involves including metadata such as title to enhance retrieval efficiency.
- **Overlapping:** it refers to the practice of deliberately splitting input documents or queries into smaller chunks with a certain amount of redundancy between them. This means that a portion of the text is included in multiple chunks. By overlapping, the model is exposed to the same information from multiple angles, which can help it better understand the overall context and avoid repetition in the generated text.

2.1.2.2 Retrieval

We delve into various strategies for information retrieval, each catering to distinct search needs. These strategies include dense retriever, sparse retriever, and a hybrid approach combining both semantic and lexical search methodologies :

- **Dense retriever:** also known as semantic search, leverages advanced natural language processing techniques to comprehend the contextual meaning of the search queries. By embedding the semantic nuances of the content, this approach facilitates a more nuanced understanding of user intent, surpassing the limitations of traditional keyword-based searches.

The dense retriever employs pre-trained neural encoders to learn vector representations and perform retrieval by using nearest-neighbor search 3.3 in this dense embedding space [27].

- **Sparse Retriever :** In contrast, the sparse retriever, or lexical search, relies on keyword matching and conventional indexing methods to retrieve documents based on explicit word matches [28]. This approach is effective for scenarios where precise lexical matches are crucial, providing a straightforward and transparent method of retrieving information based on specific terms and phrases.

Sparse retrievers are often efficient for tasks where the semantics of the query are less critical, and the emphasis is on exact lexical correspondence. The most commonly used algorithm is Best Matching 25 (BM25) [27]. The figure below shows the performance of the BM25 algorithm and a dense retriever on entity questions and natural questions :

	DPR (NQ)	DPR (multi)	BM25 -
Natural Questions	80.1	79.4	64.5
EntityQuestions (this work)	49.7	56.7	71.2
What is the capital of [E]?	77.3	78.9	90.0
Who is [E] married to?	35.6	48.1	85.9
Where is the headquarter of [E]?	70.0	72.0	85.0
Where was [E] born?	25.4	41.8	75.2
Where was [E] educated?	26.4	41.8	73.0
Who was [E] created by?	54.1	57.7	71.7
Who is [E]'s child?	19.2	33.8	82.9
(17 more types of questions)	...	...	...

Figure 2.3: Benchmarking of Dense Passage Retriever (DPR) and BM25 on entity questions and Natural Questions (NQ) [27].

- Hybrid retriever : Hybrid retriever integrates both sparse and dense retrieval methods, representing a widely adopted strategy to address the constraints associated with either dense or sparse retrievers. In a hybrid system, two distinct candidate sets are retrieved using both dense and sparse retrievers, and subsequently, they are reevaluated and reordered based on the hybrid retrieval score [27].

2.1.2.3 Post-retrieval

Once valuable context is obtained from the database, it is crucial to integrate it into the query as input for LLMs, taking into consideration the challenges presented by limits in the context window. Submitting all pertinent documents to the LLM simultaneously might surpass the context window limit, introduce unnecessary noise, and impede the concentration on essential details. Further processing of the obtained content is required to tackle these challenges.

One approach using language models is the reranking model, which first measures the similarity of each candidate document to the query [29]. It then re-ranks the documents with the highest similarity by calculating their similarity to the query using the language model.

2.1.3 Collecting data

Data collection is a fundamental stage in the development of our chatbot for requesting services within the public administration in Benin. To obtain up-to-date and accurate information on administrative services, we opted for a web-based approach to data collection. It would have been a huge task to copy and record the information by hand. To automate this task, the web scraping technique met our needs perfectly.

Web scraping

Web scraping refers to the process of extracting content and data from websites using software. Web scraping saves time, produces a database that is readable and easy to use, and reduces the risk of human error. When extracting large volumes of data that are regularly updated, web scraping is particularly useful for accessing up-to-date data in real time, and for studying how it is changing.

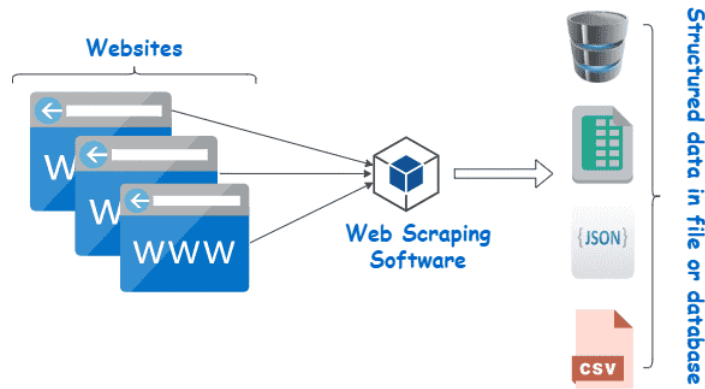


Figure 2.4: illustration of web scraping [30].

2.1.4 Model selection

2.1.4.1 Selection criteria for embedding models

The selection of an appropriate embedding model is of crucial importance in the chatbot development process, as it determines the quality of the semantic representations used to analyse and respond to user queries. In order to make an informed decision about the choice of embedding model, we used the Massive Text Embedding Benchmark (MTEB), a powerful resource made available by the research community via the Hugging Face platform.

We have focused on several characteristics of models, namely:

- **Model Size:** This refers to the amount of memory required to store the model. Larger models tend to offer better performance, but they also demand more memory and computational resources for usage.
- **Supported Languages:** This feature indicates the languages for which the model has been pre-trained and can perform natural language processing tasks.
- **Embedding Dimension:** Embeddings are vector representations of words or tokens in a continuous vector space. The embedding dimension indicates the size of these vectors. Higher-dimensional embeddings can capture finer semantic information but also require more resources.
- **Maximum Sequence Length:** This is the maximum length of text sequences that the model can effectively process. NLP models often have a limitation on the length of input text, typically expressed in the number of tokens.

The table below summarises the characteristics of the different models we have chosen.

Table 2.1: Characteristics of embedding models [31]

Model	Size (GB)	Languages	Sequence Length	Embeddings dimensions
multilingual-e5-small	0.47	Multilingual	512	384
multilingual-e5-base	1.11	Multilingual	514	768
multilingual-e5-large	2.24	Multilingual	514	1024
text-embedding-ada-002	-	Multilingual	8191	1536

2.1.4.2 Selection criteria for Re-rankers

The choice of re-rankers represents a crucial stage in our information retrieval methodology for the administrative service request chatbot in Benin. For this phase, we opted to use two re-ranking models that are widely recognised in the machine learning and information retrieval community: 'bert-multilingual-passage-reranking-msmarco' and 'mmarco-mMiniLMv2-L12-H384-v1'. These models were selected on the basis of their proven ability to improve the relevance of search results, as well as their adaptability to our multilingual linguistic context.

Table 2.2: Characteristics of re-ranking models

Model	Model size(GB)	Languages	Sequence Length	Embeddings dimensions
bert-multilingual-passage-reranking-msmarco	0.67	Multilingual	512	768
mmarco-mMiniLMv2-L12-H384-v1	0.47	Multilingual	512	384

2.1.4.3 Selection criteria for generative models

During our assessment for selecting the response generation model, we considered several crucial characteristics to evaluate the relevance of each model. These features allowed us to comprehend how each model positions itself in terms of capacity, compatibility, and performance.

- **Model Size:** The size of the model is indicated by the number of parameters it contains. The number of parameters in a model is a significant indicator of its complexity and its ability to capture information. Larger models tend to generalize better from training data, but they may be heavier to deploy.
- **Supported Languages:** Language support is a vital criterion, as our chatbot aims to provide responses in French for the Beninese context. We checked the availability of models capable of understanding and generating responses in French.
- **Sequence Length:** This refers to the maximum length of texts that each model can process. In our case, we are using 10 documents, each with a maximum size of 400 tokens. Therefore, we will need a model that can handle more than 4000 tokens as input.
- **Performance:** This pertains to the results obtained during the evaluation of models on certain popular benchmarks.

Table 2.3: Characteristics of LLMs [32]

Model	Size	Languages	Sequence Length	TruthfulQA	MMLU
Llama-2-13b-chat-hf	13B	Multilingual	4096	44.12	54.64
Mistral-7B-Instruct-v0.1	7B	Multilingual	8192	56.28	55.38
Falcon-7b-instruct	7B	Multilingual	2048	44.08	25.84

2.1.5 Data processing

Having successfully collected the data needed to feed our chatbot, the next crucial step is to prepare this data. This preparation phase ensures that our model learns effectively to generate relevant and consistent responses to user queries. Preparing this data involved several stages: data cleansing, topics modelling and splitting documents into chunks.

Cleaning data

Data cleansing is a fundamental step in preparing the information collected for use in the design and development of a language model. With this in mind, advanced techniques such as regular expressions have been used to extract, clean and restructure documents from the web such as anarchic line breaks and punctuation.

Topic modelling

Once the collected data had been successfully pre-processed, the next crucial step in our methodology was to analyse the themes inherent in the documents using the Latent Dirichlet Allocation (LDA) model. The main objective of this step was to identify the dominant topics present in the texts relating to public administration services in Benin, in order to establish a conceptual and organisational structure for the service request chatbot. This approach considerably strengthened the chatbot's ability to interact in a contextual and relevant way with citizens seeking administrative services, by providing precise responses tailored to the underlying themes of the requests.

Chunking data

When handling external documents, the initial phase revolves around dissecting them into smaller segments to extract nuanced features, which are subsequently embedded to capture their semantic nuances. However, embedding text chunks that are excessively large or too small may result in suboptimal outcomes. Hence, determining the optimal chunk size for documents within the corpus is pivotal for ensuring the accuracy and relevance of the retrieved results [26].

Selecting an appropriate chunking strategy demands thoughtful consideration of several key factors, including the nature of the indexed content, the embedding model and its ideal block size, the anticipated length and complexity of user queries, and the specific application's utilization of the retrieved results. For example, the choice of a chunking model should be guided by the content's length—whether it is lengthier or shorter. The segmentation of documents into segments was executed by taking into account natural pauses in the text, such as paragraphs or complete sentences, with the goal of minimizing any potential loss of meaning or information.

2.1.6 Choice of Vector Store and Indexing data

Finding efficient ways to store, retrieve and analyse data has become essential in today's data-driven world, where information is generated in extraordinary ways. Vector stores are designed to handle large-scale data and work well in programmes such as recommender systems, image recognition and machine learning.

A vector store is a specialised database created to store and manage large-scale data vectors

(embeddings) efficiently. It enables fast and accurate similarity searching, matching and analysis of these embeddings. In traditional databases, we generally query the rows of the database where the value generally corresponds exactly to our query. In vector databases, we apply a similarity metric to find a vector that most closely resembles our query.

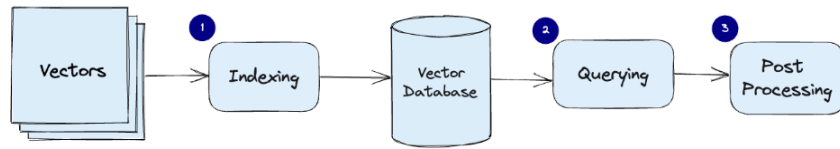


Figure 2.5: Steps for using a vector store [33].

2.1.7 Fine-tuning : Parameter-Efficient Fine Tuning (PEFT)

Fine-tuning is a process whereby a pre-trained model is adjusted to fit a specific task. The weights of the pre-trained model are adjusted using a set of annotated data for that task. The aim is to specialise the pre-trained model to better understand the specifics of the task and improve its performance.

The problem with traditional fine-tuning is that it can require a large amount of labelled data to avoid overfitting and achieve good performance on the target task. When using massive pre-trained language models with hundreds of millions or billions of parameters, this data requirement can become a major obstacle, especially when data and compute resources are limited. This is where Parameter-efficient Fine-tuning (PEFT) comes in.

