Journal of
Information Systems Engineering
and Business Intelligence

Vol.11, No.1, February 2025
Available online at: http://e-journal.unair.ac.id/index.php/JISEBI

Multi-task Learning for Named Entity Recognition and Intent Classification in Natural Language Understanding Applications

Rizal Setya Perdana [1)*], Putra Pandu Adikara [2)]

[1)2)]*Department of Informatics Engineering, Universitas Brawijaya, Malang, Indonesia*
[1)]rizalespe@ub.ac.id, [2)]adikara.putra@ub.ac.id

Abstract

Background: Understanding human language is a part of the research in Natural Language Processing (NLP) known as Natural Language Understanding (NLU). It becomes a crucial part of some NLP applications such as chatbots, that interpret the user intent and important entities. NLU systems depend on intent classification and named entity recognition (NER) which is crucial for understanding the user input to extract meaningful information. Not only important in chatbots, NLU also provides a pivotal function in other applications for efficient and precise text understanding.
Objective: The aim of this study is to introduce multitask learning techniques to improve the application's performance on NLU tasks, especially intent classification and NER in specific domains.
Methods: To achieve the language understanding capability, a strategy is to combine the intent classification and entity recognition tasks by using a shared model based on the shared representation and task dependencies. This approach is known as multitask learning and leverages the collaborative interaction between these related tasks to enhance performance. The proposed learning architecture is designed to be adaptable to various NLU-based applications, but in this work are discussed use cases in chatbots.
Results: The results show the effectiveness of the proposed approach by following several experiments, both from intent classification and named entity recognitions. The multitask learning capabilities highlight the potential of multi-task learning in chatbot systems for close domains. The optimal hyperparameters consist of a warm-up step of 60, an early stopping probability of 10, a weight decay of 0.001, a Named Entity Recognition (NER) loss weight of 0.58, and an intention classification loss weight of 0.4.
Conclusion: The performance of Dual Intent and Entity Transformer (DIET) for both tasks—intent classification and named entity recognition—is highly dependent on the data. This leads to various capabilities for the hyperparameter combinations. Our proposed model architecture significantly outperforms previous studies based on common evaluation metrics.

Keywords: Natural Language Understanding, Chatbot, Multi-task Learning, Named Entity Recognition

Article history: Received 26 March 2024, first decision 16 July 2024, accepted 11 August 2024, available online 28 March 2025

I. INTRODUCTION

Intelligence systems on human language applications rely on Natural Language Understanding (NLU) to enable machines to understand context and information about human intention. NLU is important for various applications, including information retrieval systems, customer service automation, and virtual assistants, popularly known as chatbots. Chatbot systems are an example of use cases for NLU where accurate intent classification and named entity recognition (NER) are crucial for understanding user prompts and delivering relevant responses.

Significant development on close domain chatbots (CDC) has been proven by the emergence of various applications in several domains, such as customer support [1][2][3], healthcare [4][5][6], education [7][8][9], and finance [10][11][12], where domain specific conversation assistance is needed. CDC is a sub-system of conversational agents that operate with a specific domain or topic. Different from open domain chatbots, which aim to respond to questions in general topics or wide range topics, CDC has limited capability for understanding the language subject. The open domain chatbot utilizes extensive language models trained on large scale datasets, enabling them to understand various unspecified user topic input. There are three main components in building chatbot systems, i.e., natural language understanding (interpreter), dialog management, and natural language generation (response generator) [13][14]. NLU consists of several processes to interpret the meanings and context for user input [15]. Two main processes in NLU

* Corresponding author

ISSN 2443-2555 (online) 2598-6333 (print)
doi: http://dx.doi.org/10.20473/jisebi.11.1.1-16

include intent classification and NER. Perfect intent classification and entity recognition are important components of chatbots, enabling them to understand user context and extract relevant information. This research focuses on improving these mechanisms by leveraging a multitask learning approach, specifically targeting the tasks of intent classification and NER within a domain-specific context. The main objective of this study is to enhance the NLU capabilities of the chatbot system, which are fundamental to interpreting the user prompt or inputs that will guide appropriate responses. By incorporating these two tasks, i.e., intent classification and NER within a multitask learning architecture, this study aims to develop a model that not only improves the metrics but also outperforms to understand the context.

