EJSIT

European Journal of Science, Innovation and Technology

ISSN: 2786-4936

www.ejsit-journal.com

Volume 4 | Number 3 | 2024

Review of Resources Used in Chatbot Models

Akintoba Emmanuel Akinwonmi and Joshua Oluwadamilola Faluyi
Department of Computer Science
The Federal University of Technology, Akure, Ondo State, Nigeria

ABSTRACT

This paper provides a concise overview of some elements in chatbot development, focusing on algorithmic approaches, feature extraction techniques, and datasets employed in existing models. The landscape of chatbot design is explored through diverse machine learning and neural network-based approaches. Feature extraction techniques, crucial for capturing relevant information from input data, are scrutinized for their role in enhancing chatbot performance. Additionally, this paper delves into the pivotal aspect of datasets, elucidating their significance in training and evaluating chatbot models. A comprehensive analysis of existing datasets was described as well as their impact on the robustness and adaptability of chatbots across domains. The detailed understanding of these elements is crucial and fundamental to advancing the capabilities of chatbots and ensuring their seamless integration into diverse applications and industries.

Keywords: Chatbot models, Dataset, Feature extraction, Chatbot resource, Tokenization, Glove, Bag-of-words, Tagging, Machine translation, Intent recognition, Training and optimization, Transfer learning, Corpus

INTRODUCTION

A chatbot is a computer program or an AI-based application designed to simulate human conversation through text or voice interactions (Bani and Singh, 2017). In recent years, chatbot models have experienced significant progress, and this regards the incorporation of advanced algorithmic approaches, novel feature extraction techniques, and the abundance of diverse datasets. These crucial elements are instrumental in enhancing the performance and capabilities of contemporary chatbot systems. The principal algorithmic approaches, feature extraction methods, and datasets that have been utilized in the development of current chatbot models will be discussed in this regard.

LITERATURE REVIEW

Chatbots employ diverse algorithmic techniques to comprehend user inputs and produce suitable responses. Traditional rule-based approaches earlier described in this paper utilize predefined patterns and heuristics to match user queries with predefined responses [23].

In contrast, advanced models frequently leverage machine learning algorithms. These algorithms empower chatbots to grasp intricate linguistic patterns and context, leading to more coherent and contextually relevant responses [26].

Described here are some of the algorithms used in the existing chatbot model.

Support Vector Machine (SVM)

SVMs operate on the basis of the Structural Risk Minimization Principle (SRM), an inductive principle frequently applied in machine learning. In many machine learning scenarios, it is necessary to select a generalized model from a limited dataset, which may lead to the issue of overfitting [24].

Overfitting occurs when a model learns the intricacies and noise of the training data to an extent that it adversely affects the model's performance on new, unseen data [37]. The SRM principle is said to address this problem by balancing the model's complexity against its success at fitting the training data [24].

This principle was first set out by Vladimir and Alexey (1974). SVMs have gained immense popularity for text classification and intent identification tasks. They enable assessing the likelihood of input belonging to a particular category. Cross-validation is commonly employed to evaluate this algorithm, involving resampling to assess machine learning models on a limited data sample. Accuracy assessment relies on the training and test sets. Precision and recall metrics are also used to evaluate the model's performance [5]

The Support Vector Machine (SVM) is a supervised classification technique that aims to create a clear boundary between different classes. In 2-dimensional space, this boundary is referred to as a line, while in 3-dimensional space; it becomes a plane [53].

In higher dimensions exceeding 3, the boundary is known as a hyperplane. When dealing with two classes of data, the SVM seeks to find the boundary that maximizes the margin or distance between the two classes. Although multiple planes can separate the classes, only one plane can achieve the maximum margin between them [53].

Mathematical representation:

$$MINIMIZE_{a_0,\dots,a_m}: \sum_{j=1}^{n} MAX\left\{0, 1-\left(\sum_{i=1}^{m} a_i x_{ij} + a_0\right) y_j\right\} + \lambda \sum_{i=1}^{m} (a_i)^2 \qquad (1)$$

where

n is number of data points

m is number of attributes

x_{ij} is ith attribute of j^{th} data point

y_j is 1 if data point is blue, -1 if data point is red

The above mathematical expression could be utilized in solving for a lower and upper boundary of a hyperplane.

Based on experiments, it was noted that SVMs consistently outperform similar algorithms such as K-nearest neighbor (K-NNs) and Naïve Bayes considering its level of computational demand [4].

Naive Bayes Algorithm

The primary aim of the Naive Bayes algorithm is to categorize texts into specific groups, allowing chatbots to discern the intent of a user and thereby reducing the potential range of responses. Ensuring the accuracy of this algorithm is crucial, given that intent identification is a fundamental step in chatbot conversations. The algorithm's reliance on commonality implies that certain words carry more weight for particular categories based on their frequency of appearance in those categories [4].

To evaluate the algorithm's performance, the most direct method is to employ k-fold cross-validation. This involves training the chatbot with specific inputs and their corresponding categories and then using a test set to assess how accurately the chatbot can classify new inputs. Various metrics like confusion matrices, accuracy, precision, and recall can be employed to evaluate the algorithm's effectiveness.

A limitation of the Naive Bayes algorithm is its utilization of a 'bag of words' approach, which considers words as an unordered set and selects the most significant ones to determine the input's category. As a result, it disregards the word order, which may lead to differences in the assigned category for inputs with word rearrangements [6]. This technique is also referred to as an emerging method used to analyze machine learning and deep learning models, providing insights into their decision-making processes [43].

The Naive Bayes Classifier is inspired by Bayes Theorem which states in (2)

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B)} \tag{2}$$

This equation can be rewritten as in (3), solving for the probability of y given input features X

$$P(y|X) = \frac{P(X|y)*P(y)}{P(X)} \tag{3}$$

where
X is input variables
y is output variable

Because of the naive assumption (hence the name) that variables are independent given the class, P(X|y) can be rewritten as follows:

$$P(X|y) = P(x_1|y) * P(x_2|y) * \ldots * P(x_n|y) \tag{4}$$

Also, in solving for y, P(X) is a constant which means that it can be removed from the equation and introduces proportionality. This leads to (5).

$$P(y|X) \propto P(y) * \prod_{i=1}^{n} P(x_i|y) \tag{5}$$

The equation proceeding is arrived at, the goal of Naive Bayes is to choose the class *y* with the maximum probability. **Argmax** is simply an operation that finds the argument that gives the maximum value from a target function. In this case, the maximum y value is searched for.

$$y = argmax_y[P(y) * \prod_{i=1}^{n} P(x_i|y)] \tag{6}$$

Deep Neural Network

This deep neural network represents an artificial neural network with multiple layers situated between the input and output layers. It belongs to a broader family of machine learning methods that rely on artificial neural networks for representation learning [58].