PEFT is a technique used in Natural Language Processing (NLP) to improve the performance of pre-trained language models on specific downstream tasks. It involves reusing the parameters of the pre-trained model and refining them on a smaller dataset, saving computational resources and time compared to training the entire model from scratch [34]. There are several different PEFT techniques, but they all share the common principle of adding a small number of new parameters to the pre-trained model and then fine-tuning only those new parameters.

2.1.7.1 Adapter

Adapters represent a specialized submodule introduced into pre-trained language models for the purpose of fine-tuning, allowing modification of hidden representations without altering the frozen parameters of the rest of the model. Typically integrated after the multi-head attention and feed-forward layers in the transformer architecture, adapters enable updating only the parameters within these components during fine-tuning. This targeted approach proves efficient, enhancing the model's performance on specific tasks without the need to retrain the entire model, thereby conserving time and computational resources [34].

The integration of adapters into the transformer architecture is a straightforward process. By adding adapters to each transformer layer and incorporating a classifier layer atop the pre-trained model, one can update the adapter and classifier parameters to achieve task-specific improvements without affecting the frozen parameters elsewhere in the model. The adapter module itself consists of two feed-forward projection layers linked by a non-linear activation

layer, along with a skip connection that maintains the input unchanged, providing a flexible and efficient means of adapting pre-trained language models for specific tasks.

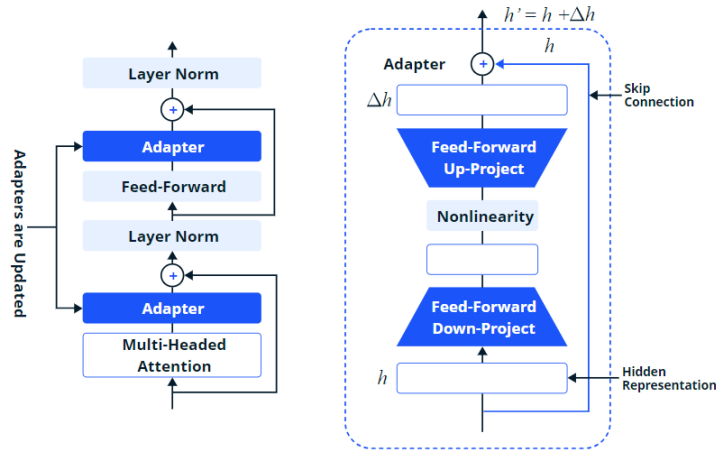


Figure 2.6: Representation of an adapter in transformer [34].

2.1.7.2 LoRA

Low-Rank Adaptation (LoRA) is an alternative approach within the realm of fine-tuning large language models for specific tasks or domains. Much like adapters, LoRA is a compact trainable submodule that can be incorporated into the transformer architecture. The key process involves immobilizing the pre-trained model weights and integrating trainable rank decomposition matrices into each layer of the transformer structure, significantly reducing the number of trainable parameters for downstream tasks. This not only facilitates more efficient task-switching but also lowers the hardware requirements, making it more accessible. Importantly, LoRA introduces no additional inference latency when compared to other methods [35].

The integration of LoRA occurs in parallel with the modules of the pre-trained transformer model, specifically alongside the feed-forward layers. In a feed-forward layer, consisting of two projection layers with a non-linear layer in between, the input vector undergoes affine transformation to produce an output vector with altered dimensionality. LoRA layers are inserted adjacent to each of the two feed-forward layers, contributing to the model's adaptability and task-specific fine-tuning.

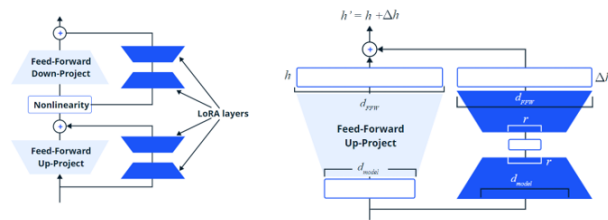


Figure 2.7: Representation of LoRA in transformer [34].

2.1.7.3 QLoRA

Quantized Low-Rank Adaptation (QLoRA) is a PEFT technique that can be used to fine-tune Large Language Models (LLMs) on limited hardware resources. QLoRA combines LoRa and quantization, a technique that reduces the precision of weights in an LLM. This enables QLoRA to achieve LoRa-like performance while using even fewer hardware resources.

2.1.7.4 Prompt tuning

Prompt tuning is a way to adapt pre-trained language models to specific tasks without requiring extensive retraining. Unlike traditional methods that adjust the entire model, prompt tuning focuses on a small, task-specific "prompt" added to the input. This prompt, learned through training, guides the model towards the desired output for that particular task.

This approach has several advantages. It outperforms other techniques like GPT-3's few-shot learning on larger models [34], improves robustness in domain transfer, and allows for efficient ensembling of different prompts. Plus, storing just the small prompt for each task is much easier than creating and storing separate models for each.

Prompt tuning works by prepending some learned vectors to the input sequence. These vectors, called "prefix embeddings," guide the model's processing and are adjusted during fine-tuning, while the rest of the pre-trained model remains unchanged. This makes it a simpler and more efficient way to adapt powerful language models to specific needs.

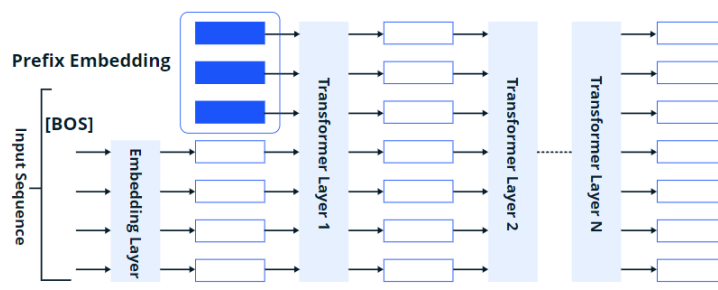


Figure 2.8: Representation of Prompt tuning in transformer [34].

2.1.8 Model evaluation metrics

2.1.8.1 Retriever evaluation metrics

The evaluation of the search focuses on the chatbot's ability to retrieve relevant documents in response to user queries. The metrics used for this evaluation are described in the table below.

Table 2.4: Retriever’s metrics

Metric	Formula	Interpretation
Precision	$\text{Precision} = \frac{\text{RestPert}}{\text{Rest}}$ [36]; where RestPert is the number of documents returned and relevant by the system, and Rest is the total number of documents returned.	Precision measures the proportion of relevant documents among the returned documents. The values are between 0 and 1. A value close to 1 indicates a better result.
Recall	$\text{Recall} = \frac{\text{RestPert}}{\text{Pert}}$ [36]; where RestPert is the number of documents returned and relevant by the system, and Pert is the total number of relevant documents available.	Recall measures the proportion of relevant documents returned among all available relevant documents. High recall indicates that the system retrieves most of the relevant documents.
F-Measure	$F1\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$ [36];	It evaluates the balance between precision and recall. High F-Measure indicates a better result.

The formulas for precision and recall were derived from a table summarizing the confusion matrix in the case of information retrieval.

Table 2.5: Confusion matrix for retrieval evaluation

	Relevant	Irrelevant
Retrieved	True Positive (TP)	False Positive (FP)
Not retrieved	False Negative (FN)	True Negative (TN)

2.1.8.2 Generator evaluation metrics

Text summary evaluation metrics are crucial to ensuring that the summaries generated are accurate, consistent and relevant. These metrics allow the quality of the language model’s work to be quantified and improved over time. Here are some of the most popular metrics used to evaluate text summaries produced by a generative model:

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) :

The assessment of summary quality often relies on the widely-used ROUGE metric, a prominent measure that gauges the excellence of summaries. It quantifies the commonality of n-grams between the text produced by the model and the reference text [37]. The ROUGE metric encompasses various versions, such as ROUGE-1, ROUGE-2, ROUGE-L. Each of these variants employs distinct approaches to compute the similarity between the two texts.

- ROUGE-1 : It calculates the proportion of words in the generated summary that are also present in the reference summary.

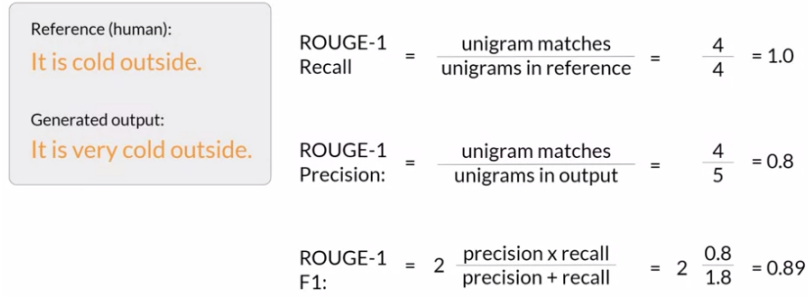


Figure 2.9: ROUGE-1 formula [38].

- ROUGE-2: It calculates the proportion of bigrams in the generated summary that are also present in the reference summary.

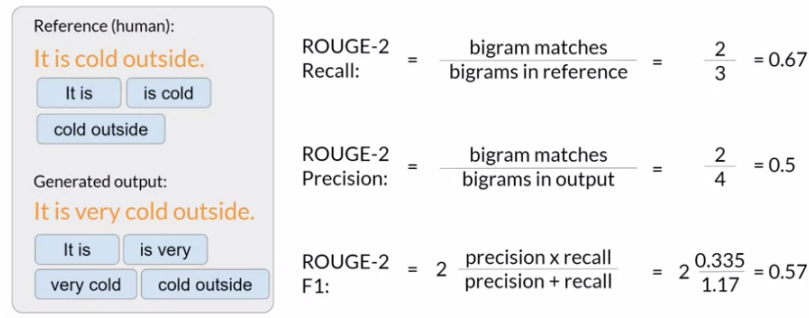


Figure 2.10: ROUGE-2 formula [38].

- ROUGE-L: It measures the similarity between the word sequence of the generated abstract and the reference abstract using the longest sequence of words in common.

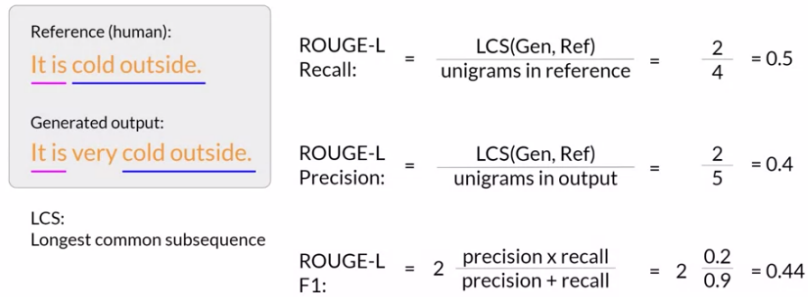


Figure 2.11: ROUGE-L formula [38].

BERTScore :

This metric relies on the BERT model and employs cosine similarity to assess the similarity between each token or n-gram in the produced summary and the reference. BERTScore excels in handling word ambiguity, synonyms, and antonyms, making it especially valuable for evaluating the excellence of summaries generated by Transformer models [39].