Both intent classification and NER are important in understanding user queries and extracting relevant information from human language. Intent classification [16] attempts to classify the category of the utterance of human language into set of predefined user intentions. While named entity recognition (NER) [17] has the objective to detect the named entity and determine the entity into predefined categories, e.g., person name, location name, and organization name. This study explores the use of multi-task learning strategy, i.e., combining intent classification and NER to improve the performance in a close domain chatbot. The objectives and significance of the study are outlined, setting the context for the subsequent sections.

According to a survey on chatbot evaluation [18], readability, relevance, consistency, informativeness, and naturalness are the five standard criteria for a chatbot system. In order to reach those criteria, especially for relevance, the chatbot needs to be good at connecting responses to context, particularly when it comes to topic connection. That is why interpretation must be done well at the early stage by doing named entity recognition and intent classification.

Multi-task Learning (MTL) [19] improves generalization by utilizing domain-specific information within the training signals of every related task within an architecture. Each task is trained in parallel using a shared representation of the data. The rise of generalization performance by practicing MTL has been proven for various models such as Unified Transformer (UniT) [20], backprop nets [21], and deep neural networks [22], [23]. This further encourages the idea that MTL has the potential to have a significant utility role in machine learning.

A decent amount of research has attempted to do Intent Classification and Named Entity Recognition as separate tasks. Deep learning methods have gained a lot of traction in those two fields [24]. LSTM [25], BiLSTM [26], and CNN [27] have been widely used to solve intent classification tasks with good results. A problem arises when an intent is quite dependent on the entity. Since these two tasks are not inherently independent [28], the model might misclassify the intent of a query because of a particular entity.

As one of the main tasks in Natural Language Processing (NLP), numerous studies have been conducted on NER in chatbot architectures using various methods ranging from SpaCy's entity recognizer [29], RNN [30], RASA NLU[30], to fine-tuning BERT [31]. These methods compute probabilities between each feature, thus performing better than models like Naive Bayes [32] which compute the probability of each feature independently. NER as a task faces several challenges such as consistency of data annotations, detecting synonyms, understanding informal text and unfamiliar entities [33].

Performing two different but related tasks separately, i.e., named entity recognition and intent classification for understanding the natural language in chatbot, will lead to a problem in the numerical representation of each learning model. Using the same dataset and a similar model for two tasks resulted in different representations in the featurization stage that affected the model generalization capability. In order to address this issue, a potential approach involves the integration of intent classification and named entity recognition (NER) through the utilization of a multi-task learning framework. In this mechanism, the training procedures allow two or more models to perform in parallel [19]. An instance of the utilization of the multi-task learning concept in chatbot development is the Dual Intent and Entity Transformer (DIET) model [34].

DIET is a multi-task transformer architecture that can handle intention classification and NER simultaneously. In contrast to other model language architectures, DIET demonstrates a modular architecture that offers enhanced flexibility and fast training capabilities. However, this research does not discuss the generalizability to different domains or languages, limiting its applicability in real-world conversational AI applications. The investigation of different configurations, including hyperparameter tuning and sequence analysis model architecture, remains unexplored for optimization and improvement. The absence of such explorations signifies a possible untapped potential for enhancing the performance of the models.

To deal with the drawbacks mentioned above, we formulate a learning strategy in multi-task learning, i.e., NER and intent classification that deal with non-English close domain chatbot and combine the two aspects of the human conversation as presented in Fig. 1. Hence, this study presents a comprehensive model that specifically targets the comprehension of natural language in a particular domain. A transfer learning strategy leverages knowledge from a vast linguistic corpus included in this research.