Inspired by the human brain, this algorithm utilizes interconnected layers of artificial neurons that learn features from data and collaborate to generate meaningful outputs. Neural networks are data-intensive, necessitating large volumes of data to effectively learn patterns and trends. To assess this algorithm's performance, it is essential to evaluate whether the chatbot produces valid responses to inputs, sustains coherent conversations, meets user needs, and exhibits linguistic characteristics similar to those of a human to a reasonable extent [58]. Adapting the Turing test might be a suitable approach for such evaluations.

One drawback of neural networks lies in their lack of explainability. Identifying which specific neuron contributed to a prediction or processed a particular feature is not straightforward. Additionally, neural networks heavily rely on vast amounts of data to undergo iterative learning processes, ensuring their responses are as valid as possible.

Markov Chains

The Markov chain is a stochastic model that characterizes sequences of possible events, with each event's probability predominantly dependent on the state reached in the preceding event. This model finds wide application in text generation and Chatbots. It operates by calculating the likelihood of transitioning from one state to another [60]. One of its notable advantages is its simplicity and summarization, as it can be conveniently represented using matrices.

In the context of Chatbots, Markov chains work by defining the order of chains, where the order refers to the number of words grouped together in each chain. For instance, a first-order chain has one word, while a third-order chain contains a group of three words. Higher-order chains more accurately represent the training data, resulting in less variance, while lower-order chains are more random and produce variable output. To evaluate the performance of Markov chain-based Chatbots, tests involve grammatical parsing, output analysis, and user feedback [60]. By constructing responses based on statistical probabilities, Markov chains generate more realistic and plausible outputs.

However, since Markov chains combine probabilities and randomness, there may be instances where the output lacks coherence. Identifying such situations is crucial, and the algorithm should be retrained to ensure the Chatbot avoids generating incomprehensible responses [61].

Natural Language Processing (NLP)

Natural language processing (NLP) is a technique concerned with the interaction between computers and human language, involving programming computers to understand vast amounts of data [13]. Its significance for chatbots lies in enabling them to comprehend and interpret text inputs, thus facilitating human-like conversations where users perceive the interaction as human-to-human [45].

NLP algorithms aim to acquire knowledge through machine learning and extensive data derived from conversations. Research studies, such as the one by [61] on chatbot design techniques in speech conversation systems, highlight NLP's role in helping bots understand text data, grammar, intent, and sentiment. NLP offers various functionalities, including text summarization, word vectorization, topic modeling, and sentiment polarity analysis [61].

Testing NLP algorithms primarily involves evaluating the chatbot's communication abilities, which is a subjective process without a standardized benchmark. User satisfaction and feedback analysis are commonly used methods to assess algorithm performance. However, relying solely on such data might not provide a comprehensive evaluation. Alternatively, assessing the algorithm's ability to mimic human linguistic conversations, in regards to the Turing test (1950), can also be considered as a means to gauge its effectiveness.

Artificial Intelligence Markup Language (AIML)

AIML, an abbreviation for Artificial Intelligence Markup Language, is derived from Extensible Markup Language (XML) and serves as the foundation for creating conversational agent software [52]. It is an XML dialect that contains a set of rules defining the chatbot's conversational capabilities. These rules are used with a linguistic communication understanding processor, allowing the chatbot to analyze and respond to user queries. As more rules are added, the chatbot's intelligence and conversational abilities improve [52].

The development of AIML can be attributed to Richard Wallace and a global community of free software enthusiasts between 1995 and 2002. AIML originally formed the basis for an extensively expanded version of Eliza, known as "A.L.I.C.E." (Artificial Linguistic Internet Computer Entity). This chatbot achieved significant success, winning the annual Loebner Prize Competition in Artificial Intelligence three times and becoming the Chatterbox Challenge Champion in 2004 [52].

Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is a neural network class that has garnered significant attention due to its remarkable performance in addressing real-world machine learning problems, particularly those involving sequential data and input-output data with varying lengths [21].

Alex Graves (2012) explains in "Supervised Sequence Labelling with Recurrent Neural Networks" that RNNs are powerful sequential learners and are commonly employed in natural language processing tasks such as machine translation [55]. They also find applications in handwriting recognition/generation [21], speech recognition [21], and human activities modeling [43].

RNNs are specifically designed to handle sequential data, unlike other data types, where features are assumed to be order-independent. To understand RNNs fully, a comprehension of "normal" feed-forward neural networks and sequential data is essential [12].

Sequential data refers to ordered data where related items follow a specific sequence. Time series data is a common example of sequential data, representing a series of data points listed chronologically [1].

The names "RNNs" and "feed-forward neural networks" are derived from the way they handle information flow. In feed-forward neural networks, information flows in a single direction, from the input layer to the output layer through the hidden layers. Such networks lack memory of past inputs, making them less effective in prediction tasks [12].

RNNs, on the other hand, process information in a loop, considering both the current input and what they have learned from previous inputs. They maintain hidden states that represent past knowledge, updating them at each time step to reflect changes in the network's understanding of the past [21].

Unlike other networks that follow linear paths during feed-forward and back-propagation, RNNs employ recurrence relations and Back-Propagation through time to learn [62]. Each time step in an RNN consists of fixed activation function units, each with an internal hidden state representing the network's past knowledge. This hidden state is updated based on a recurrence relation that captures the change in the network's understanding of the past [62].

$$h_t = f_w(X_t, h_t - 1) \tag{7}$$

where
h_t is the new hidden state
$h_t - 1$ is the old hidden state
X_t is the current input
f_w is the fixed function with trainable weights

At each sequence time step, the new hidden state is calculated using the aforementioned recurrence relation given. The new generated hidden state is then used to generate a new hidden state and so on.