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^T \hat{x}_j$$

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^T \hat{x}_j$$

where x is the reference sentence and $\hat{x}$ is the candidate sentence. x_i and $\hat{x}_j$ are the embeddings of the elements of x and y respectively.

$$F_{BERT} = 2 \frac{P_{BERT} \times R_{BERT}}{P_{BERT} + R_{BERT}}$$

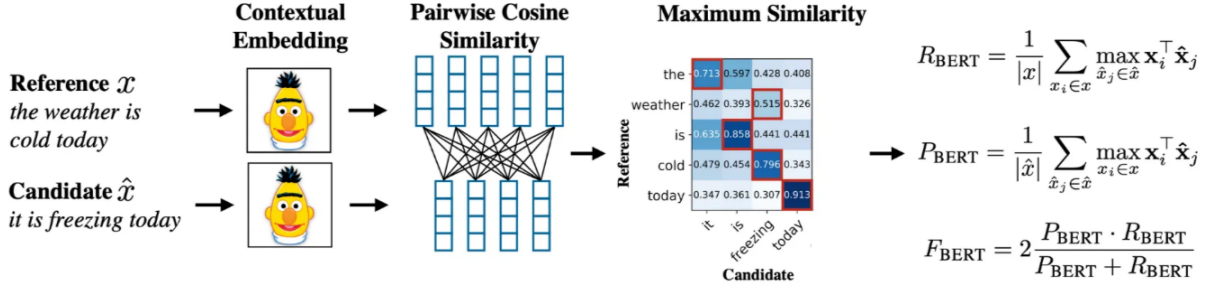


Figure 2.12: BERTScore formula [40].

2.2 Materials

For our implementation and testing, we worked :

1. online at Google Colaboratory on A100-SXM4-40GB ;
2. on a computer with the following characteristics :
 - Operating system: Windows 10 professional ;
 - Processor: Intel® Core™ i7-4700MQ CPU @ 2.40GHz
 - RAM: 16 GB ;
 - Graphics card: NVIDIA GeForce 740M 10GB+2GB.

Table 2.6: Tools description

Tools	Description	Features
Python	Python, a dynamically typed and high-level programming language, has firmly established itself as a linchpin in the realm of software development. Renowned for its readability and simplicity, Python prioritizes ease of use, making it an ideal choice for beginners and seasoned developers alike	Python 3
Google Colaboratory	commonly referred to as Google Colab, stands as a dynamic and accessible platform in the realm of cloud-based Jupyter notebooks. Leveraging the power of Google's infrastructure, Colab offers a collaborative and interactive environment for coding in Python	online
Flask	Built on the principles of simplicity and extensibility, Flask offers an elegant and efficient way to create web applications. Its minimalist design empowers developers to structure applications as needed, fostering flexibility in crafting modular APIs	3.0.2
StreamLit	Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps.	1.27.0
Selenium	Selenium is a suite of open source tools that offers a complete solution for automating web browsing tasks. It was initially developed for web application testing, but its use has quickly been extended to webscraping due to its advanced capabilities.	4.15
Langchain	LangChain is a framework for developing applications utilizing large language models, and its goal is to enable developers to conveniently utilize other data sources and interact with other applications. To enable this, LangChain provides components (modular abstractions) and chains (customizable use case-specific pipelines)	0.1.9
MySQL	MySQL stands as the cornerstone of our data management infrastructure, offering a robust and reliable relational database solution for our system by saving conversation's history.	8.0.31
Docker	Docker, a pivotal component in our system's technological stack, revolutionizes the deployment and scalability of our service request chatbot. Functioning as a containerization platform, Docker encapsulates our application, its dependencies, and runtime into a standardized unit	23.0.3
VS Code	Is a lightweight yet powerful integrated development environment (IDE), VS Code serves as our coding hub, providing an intuitive platform for crafting, testing, and debugging code. Its extensive library of extensions caters to diverse programming languages and frameworks, ensuring compatibility with our chosen technologies.	1.79
Github Actions	Leveraging GitHub Actions, our CI/CD pipeline automated the process of code integration, testing, and deployment, ensuring a consistent and reliable workflow. With each code push or pull request, automated tests were triggered, swiftly identifying potential issues and enabling the team to address them proactively. The CI/CD pipeline seamlessly orchestrated the deployment of our chatbot across different environments, facilitating rapid iterations and reducing the risk of deployment-related errors.	online

Conclusion

In summary, this chapter has delineated the methodologies employed for data collection, processing, and the development of the chatbot system. The chosen technologies, programming languages, and the logical architecture constitute the foundation for our subsequent analysis and evaluation. The forthcoming sections will delve into the results and insights derived from the application of these methods.

Results and discussions

Introduction

This chapter unveils the outcomes of our research endeavors, presenting a comprehensive analysis of the chatbot system’s performance. The discussion will delve into the evaluation of datasets, benchmarking of various models, and the system’s overall effectiveness. Through the lens of results, we aim to unravel insights into the strengths and limitations of our approach, fostering a deeper understanding of the implications for the integration of chatbots in public administration services. The ensuing dialogue will critically examine these findings, providing a nuanced perspective on the implications for future advancements in this domain.

3.1 Datasets

3.1.1 Dataset after scrapping and processing

After scraping the Benin public services site¹ using Selenium, we collected data on 32 administrations 3.3.

The table below gives some figures for our corpus of text obtained after collecting data:

Table 3.1: corpus after collecting data

Corpus	Duration	Language	Number of documents	Number of tokens
Administrations	3 H	French	1137	409628

Figure 3.1 showing the size distribution (number of tokens) of the documents reveals that the majority of documents have a size in the range [0, 1000] and more precisely in the range [0, 500]. This means that the majority of services are described in a short manner. However, this range remains high when we assume that embedding models have an average context size of 512 tokens. It is therefore wise to keep the size of our documents below 500 to limit the loss of information.

¹<https://service-public.bj/public/services/e-services>

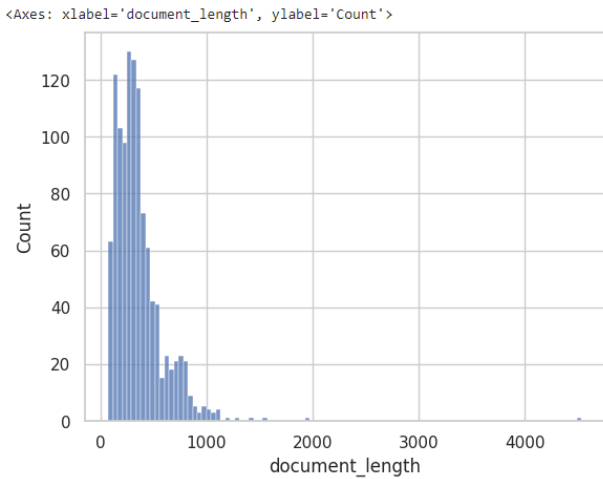


Figure 3.1: Document size distribution before chunking

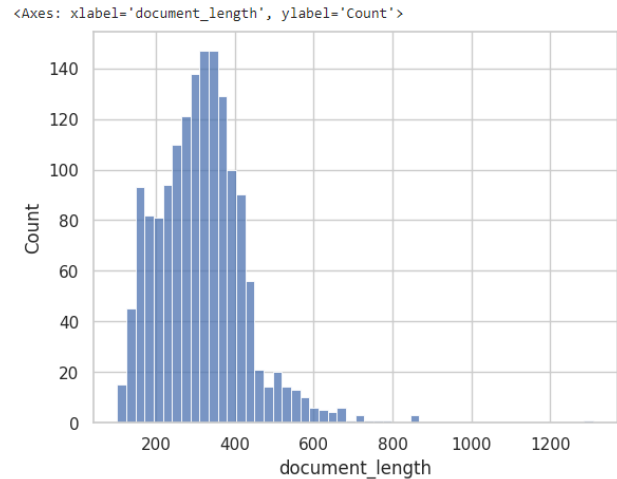


Figure 3.2: Document size distribution after chunking

Figure 3.2 shows the document size distribution after chunking. Here we can see a considerable reduction in the number of documents in the [500, 1000] range.

After processing the documents, we obtained a final corpus comprising a total of 1572 distinct documents, which formed the basis of our in-depth study for the creation of our chatbot. The figure below shows the various items of information making up our dataset.

	context	links	category	topics
0	Catégorie: Agence Nationale d'Identification d...	https://service-public.bj/public/services/serv...	Agence Nationale d'Identification des Personnes	service, identification, personnes, copie, act...
1	Catégorie: Agence Nationale d'Identification d...	https://service-public.bj/public/services/serv...	Agence Nationale d'Identification des Personnes	service, identification, personnes, copie, act...
2	Catégorie: Agence Nationale d'Identification d...	https://service-public.bj/public/services/serv...	Agence Nationale d'Identification des Personnes	service, identification, personnes, copie, act...
3	Catégorie: Agence Nationale d'Identification d...	https://service-public.bj/public/services/serv...	Agence Nationale d'Identification des Personnes	service, identification, personnes, copie, act...
4	Catégorie: Agence Nationale d'Identification d...	https://service-public.bj/public/services/serv...	Agence Nationale d'Identification des Personnes	service, identification, personnes, copie, act...
...	...	...	...	...
1567	Catégorie: Ministère du Travail et de la Fonct...	https://service-public.bj/public/services/serv...	Ministère du Travail et de la Fonction Publique	service, travail, institution, demande, foncti...
1568	Catégorie: Ministère du Travail et de la Fonct...	https://service-public.bj/public/services/serv...	Ministère du Travail et de la Fonction Publique	service, travail, institution, demande, foncti...
1569	Catégorie: Présidence de la République du Béné...	https://service-public.bj/public/services/serv...	Présidence de la République du Bénin	tel, antenne, départementale, quartier, emploi...
1570	Catégorie: Présidence de la République du Béné...	https://service-public.bj/public/services/serv...	Présidence de la République du Bénin	tel, antenne, départementale, quartier, emploi...
1571	Catégorie: Présidence de la République du Béné...	https://service-public.bj/public/services/serv...	Présidence de la République du Bénin	tel, antenne, départementale, quartier, emploi...

1572 rows x 5 columns

Figure 3.3: Details of our dataset after processing

3.1.2 Dataset for Retrievers evaluation

To assess our retriever, we opted for the Pinecone vector database due to its user-friendly interface and its capacity to manage extensive vectors.

3.1.2.1 Semantic search dataset

To assess the efficiency of our dense approach, it is imperative to have a comprehensive and varied test dataset. Instead of relying on manually annotated data, we opted for a generative method to construct our dataset. Utilizing an open-source generative model based on a given prompt, we generated specific questions for each document within our corpus. We used 513 administrative documents related to public services in Benin. For each document, our generative model incorporated a prompt tailored to the document's nature, enabling the generation of three distinct questions per document. This method resulted in a substantial dataset, comprising 513×3 , totaling 1,539 questions, each linked to its respective document. An illustration of the prompt employed in this process is detailed here:

```
user_input = ("En te servant du contexte fourni, pose 3 questions pertinentes. Les mots des questions doivent être différents des mots du contexte fourni.")
```

```
best_context = "Délivrance d'acte de naissance sécurisé. Acte de naissance PRAN/PEDEC Acte de naissance copie intégrale Acte de naissance par transcription. Durée estimée: 48h. Coût de service: 1000 FCFA. Site Internet: https://eservices.anip.bj. Qui peut faire la demande? Tout citoyen de nationalité béninoise. Institution en charge du service: Agence Nationale d'Identification des Personnes"
```

```
messages = [ "role": "system", "content": "Tu es un générateur de question pour des questions qui sont liées au service public béninois pour permettre à la population béninoise de comprendre les différentes étapes pour demander un service dans l'administration." ]
```

```
prompt = "contexte : " + best_context + ". " + user_input
```

```
Result : "Quel est le délai requis pour obtenir un acte de naissance sécurisé au Bénin ?Combien coûte la procédure de demande d'un acte de naissance sécurisé en FCFA ?Quel est le lien en ligne à utiliser pour appliquer à une copie d'acte de naissance dans l'administration béninoise ?"
```

3.1.2.2 Hybrid search dataset

First off all, we generated an assessment dataset comprising 1,539 questions (513 sets of three questions each) with the explicit aim of emphasizing the capacity to extract information when confronted with acronyms and codes, such as orders or decrees. Every question was meticulously crafted, incorporating bi-grams and tri-grams that involve acronyms or codes. This methodology was chosen to replicate authentic situations where users might inquire about particular administrative entities or procedures by referencing specific codes.

```

0          PRAN/PEDEC ?
1      ANIP localisation ?
2          ANIP définition?
3          FID définition?
4          CIP définition?
5          CIP démarches?
6      Démarches RAVIP?
7      Enrolement RAVIP?
8          Coût IFU?
9      Statut-APIEx c'est quoi?