The primary contribution of this study involves the development of a novel architecture as presented in Fig. 2 that is designed to address the challenge of understanding human language input, specifically in the context of natural language understanding. The issue at hand pertains to the Indonesian Language, which falls under the category of low-resource languages. Consequently, the close domain chatbot must devise strategies to address this challenge. The originality of this research derives from its incorporation of multitasking learning and the utilization of a pre-trained language model simultaneously. This approach enables the development of a contextualized close domain chatbot within the constraints of limited language resources. Language user input will be transformed into word embedding representation using transformers architecture. Next, a series of entity labels are predicted through the Conditional Random Field (CRF) marker layer. The resulting tokenization and NER transformers are then embedded into a single semantic vector space to perform intent classification. Detailed steps will be explained in the Method section.

This study's primary contributions are described in the three-fold. First, we introduce an end-to-end architecture of multi-task learning strategy for combining the named entity recognition (NER) and intent classification task for close domain chatbot. Secondly, the utilization of a transfer learning technique involves the extraction and utilization of knowledge from an extensive linguistic corpus that has been incorporated within the scope of this research. Finally, this study provides a series of comprehensive experiments that have been investigated to validate the higher accuracy of the suggested model.

The present study is structured in the following manner. Section 2 encompassed a comprehensive examination of prior research on natural language comprehension in chatbots, multi-task learning, and transfer learning strategy in language model. Section 3 introduces the proposed model, with a thorough explanation of the constituents for each function. Section 4 presents a comprehensive elucidation of the conducted experiment. The findings of the experiment are presented in Section 5. Section 6 offers a comprehensive analysis of the results. Section 7 provides a thorough examination and elucidation of the summaries while also identifying possibilities for future research.

II. Literature Review

A. Chatbot's Natural Language Understanding (NLU)

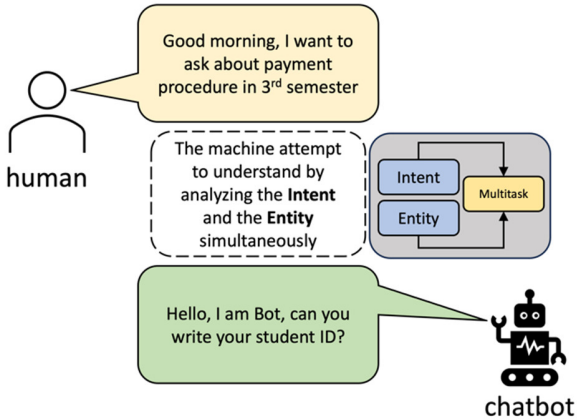


Fig. 1 An explanation of the functioning of chatbots and the overall procedure of language comprehension in a chatbot designed for a certain topic.

The message interpreter represents a vital component inside the foundational framework of a chatbot. The primary objective of an interpreter is to comprehend the underlying purpose of an incoming communication through the conversion of a natural human language into data that can be processed by a machine. The message interpreter is comprised of three key components, namely intent detection, domain identification, and entity extraction. The identification of domains is crucial when developing a chatbot that is not limited to a certain domain, such as an open domain chatbot. This study aims to develop a specialized chatbot designed for implementation within an academic enterprise. The Erasmus AI-chatbot [35] is an illustration of a chatbot system specifically developed to aid educational institutions through the utilization of university websites and curated scholarly articles. The intent extraction in the NEU-chatbot [36] is conducted using the supervised embeddings pipeline. Subsequently, the system proceeds to assign predefined intents to the incoming user messages. In the field of intent classification, researchers have utilized the pre-trained BERT model [37]. However, the NEU-chatbot system also integrates a pre-trained Dual Intent and Entity Transformer (DIET) [34], that alternative provides a more efficient and resource-efficient option compared to

BERT. The proposed approach in DIET attempts to incorporate intent classification and NER using multi-task learning within a close domain chatbot. In addition, this study involves the transfer learning strategy to extract and leverage knowledge from a larger model.