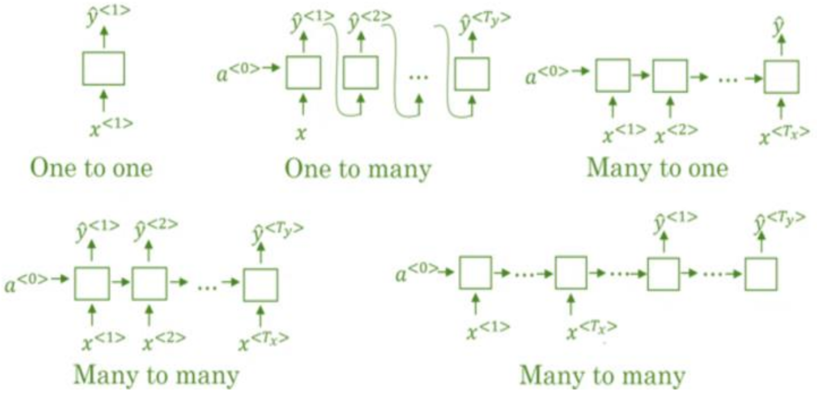

$\hat{y}^{<1>}$
$x^{<1>}$
One to one
$\hat{y}^{<1>}$ $\hat{y}^{<2>}$ $\hat{y}^{<T_y>}$
$a^{<0>}$
x
One to many
$\hat{y}$
$a^{<0>}$
$x^{<1>}$ $x^{<2>}$ $x^{<T_x>}$
Many to one
$\hat{y}^{<1>}$ $\hat{y}^{<2>}$ $\hat{y}^{<T_y>}$
$a^{<0>}$
$x^{<1>}$ $x^{<2>}$ $x^{<T_x>}$
Many to many
$\hat{y}^{<1>}$ $\hat{y}^{<T_y>}$
$a^{<0>}$
$x^{<1>}$ $x^{<T_x>}$
Many to many

Figure 1: Types of Recurrent Neural Network [62]

i. Individualized instruction: One-to-One RNNs (Tx=Ty=1) are the most basic, with a single input and a single output. It functions as a traditional neural network with fixed input and output sizes. Image Classification contains the One-to-One application.

ii. One-to-Many: One-to-Many RNNs (Tx=1,Ty>1), when given a single input, a one-to-many RNN produces multiple outputs. It accepts a fixed input size and returns a series of data outputs. It has applications in music generation and image captioning.

iii. Many-to-One: Many-to-one RNNs (Tx>1,Ty=1), When a single output is required from multiple input units or a sequence of them, many-to-one is used. A fixed output requires a sequence of inputs. Sentiment Analysis is an example of a Recurrent Neural Network of this type.

iv. Many-to-many: Many-to-Many RNNs (Tx>1,Ty>1), is a method for generating a series of output data from a series of input units. This type of RNN is further subdivided into two subcategories:

Note that for Equal Unit Size, the number of input and output units is the same in this case. Name-Entity Recognition is a popular application. And for Unequal Unit Size, the inputs and outputs have different numbers of units. Its use can be found in Machine Translation.

The RNN technique is in basically two architectures;

a. Long Short-Term Memory (LSTM):

The LSTM Recurrent Neural Network stands out as a unique variant of recurrent neural networks, allowing it to selectively retain patterns for extended periods [49]. Due to this capability, it is an excellent choice for modeling sequential data, particularly in grasping complex dynamics of human activities. The part of the network responsible for long-term memory is referred to as the cell state. Due to the recursive nature of the cells, previous information is stored within it. The forget gate placed below the cell state is used to modify the cell states. The forget gate outputs values saying which information to forget by multiplying 0 to a position in the matrix. If the output of the forget gate is 1, the information is kept in the cell. The input gates determine which information should enter the cell states. Finally, the output gate tells which information should be passed on to the next hidden state.

[48] devised an LSTM-based RNN model to address the challenge of sentence representation learning in Information Retrieval. Leveraging the LSTM's ability to retain long-term memory, the LSTM-RNN model effectively captures the semantic meaning of entire sentences while identifying the most salient and less important words as it processes each term sequentially. The LSTM-RNN model takes individual terms from a sentence in a sequential manner and embeds them into a semantic vector, departing from simple summation. Instead, [48] used a sequence of letter trigrams as inputs to generate the embedding vector for the entire sentence. This approach helps the model preserve essential long-term memory containing valuable information while discarding less salient terms. In their experiments, these researchers utilized the long short-term memory architecture, with the last output of the hidden layer representing the full sentence [63].

b. Gated Recurrent Units (GRUs):

The Gated Recurrent Units (GRU) as a variant of recurrent neural networks was introduced in 2014. GRUs, like LSTMs, were specifically designed to tackle long-term dependencies [65]. However, GRUs has a simpler structure compared to LSTMs. Both architectures have been successfully applied in polyphonic music modeling and speech recognition tasks [8]. The results indicate that GRUs are as effective as LSTMs in these applications [8].

The Gated Recurrent Unit (GRU) is an architecture of the Recurrent Neural Network (RNN) that is considered to have advantages over long short-term memory (LSTM) in some cases [66]. GRU uses less memory and is faster. However, LSTM is considered to be more accurate when using datasets with longer sequences [66].

FEATURE EXTRACTION TECHNIQUES

Preparing textual data for chatbot models involves a crucial step known as feature extraction. The feature extraction Techniques such as Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) are fundamental approaches used to transform text into numerical representations [29]. The emergence of deep learning has brought word embeddings like Word2Vec, GloVe, and FastText into the limelight. These embeddings excel in capturing semantic relationships between words, significantly enhancing the chatbot's capability to comprehend and respond to user queries more accurately (Ajay, 2020). Moreover, contextual word embeddings, including ELMo and GPT (Generative Pre-trained Transformer), take into account the surrounding context of words, further elevating the chatbot's performance [31]. The following described techniques are discussed;

Bag of Words (BoW)

The Bag of Words (BoW) technique offers a straightforward and efficient approach to converting textual data into numerical representations. It involves creating a vocabulary from all distinct words in the corpus and then tallying the occurrence of each word in every document [22]. The resulting vector reflects the word frequencies, disregarding the word order. BoW finds extensive application in tasks like text classification, sentiment analysis, and information retrieval. However, its simplicity comes with some drawbacks, notably the inability to capture word order and contextual information [2].

To demonstrate the Bag of Words (BoW) technique for chatbot research, here is a sample dataset of chatbot conversations to be considered [2].