```

Figure 3.4: Sample of acronyms questions

Then, the assessment dataset was meticulously created to mirror various real-world situations that individuals in Benin may face while engaging with public authorities. To achieve this, we randomly picked three questions from each document, encompassing both natural queries and those related to administrative acronyms. This dataset, comprising a total of 1539 questions (513×3), will be utilized as a foundation for assessing the efficiency of our hybrid approach in retrieving pertinent information and addressing intricate inquiries.

	links	context	questions
0	https://service-public.bj/public/services/serv...	passage: Délivrance d'acte de naissance sécuri...	PRAN/PEDEC ?PRAN/PEDEC ?Quel est le délai requ...
1	https://service-public.bj/public/services/serv...	passage: Délivrance d'acte de naissance sécuri...	Si les parents d'une personne née à l'étranger...
2	https://service-public.bj/public/services/serv...	passage: Délivrance de certificat de coutumes ...	Quelle est l'autorité responsable de fournir l...
3	https://service-public.bj/public/services/serv...	passage: Délivrance de l'attestation d'Identit...	FID définition ?FID définition ?Qu'est-ce qui ...
4	https://service-public.bj/public/services/serv...	passage: Délivrance de la carte d'identité nat...	À quelle adresse, et pendant quelles heures pe...
...	...	...	...
508	https://service-public.bj/public/services/serv...	passage: Arrêté d'indemnité de spécialisation ...	Comment sont calculés les pourcentages de l'in...
509	https://service-public.bj/public/services/serv...	passage: Arrêté d'indemnité de spécialisation ...	Quelle est la catégorie de professionnels auto...
510	https://service-public.bj/public/services/serv...	passage: Programme Spécial d'Insertion dans l'...	Emploi/PSIE comment ?Emploi/PSIE comment ?Quel...
511	https://service-public.bj/public/services/serv...	passage: Programme Spécial d'Insertion dans l'...	Emploi/PSIE localisation ?Emploi/PSIE localisa...
512	https://service-public.bj/public/services/serv...	passage: Programme Spécial d'Insertion dans l'...	Quelles sont les coordonnées de contact pour l...

513 rows × 3 columns

Figure 3.5: Mixed dataset for hybrid search evaluation

3.1.3 Dataset for Generators evaluation

Selecting the right dataset is a crucial step in the process of evaluating response generation models for our generative model selection. In this assessment, we have chosen to utilize the "SQuADfr" dataset. This dataset is the French adaptation of the well-known Stanford Question Answering Dataset (SQuAD), known for its high quality and diverse range of questions and answers. The relevance of the "SQuADfr" dataset to our project lies in its alignment with our chatbot's target audience—Beninese users who predominantly communicate in French

during their administrative interactions. Comprising contexts, questions, and answers extracted from these contexts, the "SQuADfr" dataset serves as a valuable resource for our evaluation.

```
{'id': '57300911947a6a140053cfb9',
 'title': 'Rhine',
 'context': 'Les tribus germaniques traversèrent le Rhin durant la période des migrations ,
 au , établissant les royaumes de Francia sur le Rhin inférieur , la Bourgogne sur le Haut-
 Rhin et l\' Alemannie sur le Haut-Rhin . Cet âge héroïque germanique se reflète dans la
 légende médiévale , comme le " Nibelungenlied " qui raconte que le héros Siegfried tua un
 dragon sur les Drachenfels ( Siebengebirge ) , près de Bonn au Rhin et des Bourguignons et
 leur cour à Worms , au trésor de Rhin et de Kriemhild , qui fut jeté dans le Rhin doré .',
 'question': 'Qui est le héros qui a tué un dragon sur les Drachenfels ?',
 'answers': {'text': ['Siegfried', 'Siegfried', 'Siegfried'],
 'answer_start': [329, 329, 329]}}
```

Figure 3.6: Sample of SQuADfr dataset

For the evaluation we used the validation data, giving a total of 10566 elements.

3.1.4 Training and Test Set

To establish a training and a test set, we employed an open-source generative model to generate responses to mixed questions. We subsequently categorized these responses into in-context and out-context sets, which act as evaluation benchmarks for our models. The training/test set, therefore, consists of questions, contexts, and the corresponding generated answers.

	question	context	answer
0	Quels sont les documents nécessaires pour l'en...	Catégorie: Agence pour la Gestion de la Logist...	Les documents nécessaires sont:\n- Un titre de...
1	Quels sont les documents nécessaires que je do...	Catégorie: Autorité de Régulation des Communic...	Pour obtenir une licence de réseau RMU ou 3RP,...
2	Quels documents sont nécessaires pour faire un...	Catégorie: Autorité de Régulation des Communic...	Les documents nécessaires pour faire une deman...
3	Comment la pratique régulière d'un sport peut ...	Catégorie: Agence Nationale d'Identification d...	Désolé, je ne suis pas capable de trouver répo...
4	Quelle est la procédure pour obtenir une autor...	Catégorie: Agence pour la Gestion de la Logist...	Désolé, je ne trouve pas de réponse exacte.

Figure 3.7: The components of our training set

Table 3.2: The statistics for our training and test sets

	Mixed questions		
	in-context	out-context	Total
Training set	3500	500	4000
Test set	900	100	1000

3.2 Benchmarking

3.2.1 Benchmarking of retrievers

3.2.1.1 Evaluation of semantic approach

To evaluate the semantic approach of our retriever, we used our dataset of natural questions generated from our corpus and successively evaluated the different processes carried out on our dataset. We calculated the recall on the top 10 documents retrieved by our vector store for each of the embedding models. The table below shows the results of our evaluation.

Table 3.3: Embeddings models evaluation on naturals questions

	Recall on Top 10(%)		
	without processing	with metadata	with metadata + overlapping
multilingual-e5-small	77.70	82.94	83.38
multilingual-e5-base	80.85	85.84	86.86
multilingual-e5-large	85.34	90.01	91.47
text-embedding-ada-002	87.04	92.21	92.54

Recall notably increased with metadata addition and even more with overlapping. Using "multilingual-e5-small" as an example, recall climbed from 77.70% to 83.38%. These findings clearly illustrate the significant gains in document search performance achieved through semantic approaches involving overlapping techniques and metadata-enriched documents.

The multilingual-e5-large embedding model stands out within the multilingual-e5 family for its high recall rate, exceeding the performance of e5-small and e5-base. Notably, its performance is comparable to OpenAI's "text-embedding-ada-002" which incurs a cost of \$0.0001/1000 tokens.

Table 3.4: Cost of model use

	Recall on Top 10 (%)	Prices	Cost for 409,628 tokens
multilingual-e5-large	91.47	free	\$0
text-embedding-ada-002	92.54	\$0.0001/1000 tokens	\$0.04

Utilizing the "text-embedding-ada-002" embedding model for our 409,628-token corpus comes at a cost of \$0.04, while opting for the e5-large model incurs a fee of \$0. Given these considerations, we have opted for the "multilingual-e5-large" model due to its cost-effectiveness, complimentary availability, and comparable performance to "text-embedding-ada-002". This decision allows us to sustain a budget-friendly operational cost for our chat-bot system.

3.2.1.2 Evaluation of Hybrid approach

To compare the effectiveness of our hybrid approach to the semantic approach, we measured their top 10 recall on two datasets: one with acronym questions and another with mixed natural and acronym questions. The results are summarized in the table below.

Table 3.5: Performance of the hybrid approach compared to the semantic approach of our retriever

	Recall on Top 10(%)	
	Semantic approach	Hybrid approach
Acronyms questions	82.90	86.64
Mixed questions	87.26	87.97

The hybrid approach exhibited a significant improvement in recall compared to the semantic approach for acronym-related queries. This improvement extended to the mixed dataset, suggesting the hybrid approach’s overall superiority in handling diverse query types.

3.2.2 Benchmarking of re-rankers

To analyze the impact of re-ranking on retrieval accuracy, we calculate recall@10 for the top re-ranked documents, retrieved from a pre-filtered set of top 10 documents. The table compares recall and latency across different re-ranking models.

Table 3.6: Performance of re-rankers on our mixed dataset of questions.

	Recall on Top 10 (%)
bert-multilingual-passage-reranking-msmarco	90.18
mmarco-mMiniLMv2-L12-H384-v1	89.47

Table 3.7: Comparison of the performance of our system without and with the re-ranker

Recall on Top 10(%)	
Without re-ranker	With re-ranker
87.97	90.18

Based on its superior performance compared to the mmarco-mMiniLMv2-L12-H384-v1 model in Table 3.6, the bert-multilingual-passage-reranking-msmarco model has been chosen as our system’s re-ranker. Table 3.7 further demonstrates the effectiveness of this selection, as including the re-ranker significantly improved the relevance of retrieved documents for user queries.

3.2.3 Benchmarking of Generators

To evaluate our generative models on the SQuADfr dataset, we calculated the ROUGE-1, ROUGE-2, ROUGE-L and BertScore.

Table 3.8: Evaluation of some generative models on the SQuADfr dataset

	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
Falcon-7b-instruct	0.203	0.121	0.197	0.833
Mistral-7B-Instruct-v0.1	0.399	0.266	0.393	0.871
Llama-2-13b-chat	0.418	0.264	0.410	0.864

Evaluation results showed that the Mistral-7B-Instruct-v0.1 model exhibited performance surprisingly close to the Llama-2-13b-chat-hf model, Despite being smaller. Due to its efficiency and comparable performance, the Mistral-7B-Instruct-v0.1 model was chosen as our generative model.

Our fine-tuning process involved applying QLoRA to the Mistral-7b-instruct model using the training data. The figure below visually represents the key metrics resulting from this training.

▼ train	
epoch	14.72
global_step	46
learning_rate	0.0002
loss	0.0207
total_flos	46,685,412,604,035,070
train_loss	0.3556471831043777
train_runtime	620.1305
train_samples_per_second	2.374
train_steps_per_second	0.074

Figure 3.8: Fine-Tuning Results

Subsequently, a meticulous evaluation was conducted, comparing the performance of both the finetuned and non-finetuned versions of Mistral-7b-instruct on our test set. The results, as detailed in the table below, provide a comprehensive insight into the effectiveness of our fine-tuning strategy.

Table 3.9: Comparative Evaluation of Mistral-7b-instruct Models on test set

	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
Mistral-7b-instruct-v0.1	0.407	0.298	0.362	0.868
Finetuned Mistral-7b-instruct-v0.1	0.587	0.382	0.583	0.872

Table 3.10: Comparison of Mistral-7b-instruct and GPT3.5 on test set

	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
Fine-tuned Mistral-7b-instruct-v0.1	0.587	0.382	0.583	0.872
GPT-3.5	0.593	0.496	0.549	0.905

Table 3.11: Comparison of Mistral-7b-instruct and GPT3.5 Latency

	Latency (ms/query)
Fine-tuned Mistral-7b-instruct-v0.1	40000
GPT-3.5	20000

Data in Table 3.9 demonstrates a performance uplift for the fine-tuned model compared to its non-fine-tuned counterpart. This suggests fine-tuning significantly impacts the response generation process of our base model.

Table 3.10 reveals fine-tuned Mistral-7b-instruct’s strength in answering questions within context, matching even GPT-3.5’s performance. This achievement positions it as a serious competitor in the language model landscape.

This attests to the efficacy of fine-tuned Mistral-7b-instruct in our chatbot system, positioning it as a formidable contender even when benchmarked against established models like GPT-3.5.

3.3 Tests and Discussions

In this section, we conduct a comparative analysis to assess the proficiency of our developed chatbot in responding to a specific service request. To benchmark the capabilities of our chatbot, we compare its responses with two well-established models in the natural language processing domain, namely ChatGPT and Bard.