B. Multi-task Learning

Multi-task learning is a conceptual approach in machine learning that utilizes the collective knowledge from multiple training domains that the model interconnects with the objective to improve the overall performance of the model. An instance of the training combination of each model inside the architecture of this mechanism is conducted in [38]. It aims to enhance the efficiency and mitigate the issue of the overfitting inside the model. In machine learning, the overfitting problem refers to the failure of a model to consider all potential outcomes due to the complex models. A survey in [39] provides an overview of recent advancements in multi-task learning techniques within the field of Natural Language Processing (NLP). The aim of this review is to categorize these techniques into two main approaches depending on the degree of task relatedness: (i) joint training and (ii) multi-step training. The study presented a novel approach that extended upon the Model-Agnostic Meta-Learning (MAML) framework in the context of personalized dialogue learning [40]. This paper proposes a framework combining reinforcement learning and prompt-based learning to steer dialogue generation without accessing model parameters. The techniques encompass the extension of Model-Agnostic Meta-Learning (MAML) to facilitate tailored conversation learning, the utilization of MTL for conversation State Tracking, the integration of structure learning into chatbot systems, and the application of MTL to expedite convergence speed and improve the generalizability of the generator model on previously encountered tasks. Involving the external information, i.e., additional prompt, still lacks scalability; thus, our proposed architecture will use transfer learning in the language model.

C. Transfer Learning for Language Model

Transfer learning in the field of natural language processing (NLP) involves the utilization of existent models to improve performance in languages that have limited resources, as well as the adjustment of these models for different domains. [41]. Pretrained transformer models, such as BERT and GPT, provide notable benefits in natural language processing (NLP) models by facilitating the transfer of knowledge to downstream tasks and effectively mitigating the difficulties associated with training a model from its initial state. Research in [42] shows that transfer learning leverages in retrieval-based open-domain dialogue systems is used to re-rank retrieved responses by re-ranking retrieved responses on a large dataset from a different domain. The limitation of this research is dependent on the availability of the context provided. A study in [43] introduces a transfer learning method to improve the performance of goal-oriented chatbots in low data regimes. The transfer learning-based approach is proposed to mitigate the effects of low in-domain data availability, which improves the bot's success rate by 20% in relative terms for distant domains and more than double it for close domains, compared to the model without transfer learning. There are two main limitations in the prior work by [43], i.e., low availability of in-domain training data and slow training process without transfer learning. Thus, to address the limitations of the previous research, we propose to leverage the transfer learning approach, especially the use of the pre-trained model in multi-task learning.

III. METHODS

The proposed architecture is presented in Fig. 2. It presents the combining IndoBERT with additional layers for multitask learning, targeting both intent classification and named entity recognition (NER). The IndoBERT Embedding Layer serves to convert tokenized input text into embeddings, capturing contextual information. The Transformer Layers further refine these embeddings to enhance the understanding of the input text. The NER component utilizes a conditional random field (CRF) layer to identify and label the named entities within the human language input. Concurrently, the intent classification component employs the [CLS] token's embedding, passed through a dense layer to determine the user's intent. In multitask learning, shared dense layer support both NER and intent classification and combined losses to guide the model's optimization during training. Each task has a specific role in enhancing the NLU aspect by handling both tasks.

A. Dataset

The modeling process in this study utilizes a secondary dataset containing a compilation of frequently asked questions (FAQ) acquired from the internal data of the Information and Communication Directorate of Universitas Brawijaya (DTI UB). The proposed model uses the vocabulary list in the tokenization and featurization based on the IndoBERT module. Data preparation using the list of frequently asked questions dataset at the TIK UB Helpdesk includes several stages. The dataset was obtained from the Brawijaya University ICT Helpdesk online ticketing system application database. The context of the data used is a collection of questions that are often asked by the academic

community of Universitas Brawijaya who need help regarding IT services at Universitas Brawijaya. This text data is presented in Bahasa Indonesia. The dataset was then reviewed to select queries that could be used in the construction of chatbot according to the intentions used, such as asking about how to create a Brawijaya University email.