Suppose the following three conversations is given:

1. User: Hi there! Can you tell me the weather forecast for today? Bot: Sure! The weather forecast for today is partly cloudy with a chance of rain in the evening.

2. User: How can I reset my password? Bot: To reset your password, go to the login page and click on the "Forgot Password" link. Follow the instructions to set a new password.

3. User: What are your opening hours? Bot: We are open from 9 AM to 6 PM on weekdays and 10 AM to 3 PM on weekends.

Applying the Bag of Words technique to represent these conversations as numerical features:

Step 1: Create Vocabulary First; we create a vocabulary from all the unique words in the conversations. The vocabulary would be:
["hi", "there", "can", "you", "tell", "me", "the", "weather", "forecast", "for", "today", "sure", "partly", "cloudy", "with", "a", "chance", "of", "rain", "in", "evening", "how", "reset", "my", "password", "to", "go", "login", "page", "click", "on", "forgot", "follow", "instructions", "set", "new", "what", "are", "your", "opening", "hours", "we", "open", "from", "9", "am", "6", "pm", "weekdays", "and", "10", "3", "weekends"]

Step 2: Count Word Frequencies Next, we count the frequency of each word in each conversation and represent them as vectors:
Conversation 1: [1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Conversation 2: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Conversation 3: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

The aforementioned vectors represent the frequency of each word in the respective conversation. Notice that the order of words is not preserved in the BoW representation.

Word2Vec

It is a word embedding technique that converts words into compact vectors within a continuous vector space [39]. Word2Vec utilizes numerical vectors to represent words, capturing the semantic associations between them by analyzing their contextual patterns in extensive text datasets [39].

Word2Vec employs a neural network to acquire word embeddings by considering the context in which words are used. During the training of Word2Vec, word embeddings are learned through two different approaches: one involves predicting the context words based on a target word, and the other entails predicting the target word given a set of context words within a given text [34]. These resulting word embeddings can serve as valuable features in various natural language processing (NLP) tasks, including sentiment analysis, text classification, machine translation, and information retrieval [34].

Term Frequency-Inverse Document

Frequency (TF-IDF)

TF-IDF stands as another technique for extracting features, which strives to indicate the significance of words within a document concerning the entire corpus. It calculates a weighted score for each word by considering its frequency in the document (TF) and its rarity in the corpus (IDF) [50]. When a word possesses high TF-IDF values, it signifies that the word is both frequently used in the document and infrequently found in the overall corpus, making it more distinctive and informative. This approach overcomes some of the limitations of the Bag-of-Words (BoW) method, particularly in capturing the contextual importance of words within a document [50].

[4] developed a chatbot for Information Service of New Student Admission Using Multinomial Naïve Bayes Classification and TF-IDF Weighting. In the work, TF-IDF was adapted for the feature extraction of the given data, which calculates information amount by its occurrence probability and aims to enhance the chatbot's ability to understand and respond to user queries effectively. It also helps to reduce noise in the processed words by downplaying the significance of common and frequently occurring words (e.g., "the," "and," "is"), which helps the chatbot to focus on the unique and informative aspects of the user's input, leading to more accurate responses.

However, it was generally noted that this technique has a limitation, in the sense that it does not inherently recognize named entities (e.g., names of people, places, organizations), which are crucial for understanding and responding accurately to user queries that involve specific entities [4].

GloVe

GloVe, or Global Vectors for Word Representation, is a significant word embedding method developed by Stanford researchers Jeffrey Pennington, Richard Socher, and Christopher D. Manning in 2014. This technique falls under the category of distributed word representations, wherein words are assigned continuous vector spaces in a manner that preserves meaningful semantic relationships between them through geometric properties of the vectors [47].

GloVe's main concept involves constructing a comprehensive word-word co-occurrence matrix from a vast text corpus [17]. Within this matrix, each element denotes the frequency of a word occurring alongside another word in context. By employing a factorization procedure on this matrix, the model acquires word embeddings in a reduced-dimensional space.

Consequently, these word vectors demonstrate meaningful relationships between words, as reflected through vector arithmetic [17].

[7] conducted a research study where they developed a chatbot system. The system integrates RASA NLU and neural network (NN) methods and utilizes global vector for word representation to implement entity extraction after intent recognition. The developed system enables automatic learning and answering of finance-related questions through experimental comparison and validation.

Although GloVe is acknowledged for its ability to capture linear relationships between words, it may not fully encompass more intricate semantic relationships or word analogies. Additionally, the model necessitates a substantial amount of training data to achieve accurate embeddings efficiently. Also, the word vectors learned by GloVe are static and fail to capture context-dependent word meanings [9].

FastText

FastText is a pioneering word embedding technique and text classification framework, originated from the collaborative efforts of Facebook's AI Research (FAIR) team.

[10] introduced this groundbreaking approach in 2016. Unlike traditional word embeddings, FastText enhances its capabilities by integrating subword information, enabling exceptional performance with morphologically rich languages and adeptly handling out-of-vocabulary (OOV) words [10].

FastText dissects words into smaller subword units, commonly known as character n-grams. By leveraging these subword embeddings, the model becomes adept at capturing morphological variations and relationships among words that share similar subword components, resulting in enhanced robustness to out-of-vocabulary (OOV) words [19].

According to the research work done by [19], a comparison was made between the word2vec, GloVe, FastText, the performance evaluation of these subwords three results revealed that FastText outperformed both GloVe and word2vec with an impressive accuracy of 97.2 percent, surpassing their respective accuracies of 95.8 and 92.5 percent. FastText exhibits the remarkable capability of generating word representations for words not present in the training data, effectively overcoming out-of-vocabulary challenges [19]. The words absent during the training process, FastText employ n-grams to form a collection of syllable sequences, enabling the creation of embedding vectors [19].

Dataset

The availability of diverse and extensive datasets has been instrumental in the development of chatbot models. Several datasets, such as the Cornell Movie Dialogs Corpus, OpenSubtitles, and Ubuntu Dialogue Corpus, provide dialogues and conversational data suitable for training and evaluating chatbots.

Furthermore, chatbot-specific datasets, like the Persona-Chat dataset, include persona-based conversations, enabling chatbots to exhibit more personalized responses [36].

Cornell Movie Dialogs Corpus

The Cornell Movie Dialogs Corpus holds significant popularity as a dataset extensively utilized for training and evaluating chatbot and conversational AI models [36][67]. Developed by Cornell University researchers, it comprises an extensive compilation of fictional conversations sourced from diverse movie scripts [67].