Here are a few screenshots

How do I apply for a CIP?

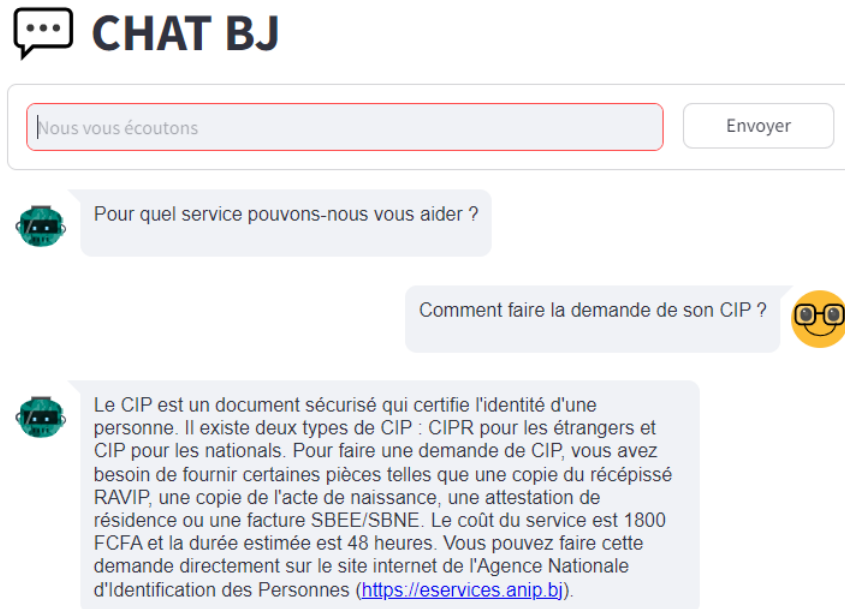


Figure 3.9: Our system's response to the request "How do I apply for a CIP ?"

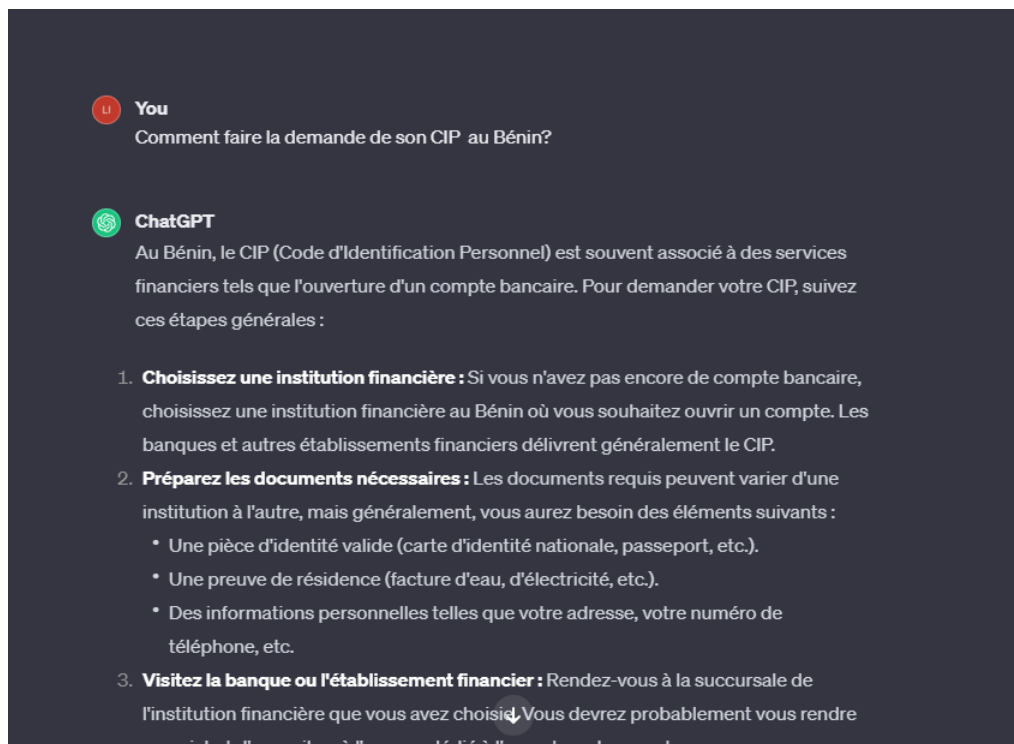


Figure 3.10: ChatGPT's response to the request "How do I apply for a CIP ?"

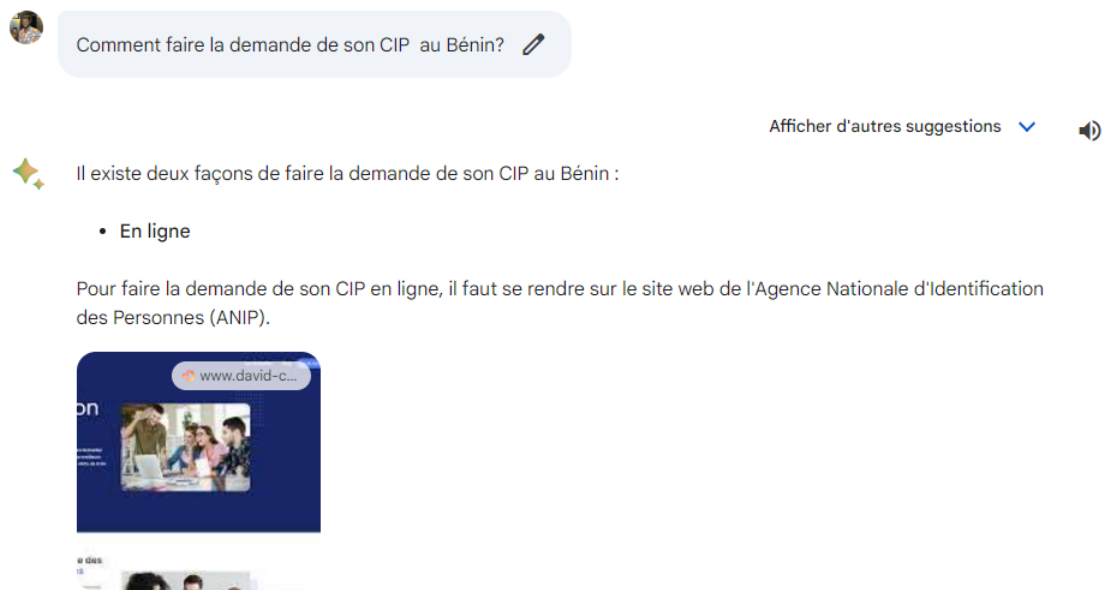


Figure 3.11: Bard's response to the request "How do I apply for a CIP ?" - 1

Il faut également fournir les copies des documents suivants :

- Acte de naissance sécurisé
- Récépissé RAVIP
- Attestation de résidence ou facture SBEE/SONEB

Une fois le formulaire rempli et les documents joints, il faut payer les frais d'établissement, soit 1800 FCFA.

Le CIP est délivré dans un délai de 48 heures.

- En présentiel

Pour faire la demande de son CIP en présentiel, il faut se rendre à l'ANIP de son lieu de résidence.

Il faut se munir des documents suivants :

- Acte de naissance sécurisé
- Récépissé RAVIP
- Attestation de résidence ou facture SBEE/SONEB

Figure 3.12: Bard's response to the request "How do I apply for a CIP ?" - 2

How do I apply for my IFU?



Figure 3.13: Our system's response to the request "How do I apply for my IFU ?" - 1

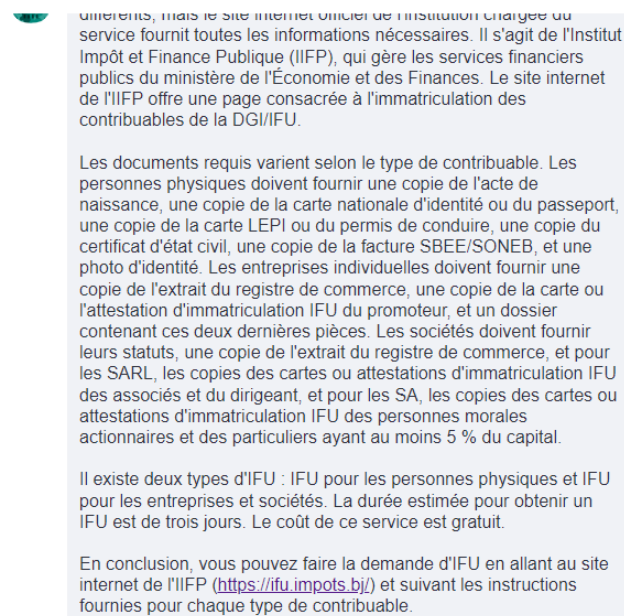


Figure 3.14: Our system's response to the request "How do I apply for my IFU ?" - 2

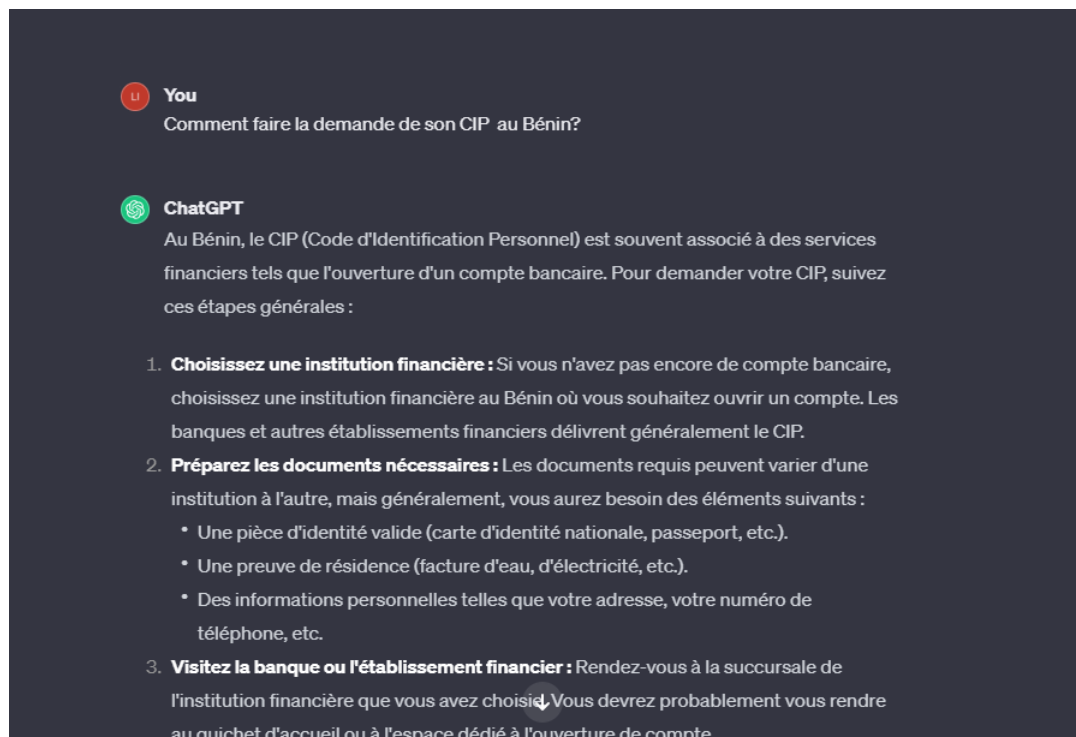


Figure 3.15: ChatGPT's response to the request "How do I apply for my IFU ?"