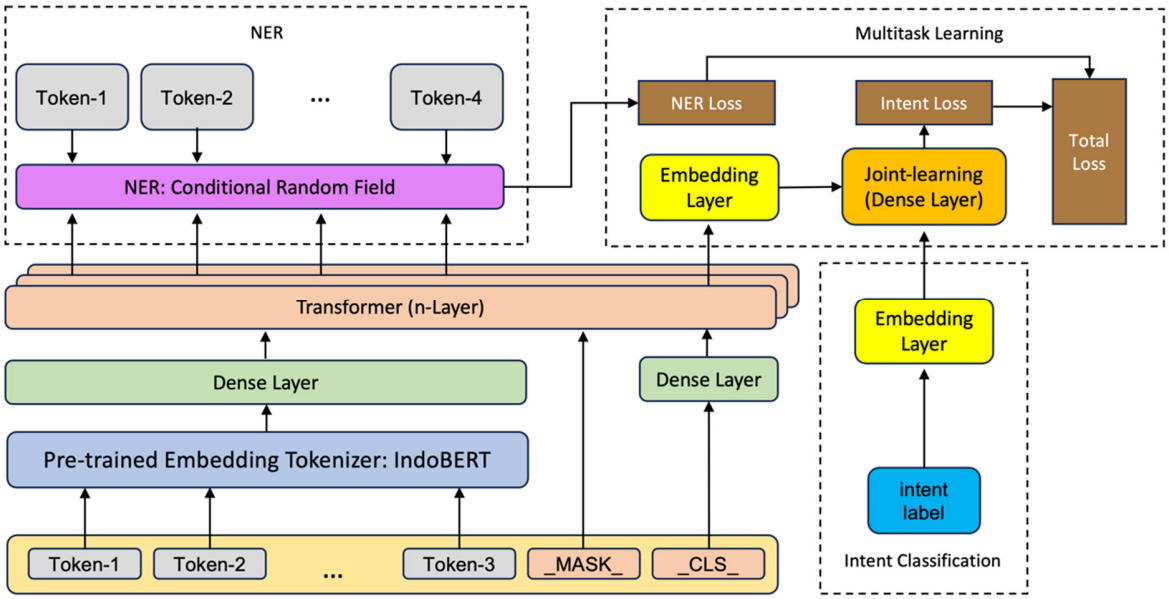


Fig. 2 The present work presents an explanation of the operational mechanisms of chatbots and the general procedure of language comprehension within a domain-specific chatbot.

The total record number observed from the database is 91977 conversations from March 2020 to December 2022. The column from the table is `id`, `title`, `body`, `pid`, `ip_address`, `thread_id`, `recipients`, `staff_id`, `extra`, `user_id`, `format`, `type`, `editor_type`, `flags`, `poster`, `editor`, `created`, and `updated`. The data was then analyzed according to the needs of this research. Based on the results of data analysis, especially on the contents of the BODY column which contains text from users, 7 intentions were determined which were used as the target class in this study. For each intention, about 100 data were taken that fell into that intention category.

To validate that the data taken falls into the category, we apply the following procedure. First, we determine the number of datasets we will take for each intention category. Then, we curate randomly and fill in the desired data number. Finally, we engage domain experts to review the curated data. The domain experts involved in this process include the person in charge as helpdesk assistant who replies to incoming questions, the moderator who acts as the manager in the organization, and the operational manager of the public relations officer in the Brawijaya University ICT Directorate. Specifically, four experts participated in the review: two helpdesk assistants, one moderator, and one operational manager. The helpdesk assistants have extensive experience in addressing and classifying incoming queries, while the moderator oversees the entire process to ensure consistency and accuracy. The operational manager provides strategic oversight and ensures alignment with organizational goals. The experts reviewed the questions, classified them, and provided answers to ensure the data's relevance and correctness.

The dataset used in this study consists of 19 columns in total. These columns include `pid`, `thread_id`, `staff_id`, `user_id`, `type`, `flags`, `poster`, `editor`, `editor_type`, `source`, `title`, `body`, `format`, `ip_address`, `extra`, `recipients`, `created`, and `updated`. However, for the purposes of this research, we focused exclusively on the `body` column. The `body` column contains the main content of the messages, which is crucial for the tasks of intent classification and named entity recognition. The following is the list of intentions used in this research is presented in the Table 1.