The dataset comprises over 220,000 conversational exchanges extracted from more than 600 movies, covering various genres and themes. Each conversation is represented as a series of alternating lines of dialogue between two characters, making it well-suited for modeling multi-turn conversations (Mehenni *et al*, 2018).

The diversity of conversations within the Cornell Movie Dialogs Corpus stands out as a major strength. It encompasses interactions between various characters, each characterized by their unique personalities, backgrounds, and speaking styles [67]. This rich diversity poses a stimulating and intriguing challenge for training chatbot models to generate contextually appropriate responses [67].

While the Cornell Movie Dialogs Corpus is a valuable resource, it also has some limitations. As the conversations are fictional and scripted, they may not fully represent real-world interactions, and some dialogues could be less varied or dynamic compared to naturally occurring conversations [17].

Ubuntu Dialogue Corpus

Ubuntu Dialogue Corpus is a valuable dataset for training chatbot models. It contains conversations from Ubuntu users seeking technical support, making it suitable for handling technical queries [68]. The dataset's real-world scenarios and large-scale interactions offer significant opportunities for developing effective and contextually relevant chatbots for customer service and technical support applications**.**

With a vast collection of over 1.8 million multi-turn conversations, the corpus is notably diverse. It presents conversations in context-response pairs, where each context encapsulates the preceding turns leading to the response [68].

Like many real-world datasets, the Ubuntu Dialogue Corpus may exhibit noise, spelling errors, and inconsistencies common in user-generated content. To ensure the training data's quality, preprocessing and data cleaning are essential steps [29].

OpenSubtitles

OpenSubtitles is a widely used and extensive dataset, which serves as a valuable resource for training and evaluating natural language processing (NLP) models, especially those geared towards dialogue and conversation generation [59]. As it is been derived from movies and TV shows, this dataset contains subtitle data, providing a rich and authentic collection of diverse conversational interactions. Here's a comprehensive review of the OpenSubtitles dataset [59].

Based on [44] discuss on this dataset, it was noted to include multi-turn conversations, allowing NLP models to understand the contextual flow and history of conversations. This context enables chatbot models to generate more relevant and contextually appropriate responses, enhancing the overall conversational experience.

Given its size and diversity, the OpenSubtitles dataset might exhibit noise, errors, and inconsistencies that are common in user-generated content. To ensure the training data's quality and the optimal performance of trained models, preprocessing and data cleaning become crucial steps [63].

Persona-Chat Dataset

The Persona-Chat dataset, introduced by [36], in the research work which has gained significant popularity and widespread adoption as a valuable resource for training and evaluating chatbot models [36].

With more than 162,000 persona-based dialogues, the Persona-Chat dataset proves to be a substantial asset for training and evaluating conversational AI models. It is structured as a collection of conversations, and each dialogue incorporates multiple turns of interaction [41].

A distinctive characteristic of the Persona-Chat dataset is its incorporation of persona descriptions for every participant engaged in the dialogue [67]. Each persona is a brief paragraph outlining the individual's characteristics, preferences, and background. For instance, a persona could encompass information such as "I have a fondness for dogs, enjoy hiking, and am enthusiastic about gardening [67].

[53] noted that the Persona-Chat dataset has been widely used and influential, it does have limitations. Persona descriptions are often subjective and might not fully capture the complexity of real-world personalities. Also, evaluating smooth persona switching within a single conversation remains a challenge [53].

Natural Language Processing (NLP) Techniques Used in Chatbot

Natural Language Processing (NLP) is a branch of artificial intelligence dedicated to empowering machines with the ability to comprehend, interpret, and produce human language. Within this field, NLP plays a crucial role in shaping the evolution of chatbots, which are software applications engineered to replicate human-like dialogues with users [38].

The ubiquity of chatbots spans diverse sectors, ranging from customer support and virtual assistants to interactive storytelling and language education. Their wide-ranging applications showcase the significant impact NLP has had on enhancing human-machine interactions and transforming the way we engage with technology in various aspects of our lives [20].

Chatbots leverage various natural language processing (NLP) techniques to understand user inputs, generate appropriate responses, and engage in meaningful conversations. Here are some common NLP techniques used in chatbots:

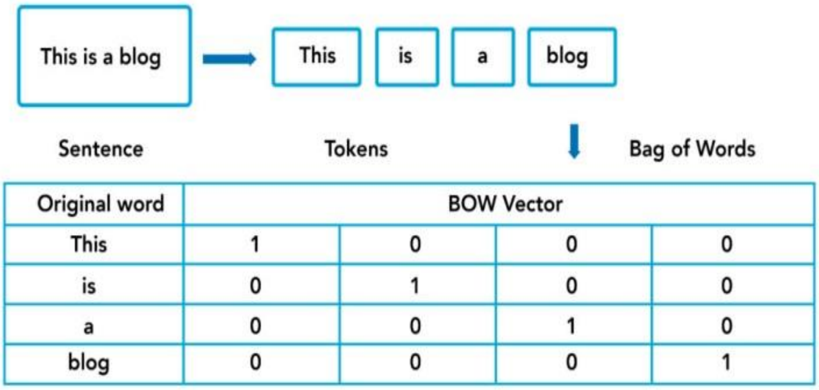
a. Tokenization

Tokenization is the process of breaking down text into smaller units, known as tokens. Tokens can be words, subwords, or characters. Tokenization helps in preparing the input text for further NLP tasks [57].

Tokenization is mostly understood as the first step of any kind of natural language text preparation. The primary objective of this initial (pre-linguistic) undertaking is to transform a sequence of symbols into a sequence of computational units referred to as tokens [23].

To enable our computer to comprehend any given text, we must deconstruct the words into a format that the machine can grasp. This is where the notion of tokenization comes into play before further process of the data to arrive at needed state.

In this stage, the text data that is presented in paragraphs and sentences for training is divided into smaller units called tokens that analyzed and understood, while disregarding certain characters such as punctuation marks. These tokens are considered to be linguistically indicative of the text.



Original word	BOW Vector			
This	1	0	0	0
is	0	1	0	0
a	0	0	1	0
blog	0	0	0	1

Figure 2: Tokenization

b. Part-of-Speech (POS) Tagging

POS tagging involves labeling each word in a sentence with its corresponding part of speech, such as noun, verb, adjective, etc. This information is crucial for understanding the grammatical structure of sentences [30].