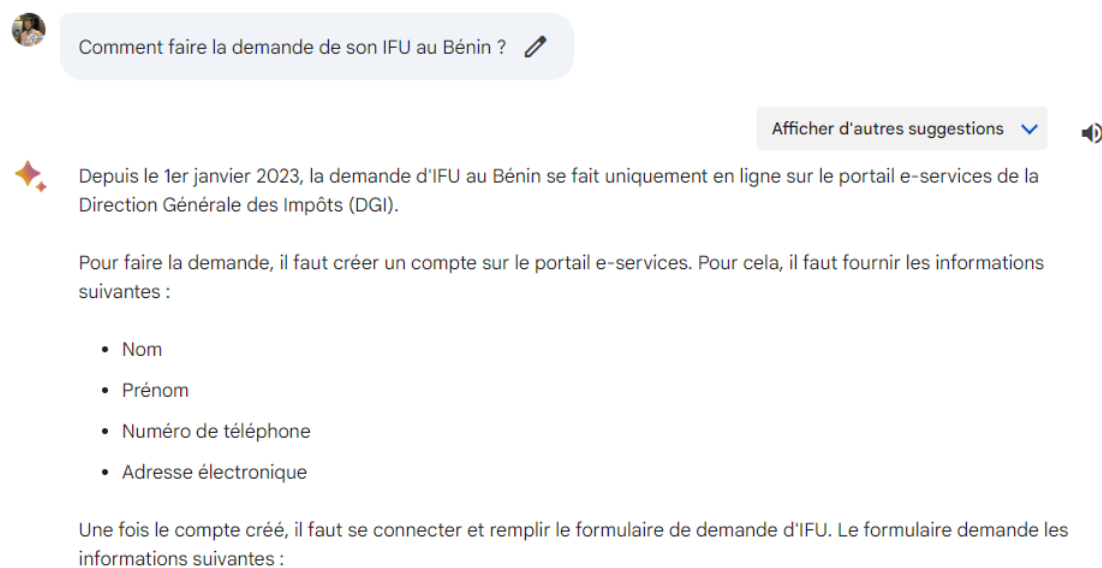


Figure 3.16: Bard's response to the request "How do I apply for my IFU ?" - 1

Il faut également fournir les copies des documents suivants :

- Acte de naissance sécurisé
- CIP
- Attestation de résidence ou facture SBEE/SONEB

Une fois le formulaire rempli et les documents joints, il faut payer les frais d'établissement, soit 1000 FCFA.

La demande est traitée dans un délai de 48 heures. L'IFU est délivré en ligne.

Voici les étapes à suivre pour faire la demande de son IFU au Bénin :

1. Rendez-vous sur le portail e-services de la DGI.
2. Cliquez sur "S'inscrire".
3. Remplissez le formulaire d'inscription et cliquez sur "Valider".
4. Vérifiez votre adresse électronique et cliquez sur le lien de confirmation.
5. Connectez-vous avec votre adresse électronique et votre mot de passe.
6. Cliquez sur "Faire une demande d'IFU".

Figure 3.17: Bard's response to the request "How do I apply for my IFU ?" - 2

We can see that the responses generated by our system (CHAT BJ) are very accurate. These responses are well adapted to our context and provide correct information. Unlike the responses generated by ChatGPT, which are completely out of context, and Bard, which are erroneous in places. However, the response time of our system is longer than those of ChatGPT and Bard.

Conclusion

In conclusion, the outcomes of our study provide a comprehensive view of the chatbot system's performance, encompassing evaluations of retrievers, re-rankers, and generative models. The benchmarking results, coupled with qualitative assessments and captured screenshots, collectively contribute to a nuanced understanding of the chatbot's effectiveness in addressing service requests within the context of the Beninese public administration.

Conclusion & Perspectives

This dissertation has unveiled the contours of an innovative initiative to improve access to administrative services for Beninese citizens via a dedicated chatbot. Through a thorough exploration of the context, the problem, and the defined objectives, this research has led to the design and evaluation of an essential tool in the process of modernizing public administration. The chatbot offers an interactive, instant, and adaptive solution, thus helping to overcome the obstacles inherent in the complexity of online administrative procedures. This advancement is in line with the government's vision of making Benin a leading digital hub in West Africa. By optimizing the interaction between citizens and public services, this project confirms the pivotal role of emerging technologies in building a more transparent, efficient, and accessible administration. The success of this endeavor constitutes a significant step on the path to the digital transformation of the public sector in Benin, testifying to the relevance of technological solutions to meet the concrete needs of citizens and promote innovative governance.

However, this conclusion does not mark the end, but rather the beginning of an evolutionary trajectory. To ensure the continued relevance and progress of our chatbot for requesting administrative services in Benin, several avenues for further development emerge.

1. Integrate a Fon-Fon Speech-To-Speech model, which will be a decisive step towards inclusivity. This will allow the chatbot to be useful to illiterate people, a crucial step in ensuring broader accessibility to administrative services.
2. Choose more powerful models in the future to enhance our system's ability to generate even clearer and more relevant answers, while striking a balance between the size of these models and the inference time required.

Bibliography

- [1] N. Djossou, S. Hekponhoue, M. Senou, and B. Ahodode. “Digitalisation des services publics au benin challenges et opportunités”. (2022), [Online]. Available: <https://includeplatform.net/wp-content/uploads/2022/11/Digitalisation-des-Services-Publics%E2%80%AFau-Benin%E2%80%AFChallenges-et-Opportunit%C3%A9s.pdf> (visited on 12/04/2023).
- [2] ARCEP. “Rapport annuel d’activités 2022 arcep”. (2023), [Online]. Available: <https://arcep.bj/wp-content/uploads/2023/06/Rapport-Annuel-dActivit%C3%A9s-2022-ARCEP-BENIN.pdf> (visited on 12/04/2023).
- [3] I. Talbi. “Introduction aux réseaux de neurones Transformers”, La revue IA. (Feb. 4, 2023), [Online]. Available: <https://larevueia.fr/introduction-aux-reseaux-de-neurones-transformers/> (visited on 06/17/2023).
- [4] E. R. Collins. “Transformer vs LSTM: a helpful illustrated guide”, Be on the Right Side of Change. (Jul. 4, 2023), [Online]. Available: <https://blog.finxter.com/transformer-vs-lstm/> (visited on 09/04/2023).
- [5] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need”, in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- [6] S. R. PhD. “Understanding encoder and decoder LLMs”. (), [Online]. Available: <https://magazine.sebastianraschka.com/p/understanding-encoder-and-decoder> (visited on 08/05/2023).
- [7] C. Raffel, N. Shazeer, A. Roberts, *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer”,
- [8] M. Lewis, Y. Liu, N. Goyal, *et al.*, “BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension”, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. DOI: [10.18653/v1/2020.acl-main.703](https://doi.org/10.18653/v1/2020.acl-main.703).
- [9] *Plongement lexical*, in *Wikipédia*, Page Version ID: 204575906, May 24, 2023. [Online]. Available: https://fr.wikipedia.org/w/index.php?title=Plongement_lexical&oldid=204575906 (visited on 08/05/2023).

-
- [10] Z. Ghrib. "Pre-trained word embedding for text classification: end2end approach", Medium. (Nov. 13, 2020), [Online]. Available: <https://towardsdatascience.com/pre-trained-word-embedding-for-text-classification-end2end-approach-5fbf5cd8aead> (visited on 11/06/2023).
- [11] Ina. "The history of chatbots – from ELIZA to ChatGPT", AI-chatbot software for complex requirements. (Mar. 15, 2022), [Online]. Available: <https://onlim.com/en/the-history-of-chatbots/> (visited on 11/27/2023).
- [12] ELIZA, in *Wikipedia*, Page Version ID: 1186344952, Nov. 22, 2023. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=ELIZA&oldid=1186344952> (visited on 11/27/2023).
- [13] K. Kuligowska. "Chatbot parry, kenneth mark colby | virtual assistant parry | virtual agent parry | chat bot parry | conversational agent parry | (9706)", chatbots.org. (), [Online]. Available: <https://www.chatbots.org/chatbot/parry/> (visited on 11/30/2023).
- [14] Jabberwacky, in *Wikipedia*, Page Version ID: 1166906520, Jul. 24, 2023. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Jabberwacky&oldid=1166906520> (visited on 11/30/2023).
- [15] "Dr. sbaitso", ClassicReload.com. (Dec. 21, 2015), [Online]. Available: <https://classicreload.com/dr-sbaitso.html> (visited on 11/30/2023).
- [16] R. Wallace. "Chatbot a.l.i.c.e., a.l.i.c.e. a.i foundation | virtual assistant a.l.i.c.e. | virtual agent a.l.i.c.e. | chat bot a.l.i.c.e. | conversational agent a.l.i.c.e. | (4510)", chatbots.org. (), [Online]. Available: <https://www.chatbots.org/chatbot/a.l.i.c.e/> (visited on 11/30/2023).
- [17] A. Rodrigues. "A history of SmarterChild", Vice. (Mar. 16, 2016), [Online]. Available: <https://www.vice.com/en/article/jpgpey/a-history-of-smarterchild> (visited on 11/30/2023).
- [18] A. Alsheddi and L. Alhenaki, "English and arabic chatbots: a systematic literature review", *International Journal of Advanced Computer Science and Applications*, vol. 13, Jan. 1, 2022. DOI: 10.14569/IJACSA.2022.0130876.
- [19] S. Hussain, O. Sianaki, and N. Ababneh, "A survey on conversational agents/chatbots classification and design techniques", in Mar. 2019, pp. 946–956, ISBN: 978-3-319-98284-7. DOI: 10.1007/978-3-030-15035-8_93.
- [20] J. Deng and Y. Lin, "The benefits and challenges of ChatGPT: an overview", *Frontiers in Computing and Intelligent Systems*, vol. 2, no. 2, pp. 81–83, 2022, Number: 2, ISSN: 2832-6024. DOI: 10.54097/fcis.v2i2.4465.
- [21] D. M. Ziegler, N. Stiennon, J. Wu, et al., *Fine-tuning language models from human preferences*, 2020. arXiv: 1909.08593 [cs.CL].
- [22] L. Rédaction. "Nouvelle plateforme pour répondre aux préoccupations des Béninois : "Gpt.bj", la grande innovation du Senia 2023", LE MATINAL. Section: Société. (May 12, 2023), [Online]. Available: <https://lematinal.bj/nouvelle-plateforme-pour-repondre-aux-preoccupations-des-beninois-gpt-bj-la-grande-innovation-du-senia-2023/> (visited on 12/15/2023).