The predefined entities or categories for the NER task as the label for the user queries, including prefixes such as B- (begin) and I- (inside) to indicate the position of the entity within a text segment. The labels representing structured information from unstructured text are important for the chatbot's ability to understand the intention from the human language. The list of the NER is as follows, B-person, I-person, B-organization, I-organization, B-apps, I-apps, and O (other). An example for the labeled sentence "`Pengajuan Email Lembaga Tim Kepenasihatan Akademik S1 Kedokteran`" are labeled "`Email: B-apps, Lembaga: B-organization, Tim: I-organization, Kepenasihatan: I-organization, Akademik: I-organization, S1: I-organization, Kedokteran: I-organization`".

TABLE 1
INTENT CLASS DESCRIPTION

Intent Label	Description	Num. of Instance	Training set	Validation Set
`Gagal_login`	The message sender failed to log into an application or domain.	100	70	30
`Masalah_vpn`	The sender of the message has a VPN-related problem.	80	56	24
`Mengajukan_email`	The sender of the message intends to apply for a new email address	90	63	27
`Nilai_IT_tidak keluar`	The sender of the message questions the value of IT certification results that have not been listed in student portal.	120	84	36
`Permintaan_lisensi_office`	Message sender requests Microsoft Office 365 access	110	77	33
`Pengajuan_webhosting`	Message sender requests webhosting	95	66	29
`Ucapan_terimakasih`	The sender of the message only expressed their gratitude	130	91	39
	Total	725	507	218

The data labeling process was performed automatically using a Python script that takes a conversation document and a list of entities as input. Words within the document that represent entities are labeled according to the IOB (Inside, Outside, Beginning) format. If a word is the first part of an entity, it is tagged with B (Begin), followed by the corresponding entity category. Subsequent words that are part of the same entity are tagged with I (Inside) and the entity category. The summarize of the number of instances for the 7 NER labels is presented in Table 2.

TABLE 2
NUMBER OF INSTANCE FOR EACH NER LABEL

NER Label	Num. of Instance
`O (other)`	89250
`B-aplikasi`	2823
`I-aplikasi`	1198
`B-orang`	1096
`I-orang`	1679
`B-orang`	1009
`I-organisasi`	1959

1) Labels Used.

The dataset was labeled with specific intent categories that represent the primary goals or actions of the users' queries. `Gagal_login` (login failure), `Masalah_vpn` (VPN issues), `Mengajukan_email` (email application), `Nilai_IT_tidak_keluar` (IT certification issues), `Permintaan_lisensi_office` (Microsoft Office license requests), `Pengajuan_webhosting` (web hosting requests), and `Ucapan_terimakasih` (expressions of gratitude) are the intention labels. These labels are the common types of incoming inquiries received by the helpdesk. Once the helpdesk obtained sets of user inquiries, it assigned a category based on their content and performed classification and following process by chatbot.

2) Named Entity Labels

We annotate the dataset with named entity labels to distinguish the entities mentioned in the text input. The labels used include B-person (beginning of a person's name), I-person (inside a person's name), B-organization (beginning of an organization's name), I-organization (inside an organization's name), B-apps (beginning of an application name), I-apps (inside an application name), and O (other, indicating non-entity text). These labels follow the IOB (Inside, Outside, Beginning) format, which is a convention in named entity recognition tasks to indicate the boundaries and positions of entities within a text segment.

3) Annotation Team

The labeling process was performed by four people who have expertise in this domain. Two helpdesk assistants had experience in handling and categorizing the user queries based on the problem or inquiry. They have insight to solve common issues and identify the intent communicated by users. One moderator: this person ensures that the labeling is consistent across different annotators and gives guidance on resolving any problem. One operational manager: this individual from the ICT Directorate ensures that the labeling matches with the operational needs and the aims of the organization.