Part-of-speech tagging has seen substantial advancements in methodologies and techniques. Rule-based approaches employ predefined linguistic rules to assign parts of speech. However, statistical and machine learning methods, such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs), have gained prominence due to their ability to learn from data and adapt to varying contexts [33].

Recently, neural network-based approaches, particularly recurrent and transformer-based models, have demonstrated state-of-the-art performance, surpassing traditional methods [33].

According to [46], while part-of-speech tagging offers remarkable advantages, it also presents challenges that researchers and practitioners have tirelessly addressed. Ambiguity is a major hurdle; words often have multiple possible parts of speech based on context, requiring sophisticated algorithms to make accurate decisions. Additionally, languages with complex morphologies or those lacking strict word boundaries pose difficulties in accurate tagging. Handling rare or previously unseen words also poses a challenge, necessitating effective strategies for handling out-of-vocabulary terms [46].

c. Named Entity Recognition (NER)

Named Entity Recognition (NER) is a central cornerstone in the domain of Natural Language Processing (NLP), orchestrating the conversion of unprocessed text into organized data. This complex procedure revolves around the identification and categorization of distinct entities within the text [42].

NER identifies and classifies entities (e.g., names of persons, organizations, locations) in the text. This is essential for extracting specific information from user queries or generating meaningful responses [40].

[40] also describes these entities to span a diverse array of elements, encompassing personal names, organizational entities, geographic locations, dates, and a myriad of other descriptors.

NER is not without its challenges. One prominent issue is the contextual ambiguity of named entities, where the same string can belong to multiple categories based on the context. Handling unseen or rare entities is also a hurdle. Morphological variations, multilingualism, and domain adaptation present additional complexities [16]. NER models may struggle with nested entities, such as an organization within a location. Balancing precision and recall is crucial, as missing important entities or misclassifying them can impact downstream applications [16].

d. Dependency Parsing

Dependency parsing stands as a pivotal technique within the realm of Natural Language Processing (NLP), offering profound insights into the structural and semantic intricacies of textual data [37]. Dependency parsing examines the grammatical arrangement within a sentence and defines connections among words through dependency arcs. This process aids in comprehending both the sentence's syntax and its underlying meaning [37].

Dependency parsing faces a range of difficulties described by [39]. Ambiguity and polysemy, where words take on multiple meanings, introduce intricacy. Addressing non-projective structures, where dependencies intersect, demands distinct algorithms. The Parsing languages with flexible word order, morphological diversity, or sparse punctuation presents extra challenges [39]. Creating parsers for languages with limited resources continues to be a work in progress, along with effectively handling words not found in the parser's vocabulary [39].

e. Machine Translation

Machine Translation (MT) stands as a transformative force within the realm of Natural Language Processing (NLP), revolutionizing cross-lingual communication and content accessibility [6]. Machine translation enables chatbots to handle multilingual conversations by translating text between different languages [44].

Machine translation serves as a pivotal tool in overcoming language barriers, facilitating smooth communication among individuals from different linguistic backgrounds [54]. It goes beyond geographical confines, nurturing worldwide connectivity and the availability of information. The role of MT is crucial in dismantling language isolation, fostering cross-cultural comprehension, and driving forward international business ventures [44].

According to a research work by [35] Machine translation was noted to faces range of intricate obstacles. Challenges include deciphering ambiguity, grappling with idiomatic phrases, and navigating cultural subtleties that impede precise translations. The task of managing low-resource languages with constrained training data necessitates inventive transfer learning strategies [35].

Sustaining context, particularly in lengthy texts, presents an additional complexity. Also, striving for equivalence in translation quality across language pairs is a considered a continual pursuit, which reflects the inherent variation in linguistic structures [35].

CHATBOT ALGORITHMIC CLASSIFICATION PROCESS

Algorithmic classification in chatbots involves utilizing algorithms and methodologies to categorize and classify user inputs or intents during a chatbot conversation. And this categorization as stated by [3] empowers the chatbot to grasp the user's intention or inquiry, enabling it to generate appropriate responses.

Intent Recognition

Algorithmic classification is often used to identify the user's intent, which is considered the underlying purpose of their message [27]. This process entails educating the chatbot using a dataset comprising instances of varied intents. Machine learning algorithms, including supervised learning or deep learning models, are harnessed to grasp patterns and connections within the data [27].

And this is considered to empower the chatbot to precisely categorize novel user inputs into predefined intent groups [25]. Intent recognition fosters seamless communication, transforming the way we interact with technology and techniques that it involves are;

i. Supervised learning: Utilize labeled training data, supervised learning methods train classifiers to recognize specific intents. Common algorithms include Support Vector Machines (SVM) which is effective for binary intent classification, Naive Bayes which is suitable for text classification tasks, and Neural Networks which capture intricate patterns in text [68].

ii. Transfer learning: It leverages knowledge from pre-trained models to enhance intent recognition in data-scarce scenarios. Popular pre-trained models include: Bidirectional Encoder Representations from Transformers (BERT) and GPT Generative Pre-trained Transformer (GPT) [18].

iii. Unsupervised Learning: The Unsupervised learning methods discover underlying patterns without labeled data. Techniques include; clustering which entails grouping similar user inputs to infer latent intents and Latent Dirichlet Allocation (LDA) which involves topic modeling for discovering hidden topics in text data [11].

*iv. Hybrid Approaches***:** This involves combining multiple methods which often yields improved results. Such as Supervised-Self Training which utilizes labeled and predicted data for training and Unsupervised-Transfer Learning which involves Pre-training on unsupervised data followed by fine-tuning with labeled data [15].

Training Data Preparation

High-quality training data is the cornerstone of building effective machine learning models [15]. Clean, well-structured, and representative data ensures that models learn relevant patterns and relationships [28]. Poorly prepared data can lead to biased, inaccurate, and unreliable models, hindering their real-world applicability [32].

Successful algorithmic classification requires a diverse and well-labeled training dataset which have been described earlier. Human annotators label examples of user messages with their corresponding intents, allowing the algorithm to learn from these labeled examples [56].

Algorithms often require input data to be represented as numerical features. Techniques like word embedding (e.g., Word2Vec, GloVe) convert text into vectors that algorithms can process [7].