-
- [23] ChatGPT, in *Wikipédia*, Page Version ID: 210748136, Dec. 21, 2023. [Online]. Available: <https://fr.wikipedia.org/w/index.php?title=ChatGPT&oldid=210748136> (visited on 12/21/2023).
- [24] Y. Liu, T. Han, S. Ma, *et al.*, “Summary of chatgpt-related research and perspective towards the future of large language models”, *Meta-Radiology*, vol. 1, no. 2, p. 100 017, Sep. 2023, ISSN: 2950-1628. DOI: [10.1016/j.metrad.2023.100017](https://doi.org/10.1016/j.metrad.2023.100017). [Online]. Available: <http://dx.doi.org/10.1016/j.metrad.2023.100017>.
- [25] “What are google bard limitations? the real story december 2023 - bard AI”. Section: Google Bard. (Nov. 17, 2023), [Online]. Available: <https://aibard.online/google-bard-limitations/> (visited on 12/21/2023).
- [26] Y. Gao, Y. Xiong, X. Gao, *et al.*, *Retrieval-augmented generation for large language models: a survey*, 2024. arXiv: [2312.10997](https://arxiv.org/abs/2312.10997) [cs.CL].
- [27] X. Chen, K. Lakhotia, B. Oguz, *et al.*, “Salient phrase aware dense retrieval: can a dense retriever imitate a sparse one?”, in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 250–262. DOI: [10.18653/v1/2022.findings-emnlp.19](https://doi.org/10.18653/v1/2022.findings-emnlp.19). [Online]. Available: <https://aclanthology.org/2022.findings-emnlp.19>.
- [28] T. Chen, M. Zhang, J. Lu, M. Bendersky, and M. Najork, “Out-of-domain semantics to the rescue! zero-shot hybrid retrieval models”, in *Advances in Information Retrieval*, M. Hagen, S. Verberne, C. Macdonald, *et al.*, Eds., vol. 13185, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2022, pp. 95–110, ISBN: 978-3-030-99735-9 978-3-030-99736-6. DOI: [10.1007/978-3-030-99736-6_7](https://doi.org/10.1007/978-3-030-99736-6_7). [Online]. Available: https://link.springer.com/10.1007/978-3-030-99736-6_7 (visited on 01/18/2024).
- [29] Y. Sasazawa, K. Yokote, O. Imaichi, and Y. Sogawa, *Text retrieval with multi-stage re-ranking models*, 2023. arXiv: [2311.07994](https://arxiv.org/abs/2311.07994) [cs.IR].
- [30] “Qu’est-ce que le Web Scraping ? Comment extraire légalement du contenu web”, Kinsta®. (Dec. 19, 2022), [Online]. Available: <https://kinsta.com/fr/base-de-connaissances/web-scraping/> (visited on 08/16/2023).
- [31] “MTEB leaderboard - a hugging face space by mteb”. (), [Online]. Available: <https://huggingface.co/spaces/mteb/leaderboard> (visited on 07/28/2023).
- [32] “Open LLM leaderboard - a hugging face space by HuggingFaceH4”. (), [Online]. Available: https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard (visited on 10/24/2023).
- [33] “What is a vector database?”, Pinecone. (), [Online]. Available: <https://www.pinecone.io/learn/vector-database/> (visited on 06/29/2023).
- [34] A. Takyar. “A guide to parameter-efficient fine-tuning(PEFT)”, LeewayHertz - AI Development Company. (May 16, 2023), [Online]. Available: <https://www.leewayhertz.com/parameter-efficient-fine-tuning/> (visited on 07/31/2023).
- [35] E. J. Hu, Y. Shen, P. Wallis, *et al.*, *Lora: low-rank adaptation of large language models*, 2021. arXiv: [2106.09685](https://arxiv.org/abs/2106.09685) [cs.CL].

-
- [36] K. Pinel-Sauvagnat and J. Mothe, “Mesures de la qualité des systèmes de recherche d’information”, *Revue des Sciences et Technologies de l’Information - Série ISI : Ingénierie des Systèmes d’Information*, Ingénierie des Systèmes d’Information, vol. 18, no. 3, pp. 11–38, 2013. DOI: [10.3166/isi.18.3.11-38](https://doi.org/10.3166/isi.18.3.11-38). [Online]. Available: <https://hal.science/hal-03467001>.
- [37] F. Falcão. “Metrics for evaluating summarization of texts performed by transformers: how to evaluate the...”, Medium. (Apr. 22, 2023), [Online]. Available: <https://fabianofalcao.medium.com/metrics-for-evaluating-summarization-of-texts-performed-by-transformers-how-to-evaluate-the-b3ce68a309c3> (visited on 10/09/2023).
- [38] “Évaluation du modèle - Semaine 2”, Coursera. (), [Online]. Available: <https://www.coursera.org/lecture/generative-ai-with-llms/model-evaluation-8Wvg3> (visited on 01/12/2024).
- [39] E. Yan. “Evaluation & hallucination detection for abstractive summaries”, eugeneyan.com. Section: posts. (Sep. 3, 2023), [Online]. Available: <https://eugeneyan.com/writing/abstractive/> (visited on 10/06/2023).
- [40] P. Mishra. “BERTScore: evaluating text generation with BERT”, Medium. (May 23, 2022), [Online]. Available: <https://towardsdatascience.com/bertscore-evaluating-text-generation-with-bert-beb7b3431300> (visited on 10/10/2023).
- [41] “Utilisation des techniques de deep learning pour l’extraction des concepts à partir des documents textuels”. (), [Online]. Available: http://archives.univ-biskra.dz/bitstream/123456789/13188/1/berri_nacira.pdf (visited on 08/19/2023).
- [42] A. Medjdoubi, “L’analyse du sentiment utilisant le deep learning”, Université Dr. TAHAR MOULAY SAIDA, 2019.
- [43] T. Keldenich. “Fonction d’activation, comment ça marche ? - Une explication simple”, Inside Machine Learning. (Feb. 8, 2021), [Online]. Available: <https://inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simple/> (visited on 06/17/2023).
- [44] F. Rahutomo, T. Kitasuka, and M. Aritsugi, *Semantic Cosine Similarity*. Oct. 29, 2012.

Appendix

Appendix 1 : Deep Learning

Deep learning is a branch of machine learning that focuses on the use of deep artificial neural networks to model and solve complex problems. It is inspired by the architecture and functioning of the human brain to create computer models capable of learning and making autonomous decisions from data.

Machine learning is not useful when working with large data sets. It can't solve crucial artificial intelligence problems such as NLP, image recognition and so on. Feature extraction is one of the big challenges of traditional machine learning models. [41]. This automated feature extraction enables Deep Learning models to achieve a particularly high level of accuracy for computer vision or natural language processing tasks.

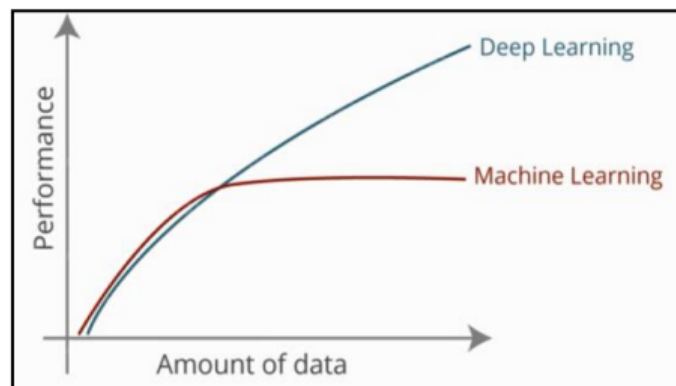


Figure 3.18: Illustration of deep learning and machine learning performance[41].

Neural networks

A neural network is defined as a set of nodes (called neurons) that are interconnected.

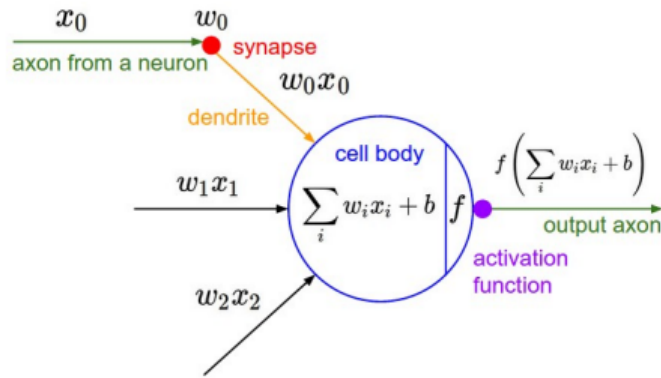


Figure 3.19: Architecture of an artificial neuron [42].

x_i are numerical values representing either input data or output values from other neurons. The weights w_i are numerical values representing either the power value of the inputs or the power value of the connections between the neurons. The artificial neuron makes a weighted sum between the weight (w) and the input value (x), then adds a bias (b). The result is passed to an activation function (f) which adds a certain non-linearity. Deep neural networks are made up of several layers of interconnected neurons (often dozens or more), unlike traditional neural networks which had fewer layers. Each layer of the network processes the information received from the previous layer and transmits it to the next layer, providing a hierarchical and abstract representation of the data. [42].

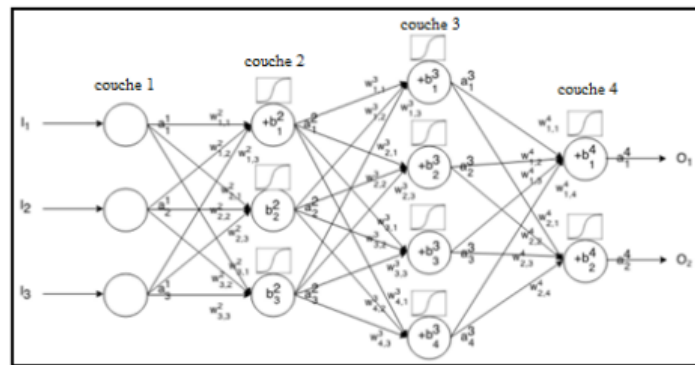


Figure 3.20: Deep neural network topology [42].

The activation functions

In the field of artificial neural networks, the activation function is a mathematical function applied to an output signal from an artificial neuron. The term activation function comes from the biological equivalent "activation potential", a stimulation threshold which, once reached, causes the neuron to respond. The activation function is a non-linear function. There are many types of these functions, including :

- Sigmoid function: This is often used in binary classification problems, where the objective is to predict one of two classes (for example, true or false, positive or negative). The output of the sigmoid function can be interpreted as the probability of belonging to the positive class. If the value returned by the sigmoid function is greater than

0.5, the prediction will be the positive class, otherwise it will be the negative class. $\text{sigmoid}(x) = \frac{1}{1+e^x}$ [43];

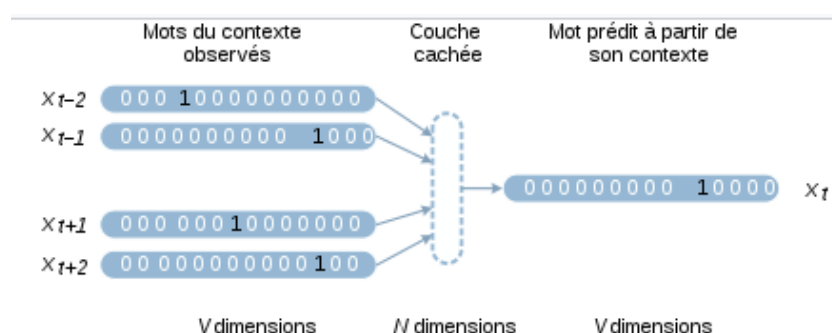
- The ReLu (Rectified Linear Unit) function: This function has recently become the choice for many tasks (particularly in computer vision and NLP). The ReLU function returns 0 if the input x is less than 0 and returns x itself if it is greater than 0. $\text{relu}(x) = \max(0, x)$ [43];
- The Softmax function: this is almost similar to Sigmoid, but calculates the probabilities of the event over 'n' different classes, which will be useful for determining the target in multi-class classification problems. $\text{softmax}(x) = \frac{e^x}{\sum_{k=1}^j e^x}$ [43];

Appendix 2 : Word2Vec and vector representation techniques

Word2Vec is a technique, introduced by Mikolov et al. in 2013, based on self-supervised learning from large corpora of unannotated text. Word2Vec offers two main approaches to learning word vectors: Continuous Bag-of-Words (CBOW) and Skip-gram. Both approaches aim to capture contextual relationships between words, but they differ slightly in their objectives and methods.

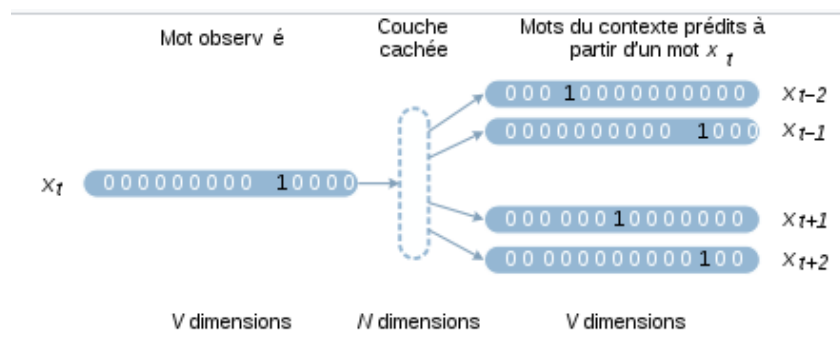
Continuous Bag-of-Words (CBOW)

The CBOW approach attempts to predict a target word based on its context. It uses a fixed context of neighbouring words to estimate the probability of a given word as a target. For example, given the words "I", "eat", "big" and "chocolate" in the sentence "I eat a big chocolate", CBOW tries to predict the target word "a". This method is effective for learning vector representations of frequent words.