4) Inter-Annotator Agreement

The Inter-Annotator agreement was estimated using Cohen's kappa coefficient to ensure the reliability and consistency of the labeling process. It evaluates the level of agreement between two or more annotators. After an initial round of labeling on a sample of the dataset, the kappa coefficient was calculated 0.85, which indicates a high level of agreement among the annotators. A kappa score between 0.81 and 1.00 is generally considered as almost perfect agreement.

B. Embedding Representation and Transfer Learning

Our methodology employs multi-task learning, which involves the transformation of two data structures. Initially, the raw text will be tokenized and subsequently converted into numerical vectors, commonly known as word embeddings. To establish the contextual meaning of the sentence, the encoding procedure employs two levels of transformers that utilize weights from all layers. The proposed method adds a training objective to predict randomly masked input tokens. This masking process is added as a regulator and helps the model to learn general features of the text without discriminating specific features for classification.

We specifically utilize transfer learning by employing the Bidirectional Encoder Representations from Transformers (BERT) [37] model, which is a pre-trained model that uses transformers and is used in natural language processing models. The BERT architecture consists of multiple layers of transformers that utilize attention mechanisms to infer dependencies between words. During the pre-train phase, BERT captures context information in both directions by training the model to predict missing words based on the context of the surrounding words. Therefore, BERT can infer the numerical representation of words contextually.

This research uses a variation of the pre-trained BERT model trained using Indonesian vocabulary called IndoBERT [44]. It has an input size of 512 with 12 attention heads, 12 hidden layers with dimensions of 768, and a feed-forward hidden layer built on an uncased BERT base. This model was trained using more than 220 million words from numerous official sources, such as Wikipedia Indonesia, and focused only on one language.

The ability to transfer learning and adapt the learned model to new tasks was implemented. It uses the IndoBERT model to transfer knowledge and adapt the classification and NER tasks in the Indonesian language. Input and output formats are important for ensuring the model is effective for new tasks. Formatting the input into the expected format is performed by tokenizing the text using WordPiece tokenization to break down sentences into smaller tokens. A special purpose token is added, such as [CLS] that appears at the beginning of every sentence and [SEP] at the end of each sentence. Sentences were either padded or truncated to the same length of 512 tokens, using the [PAD] token. A masking strategy was performed where some tokens were randomly masked to make the model learn contextual representation effectively. The tokenized input was then transformed into a numerical representation, i.e., an embedding vector, using a pre-trained IndoBERT model. For the output format, the model's intent classification task used the embedding of the [CLS] token. The intent labels as the output were produced by a softmax layer to this embedding that calculates probability distribution with the highest probability indicating the predicted intent. For the NER task, the model outputs a sequence of labels corresponding to each token in the input sentence, following the IOB (Inside, Outside, Beginning) format. A conditional random field (CRF) layer was used to ensure that the predicted labels were consistent and accurate.

C. Feature Extraction

In text mining, the extracting feature using a pre-trained model is called vectorization. This process will transform the text into numerical vectors for the consecutive step. This research employed the IndoBERT model to obtain a representative embedding vector from the input text. Subsequently, a specialized token [CLS] was added to the start of the sentence, while [SEP] was appended to the end of the sentence in order to modify the array of tokens. The padding token, [PAD], was introduced to ensure that all arrays of tokens have uniform lengths, with the length being equal to the maximum length of the tokens. At last, the pre-trained model was used to convert each token into token IDs. This procedure guarantees that the tokens are appropriately calibrated for use in the pre-training model, which will produce predetermined embeddings with a dimensionality of 512. Subsequent learning processes will make use of these embeddings.

The transformer mechanism incorporates self-attention, which is a specific instance of multi-head attention (presented in Equation 1), where the keys K, inputs Q, and values V are determined by the same hidden layer. The inputs for this study include a query q_i, a set of keys $K = (k_{1,\dots,}k_j)$ and set of values $\mathrm{V} = (v_{1,\dots,}v_j)$ where j represents the number of elements. The dot-product attention is presented Equation 2.