Model Training and Optimization

Once the training data is prepared, the selected algorithm is trained using the labeled examples. Parameters are adjusted to optimize the model's performance. Techniques like hyperparameter tuning ensure the algorithm generalizes well to new data.

During a chatbot interaction, when a user sends a message, the chatbot applies the trained algorithm to classify the intent [14]. The algorithm computes a probability score for each intent, and the one with the highest score is chosen as the predicted intent.

i. Dialogue Management

Intent recognition is often coupled with dialogue management [67]. The chatbot maintains the context of the conversation and adapts responses based on the recognized intent.

ii. Continual Learning

Chatbots can improve their classification accuracy over time by incorporating user feedback [51]. Reinforcement learning techniques allow the chatbot to learn from user interactions and adjust its intent recognition models accordingly.

Reinforcement Learning (RL) constitutes a machine learning framework which is centered on instructing agents to navigate sequential choices within an environment, aiming to optimize the accumulation of rewards [51]. Through iterations of experimentation, RL methods empower chatbot to refine their decision-making prowess progressively.

DISCUSSION AND CONCLUSION

The comprehensive examination of algorithmic approaches, feature extraction techniques, and datasets employed in existing chatbot models has provided valuable insights into the many-sided landscape of conversational AI. The diverse algorithms, ranging from rule-based systems to machine learning and neural network-based models, showcase the adaptability of chatbots to different use cases and requirements.

The exploration of feature extraction techniques underscores the pivotal role of preprocessing in extracting meaningful information from raw data, enhancing the efficiency and accuracy of chatbot interactions. From traditional methods to more sophisticated embeddings and representation learning, the selection of appropriate features plays a critical role in shaping the chatbot's understanding and response generation capabilities.

Moreover, the scrutiny of datasets used in training and evaluation sheds light on the importance of diverse and representative data sources. The quality and diversity of data directly influence the chatbot's ability to handle a wide range of user queries and contexts. This understanding is vital in addressing biases and ensuring the ethical deployment of chatbots in real-world scenarios. As the field of conversational AI continues to evolve, the merge of algorithmic advancements, feature extraction innovations, and careful dataset curation will be pivotal in shaping the next generation of chatbot models.

REFERENCES

[1] Abbott, A., & Hrycak, A. (1990). Measuring resemblance in sequence data: An optimal matching analysis of musicians' careers. *American journal of sociology, 96*(1), 144 185.

[2] Abdulaziz, W., Ameen, M. M., & Ahmed, B. (2019). An overview of bag of words; importance implementation applications and challenges.

[3] Adam, M., Wessel, M., & Benlian, A. (2021). AI-based chatbots in customer service and their effects on user compliance. *Electronic Markets, 31*(2), 427-445.

[4] Aelani, K., & Gustaman, G. (2021). Chatbot for Information Service of New Student Admission Using Multinomial Naïve Bayes Classification and TF-IDF Weighting. In *2nd International Seminar of Science and Applied Technology (ISSAT 2021)* (pp. 115-122). Atlantis Press.

[5] Agarap, A. F. M. (2018). A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. In *Proceedings of the 2018 10th international conference on machine learning and computing* (pp. 26-30).

[6] Anastasiou, D., Ruge, A., Ion, R., Segărceanu, S., Suciu, G., Pedretti, O., & Afkari, H. (2022). A Machine Translation-Powered Chatbot for Public Administration. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation* (pp. 329-330).

[7] Ajay H., (2020). Word2Vec, GloVe, and FastText, Explained. https://towardsdatascience.com/word2vec-glove-and-fasttext-explained-215a5cd4c06f

[8] Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., & Bengio, Y. (2016). End to-end attention-based large vocabulary speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4945-4949). IEEE.

[9] Bhoir, S., Ghorpade, T., & Mane, V. (2017). Comparative analysis of different word embedding models. In *2017 International conference on advances in computing, communication and Control (ICAC3)* (pp. 1-4). IEEE.

[10] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics, 5*, 135-146.

[11] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research, 3*(Jan), 993-1022.

[12] Cao, W., Wang, X., Ming, Z., & Gao, J. (2018). A review on neural networks with random weights. *Neurocomputing, 275*, 278-287.

[13] Chowdhary, K., & Chowdhary, K. R. (2020). Natural language processing. *Fundamentals of artificial intelligence*, 603-649.

[14] Chen, Y., & Luo, Z. (2023). Pre-Trained Joint Model for Intent Classification and Slot Filling Semantic Feature Fusion. *Sensors, 23*(5), 2848.

[15] Chen, X., & Cardie, C. (2018). Efficient Supervised Self-training of Named Entity Recognizers with Arbitrary Lexicons. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2.*

[16] Chen, Q., Gong, Y., Lu, Y., & Tang, J. (2022). Classifying and measuring the service quality of AI chatbot in frontline service. *Journal of Business Research, 145*, 552-568.

[17] Cochez, M., Ristoski, P., Ponzetto, S. P., & Paulheim, H. (2017). Global RDF vector space embeddings. In *The Semantic Web–ISWC 2017: 16th International Semantic Web Conference*, Vienna, Austria, October 21–25, 2017, Proceedings, Part I 16 (pp. 190-207).

[18] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

[19] Dharma, E. M., Gaol, F. L., Warnars, H. L. H. S., & Soewito, B. (2022). The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (CNN) text classification. *J Theor Appl Inf Technol, 100*(2), 31.

[20] Eisenstein, J. (2019). Introduction to natural language processing. MIT press.

[21] Graves, A. (2012). *Supervised sequence labelling. In Supervised sequence labelling with recurrent neural networks* (pp. 5-13). Springer, berlin, Heidelberg.

[22] Harris, L., & Dennis, C. (2011). Engaging customers on Facebook: Challenges for e-retailers. *Journal of Consumer Behaviour, 10*(6), 338-346.

[23] Hassler, M., & Fliedl, G. (2006). Text preparation through extended tokenization. *WIT Transactions on Information and Communication Technologies, 37*.

[23] Hussain, S., Ameri Sianaki, O., & Ababneh, N. (2019). A survey on conversational agents/chatbots classification and design techniques. In *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019)* 33 (pp. 946-956). Springer International Publishing.

[24] Kecman, V. (2005). Support vector machines–an introduction. In *Support vector machines: theory and applications* (pp. 1-47). Berlin, Heidelberg: Springer Berlin Heidelberg.