Skip-gram

The Skip-gram approach, on the other hand, reverses the CBOW process. It attempts to predict the context from a given word. Using the word 'a' as a starting point, Skip-gram tries to predict neighbouring words such as 'I', 'eat', 'big' and 'chocolate'. This method is best suited to learning vector representations of rare or specialised words.



Appendix 3 : How Transformers learn tasks ?

All of the Transformer models like GPT, BERT, BART, T5, etc have been trained as language models. This means that they have been trained on large amounts of raw text in a self-supervised manner. Self-supervised learning is a type of training in which the output is automatically determined from the model's inputs. This means that humans do not need to label the data.

This type of model develops a statistical understanding of the language on which it was trained, but it is not very useful for specific practical tasks. That is why the general pre-trained model is subjected to a process called transfer learning. During this process, the model is fine-tuned in a supervised manner - that is, using human-annotated labels - for a given task.

Pre-training

Pre-training is the process of training a model from scratch: the weights are initialized randomly and the training starts without any prior knowledge.

Model like BERT is pre-trained on two different but related NLP tasks:

- Masked language modeling (MLM): the algorithm starts by masking a word in a sentence. Then, the algorithm tries to predict which words have the highest probability of being good replacements for the masked word, based on the context;
- Next sentence prediction (NSP): the algorithm predicts whether two given sentences have a logical (or sequential) connection or if there is no relationship between the sentences. For example: Pierre is sick. He has the flu. In the last sentence, the pronoun "he" refers to Pierre. There is therefore a logical causal connection between these two sentences.

Fine Tuning

Fine-tuning is a process where a pre-trained model is adjusted to fit a specific task. The weights of the pre-trained model are adjusted using a labeled dataset for that task. The goal is to specialize the pre-trained model to better understand the specifics of the task and improve its performance.

There are several methods for fine-tuning a Transformer model, the main ones being:

- Full fine-tuning: In this method, all the weights of the model, including the pre-trained weights, are adjusted during training on the labeled data for the specific task. The labeled data is used to calculate the error and update the weights to minimize that error. This allows the model to fully adapt to the specific task, but it can require more labeled data and computational resources;
- Partial fine-tuning: Unlike full fine-tuning, only certain parts of the model are adjusted, while the pre-trained weights remain frozen. The upper layers of the model, which are more task-specific, are often modified, while the lower layers that capture low-level features and general relations are kept. This approach is useful when labeled data is limited and one wants to leverage the prior knowledge learned during pre-training.

Appendix 4 : Similarity Metrics

Similarity measures are the basis of how a vector database compares and identifies the most relevant results for a given query. These measures are mathematical methods for determining how similar two vectors are in a vector space. Similarity measures are used in vector databases to compare the vectors stored in the database and find those that are most similar to a given query vector. Several similarity measures can be used, including :

- Euclidean distance: measures the straight-line distance between two vectors in a vector space. It ranges from 0 to infinity, where 0 represents identical vectors and larger values represent increasingly dissimilar vectors. The formula for Euclidean distance is defined as follows:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.1)$$

- Cosine similarity: this is based on the idea that the similarity between two vectors depends on the angle formed between these vectors in a multidimensional space. More precisely, it measures similarity by calculating the cosine of the angle between the two vectors. It ranges from -1 to 1, where 1 represents identical vectors, 0 represents orthogonal vectors and -1 represents diametrically opposed vectors. The formula is [44]:

$$s(x, y) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} \quad (3.2)$$

- Dot product: often used as a measure of similarity, is an algebraic operation between two vectors that produces a real number. In the context of similarity measurement, the scalar product is a measure of the proximity or similarity between two vectors. In terms of similarity, a high scalar product indicates high similarity between the two vectors, while a low or zero scalar product suggests dissimilarity.

$$x \cdot y = \sum_{i=1}^n x_i y_i \quad (3.3)$$

where :

x and y are two vectors of dimension n ;

x_i and y_i are the i -th components of the vectors x and y .

Appendix 5 : Question Answering domains and techniques

Question answering (QA) is a field of artificial intelligence that aims to automatically answer questions posed by humans.

We can do Question Answering on 2 domains: open-domain and closed-domain.

Open-Domain Question Answering

Open-Domain Question Answering is an artificial intelligence method that aims to answer questions posed by users based on a wide range of available information sources, without being limited to a specific domain. Common techniques used in Open-Domain QA include the use of pre-trained language models and information retrieval.

These techniques allow the system to understand the question being asked, extract the relevant information from the different available sources, and generate a response that is both informative and comprehensive.

Closed-Domain Question Answering

Closed-domain question answering (QA) is an artificial intelligence approach that focuses on answering questions asked by users in a specific, restricted domain.

Unlike open-domain QA, which aims to answer questions on any topic, closed-domain QA is designed to provide accurate, specialized answers in a specific knowledge domain. This knowledge base can be manually constructed by experts or populated by specialized data and reliable sources in the domain.

Based on one of these domains, we can perform QA using 2 approaches: extractive QA or abstractive QA.

Extractive Question Answering

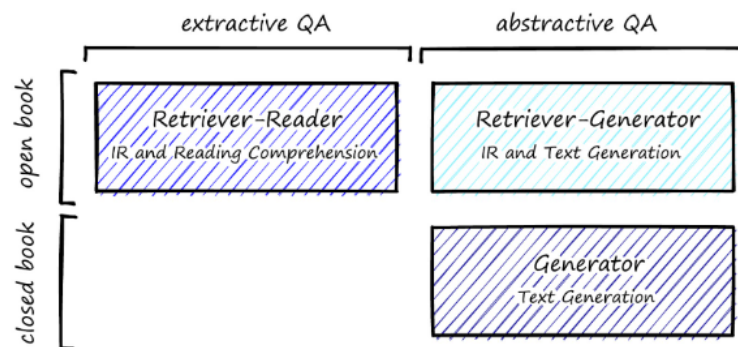
Extractive Question Answering focuses on extracting answers directly from a given source text. It identifies and extracts the portions of the source text that contain the information required to answer a given question using a vector store.

Abstractive Question Answering

Abstract Question Answering is a method that aims to generate complete and consistent answers from source data, using language models to generate new sentences rather than extracting existing information.

Abstract Question Answering can generate more expressive and natural-sounding answers, as it is not limited to simply extracting parts of the source text. It can rephrase and synthesize information to provide more complete and contextualized answers. However, given

that Abstract Question Answering generates answers from the extracted information and context, there is a risk of generating inaccurate or irrelevant answers. Language models can sometimes produce answers that are out of context or incorrect.



Appendix 6: Complete list of administrations indexed in our work

Nous avons collecté des données sur 32 administrations à savoir :

1. le Ministère de la Justice et de la Législation
2. Ministère de l'Economie et des Finances
3. Ministère des Enseignements Secondaires, Technique et de la Formation Professionnelle
4. Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
5. Ministère de l'Eau et des Mines, Ministère de la Santé
6. Ministère des Infrastructures et des Transports
7. Ministère de l'Industrie et du Commerce
8. Ministère des Enseignements Maternel et Primaire
9. Ministère des Petites et Moyennes Entreprises et de la Promotion de l'Emploi
10. Ministère de l'Agriculture, de l'Élevage et de la Pêche
11. Ministère des Sports
12. Ministère du Cadre de vie et du Développement Durable
13. Ministère du Développement et de la Coordination de l'action gouvernementale
14. Ministère du Numérique et de la Digitalisation
15. Ministère du Tourisme, de la Culture et des Arts
16. Autorité de Régulation des Communications Électroniques et de la Poste
17. Ministère du Travail et de la Fonction Publique
18. Ministère de l'Énergie

-
19. Ministère de l'Intérieur et de la Sécurité Publique
 20. Ministère de la Décentralisation et de la Gouvernance Locale
 21. Ministère de la Défense Nationale
 22. Ministère des Affaires étrangères et de la Coopération
 23. Agence Nationale d'Identification des Personnes
 24. Ministère des Affaires Sociales et de la Microfinance
 25. Agence de Promotion des Investissements et des Exportations
 26. Agence pour la Gestion de la Logistique des Officiels
 27. Haute Autorité de l'Audiovisuel et de la Communication
 28. Délégation au Contrôle et à l'Ethique
 29. Cour des compte
 30. Autorité de Régulation du Secteur de la Santé
 31. Autorité de Régulation des Marchés Publics
 32. Présidence de la République du Bénin

Contents

Dedication	ii
Acknowledgements	iii
Résumé	iv
Abstract	v
List of Figures	vi
List of Tables	viii
List of Acronyms	ix
General Introduction	1
1 Related works	3
Introduction	3
1.1 Digitalization in Benin	3
1.1.1 Legal and institutional reforms of the digital economy in Benin	3
1.1.2 Development indicators for the digital economy to the end of 2022	4
1.1.3 Digitalization of public services in Benin: Smart Gouv	5
1.1.4 National Strategy for Artificial Intelligence and Megadata	6
1.2 NLP and Transformers in AI's Pinnacle	7
1.2.1 Transformer architecture	7
1.2.2 Attention	8
1.2.3 The Embeddings	10
1.3 Chatbots	11
1.3.1 History of Chatbots	11
1.3.2 Chatbot approaches	15
1.3.3 Chatbot Techniques	16
1.4 Literature review about chatbot on public services in Beninese Administration	17
1.4.1 Related works	17
1.4.1.1 ChatGPT	17
1.4.1.2 Bard	18
1.4.1.3 GPT BJ	18

1.4.2 Critique of the existing	19
Conclusion	20
2 Material and Methods	21
Introduction	21
2.1 Methods	21
2.1.1 Functional architecture	22
2.1.2 Advanced RAG	23
2.1.2.1 Pre-retrieval	24
2.1.2.2 Retrieval	24
2.1.2.3 Post-retrieval	25
2.1.3 Collecting data	25
2.1.4 Model selection	26
2.1.4.1 Selection criteria for embedding models	26
2.1.4.2 Selection criteria for Re-rankers	27
2.1.4.3 Selection criteria for generative models	27
2.1.5 Data processing	28
2.1.6 Choice of Vector Store and Indexing data	28
2.1.7 Fine-tuning : Parameter-Efficient Fine Tuning (PEFT)	29
2.1.7.1 Adapter	29
2.1.7.2 LoRA	30
2.1.7.3 QLoRA	31
2.1.7.4 Prompt tuning	31
2.1.8 Model evaluation metrics	31
2.1.8.1 Retriever evaluation metrics	31
2.1.8.2 Generator evaluation metrics	32
2.2 Materials	34
Conclusion	36
3 Results and discussions	37
Introduction	37
3.1 Datasets	37
3.1.1 Dataset after scrapping and processing	37
3.1.2 Dataset for Retrievers evaluation	38
3.1.2.1 Semantic search dataset	39
3.1.2.2 Hybrid search dataset	39
3.1.3 Dataset for Generators evaluation	40
3.1.4 Training and Test Set	41
3.2 Benchmarking	42
3.2.1 Benchmarking of retrievers	42
3.2.1.1 Evaluation of semantic approach	42
3.2.1.2 Evaluation of Hybrid approach	43
3.2.2 Benchmarking of re-rankers	43
3.2.3 Benchmarking of Generators	44

3.3 Tests and Discussions	45
Conclusion	50
Conclusion & Perspectives	51
Bibliography	52
Appendix	56
Contents	64