[25] Kumar, P., Sharma, M., Rawat, S., & Choudhury, T. (2018). Designing and developing a chatbot using machine learning. In *2018 International Conference on System Modeling & Advancement in Research Trends (SMART)* (pp. 87-91). IEEE.

[26] Kumar, R., & Ali, M. M. (2020). A review on chatbot design and implementation techniques. *Int. J. Eng. Technol, 7*(11).

[27] Koniew, M. (2020). Classification of the User's Intent Detection in Ecommerce systems-Survey and Recommendations. *International Journal of Information Engineering & Electronic Business, 12*(6).

[28] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., & Zettlemoyer, L. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.

[29] Liu, C. Z., Sheng, Y. X., Wei, Z. Q., & Yang, Y. Q. (2018). Research of text classification based on improved TF-IDF algorithm. In *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)* (pp. 218-222). IEEE.

[30] Martinez, A. R. (2012). Part-of-speech tagging. *Wiley Interdisciplinary Reviews: Computational Statistics, 4(*1), 107-113.

[31] Mars, M. (2022). From word embeddings to pre-trained language models: A state-of-the-art walkthrough. *Applied Sciences, 12*(17), 8805.

[32] Mondal, A., Dey, M., Das, D., Nagpal, S., & Garda, K. (2018). Chatbot: An automated conversation system for the educational domain. In *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)* (pp. 1-5). IEEE.

[33] Pham, B. (2020). Parts of speech tagging: Rule-based.

[34] Jatnika, D., Bijaksana, M. A., & Suryani, A. A. (2019). Word2vec model analysis for semantic similarities in English words. *Procedia Computer Science, 157*, 160-167.

[35] Jinfang, Y (2023). Exploring the Advantages and Limitations of Machine Translation in the Performance of Construction Industry.

[36] Kim, M., Kwak, B. W., Kim, Y., Lee, H. I., Hwang, S. W., & Yeo, J. (2022). Dual Task Framework for Improving Persona-Grounded Dialogue Dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 36, No. 10, pp. 10912-10920).

[37] Kübler, S., McDonald, R., & Nivre, J. (2009). Dependency parsing. In *Dependency parsing* (pp. 11-20). Cham: Springer International Publishing.

[38] Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications, 82*(3), 3713-3744.

[39] Li, Y., & Yang, T. (2018). Word embedding for understanding natural language: a survey. *Guide to big data applications*, 83-104.

[40] Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering, 34*(1), 50-70.

[41] Liu, J., Symons, C., & Vatsavai, R. R. (2022). Persona-Based Conversational AI: State of the Art and Challenges. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 993-1001). IEEE.

[42] Mansouri, A., Affendey, L. S., & Mamat, A. (2008). Named entity recognition approaches. *International Journal of Computer Science and Network Security, 8*(2), 339-344.

[43] Makantasis, K., Doulamis, A., Doulamis, N., & Psychas, K. (2016). Deep learning based human behavior recognition in industrial workflows. In *2016 IEEE International conference on image processing (ICIP)* (pp. 1609-1613). IEEE.

[44] Mueller, A., Nicolai, G., McCarthy, A. D., Lewis, D., Wu, W., & Yarowsky, D. (2020). An analysis of massively multilingual neural machine translation for low-resource languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 3710-3718).

[45] Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association, 18*(5), 544-551.

[46] Onyenwe, I. E., Hepple, M., Chinedu, U., & Ezeani, I. (2019). Toward an effective Igbo part-of-speech tagger. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 18*(4), 1-26.

[47] Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

[48] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., & Ward, R. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 24*(4), 694-707.

[49] Peter, D., Alavi, A., Javadi, B., & Fernandes, S. L. (Eds.). (2020). *The Cognitive Approach in Cloud Computing and Internet of Things Technologies for Surveillance Tracking Systems*. Academic Press.

[50] Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation, 60*(5), 503-520.

[51] Ricciardelli, E., & Biswas, D. (2019). Self-improving chatbots based on reinforcement learning. In *4th Multidisciplinary Conference on Reinforcement Learning and Decision Making.*

[52] Satu, M. S., & Parvez, M. H. (2015). Review of Integrated Applications with AIML based chatbot. In *2015 International Conference on Computer and Information Engineering (ICCIE)* (pp. 87-90). IEEE.

[53] Song, H., Zhang, W. N., Cui, Y., Wang, D., & Liu, T. (2019). Exploiting persona information for diverse generation of conversational responses. arXiv preprint arXiv:1905.12188.

[54] Steigerwald, E., Ramírez-Castañeda, V., Brandt, D. Y., Báldi, A., Shapiro, J. T., Bowker, L., & Tarvin, R. D. (2022). Overcoming language barriers in academia: Machine translation tools and a vision for a multilingual future. *BioScience, 72*(10), 988-998.

[55] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems, 27.*

[56] Tebenkov, E., & Prokhorov, I. (2021). Machine learning algorithms for teaching AI chat bots. *Procedia Computer Science, 190,* 735-744.

[57] Toraman, C., Yilmaz, E. H., Şahinuç, F., & Ozcelik, O. (2023). Impact of tokenization on language models: An analysis for Turkish. *ACM Transactions on Asian and Low-Resource Language Information Processing, 22*(4), 1-21.

[58] Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. *Advances in neural information processing systems*, *26*.

[59] Wei, J., Kim, S., Jung, H., & Kim, Y. H. (2023). Leveraging large language models to power chatbots for collecting user self-reported data. arXiv preprint arXiv:2301.05843.

[60] Cameron, F. (2014). A Simple Markov Chain Chatbot. Retrieved from https://cameron.cf/posts/2014-04-13-Markov%20Chain%20Chatbot.html.

[61] Abdul-Kader, S. A., & Woods, J. C. (2015). Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, *6*(7).

[62] Yin, Z., Chang, K. H., & Zhang, R. (2017). Deepprobe: Information directed sequence understanding and chatbot design via recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 2131-2139).

[63] Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review, 53*(8), 5929-5955.

[64] Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., & Wei, J. (2022). Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416.

[65] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

[66] Shewalkar, A. (2019). Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. *Journal of Artificial Intelligence and Soft Computing Research*, *9*(4), 235-245.

[68] Lowe, R., Pow, N., Serban, I., & Pineau, J. (2015). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. arXiv preprint arXiv:1506.08909.

[68] Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. arXiv preprint.