$$MultiAtt(q, K, V) = W^o(head_1, \dots, head_N) \qquad (1)$$

$$Att(q, K, V) = softmax(\frac{q_i K^T}{\sqrt{d}})V \quad (2)$$

D. Intent Classification

Intention classification is one of the tasks in the main component of a chatbot, which is natural language understanding (NLU) [45]. Intention classification plays a role in understanding the topic of discussion when chatbots communicate with humans, so it also indirectly affects the determination of the correct answer in the conversation [46]. The intents labels y_{intent} are encoded into an embedded representation semantic vector space $h_{CLS} = E(a_{CLS})$ and $h_{Intent} = E(y_{Intent})$. The loss function for intent classification is defined in Equation 3.

$$L_{INTENT} = -\langle S_1^+ - \log\left(e^{S_1^+} + \sum_{\Omega_1^-} e^{S_1^-}\right)\rangle \quad (3)$$

In our intent identification process, the "default" label serves as a catch-all category for user inputs that do not match any of the predefined intent categories. The scope of the "default" label encompasses any ambiguous, irrelevant, or unrecognized queries that the chatbot encounters. For the intent classification task, the model was trained on all data points at once, across all intent categories. The dataset was split into training (70%) and validation (30%) sets. The model used the entire dataset in a single pass to learn from the distribution of intents, following a standard classification process. Each intent was treated as a distinct class within the classification task, and the model was optimized using the categorical cross-entropy loss function.

E. Named Entity Prediction: Conditional Random Field (CRF)

In this research, Conditional Random Field (CRF) tagging layer on top of the transformer output sequence to perform Named Entity Recognition (NER). The transformer output for the CLS token and the intent labels are embedded into a semantic vector space, and the similarity between the transformer output and the target label is maximized using the dot-product loss. The evaluation of NER is conducted using the micro-averaged F1 score. A true positive is defined as an item with a prediction span that precisely matches the gold span and its label match. In this research, the NER performance will be evaluated with and without pre-trained embeddings. The first step is to prepare the training data, which consists of annotated text with named entities labeled. The data is usually in the IOB format, which stands for Inside, Outside, and Beginning. Each word in the text is labeled as either B (beginning of a named entity), I (inside a named entity), or O (outside a named entity). The next step is to extract features from the training data. The features can be anything that helps the model predict the named entities label y_{entity}, such as the word itself, it's part of speech, its capitalization, and its position in the sentence. Once the features are extracted, the CRF model is trained on the training data. The model learns to assign probabilities to each label (B, I, or O) for each word in the text based on the extracted features. In this research, the loss value for NER task denotes by $L_{CRF(.)}$ that presents negative log-likelihood for a CRF as present in the Equation 4.

$$L_{ENTITY} = L_{CRF}(a, y_{entity}) \quad (4)$$

F. Multi-Task Learning

This process incorporates a shared encoder layer that captures the contextual information of the input text, followed by separate decoder layers for intent classification and entity recognition. This study investigates the utilization of multi-task learning for the purpose of performing two prevalent dialogue language understanding tasks, namely intent prediction, and entity prediction. This study builds on the shared knowledge across the tasks by conducting joint training, resulting in enhanced performance in predicting both intent and entity. There are several strategies used in multi-task learning for NLP i.e., joint learning, multi-step training, auxiliary tasks, transfer learning, and structure learning. In this research, the main strategy used is joint learning for the fusion of the NER and intent classification, followed by transfer learning for the use of a pre-trained IndoBERT model. The training process in multi-task performed by minimizing the total loss L_{total} as defined in Equation 5.

$$L_{total} = L_I + L_E \quad (5)$$

In this study, a multitask learning framework was implemented to handle two distinct tasks: intent classification and named entity recognition (NER). Both tasks share a common IndoBERT encoder, which is pre-trained on a large corpus of Indonesian text. The input text is tokenized and passed through IndoBERT, which generates contextual

embeddings. These embeddings serve as input for the two sub-models—one for intent classification and the other for NER.

For the intent classification task, the embedding of the [CLS] token, which summarizes the meaning of the entire input, is fed into a dense layer. The softmax activation function is employed to predict the class intent from the predefined categories. The NER sub-model leverages the IndoBERT to generate token embeddings. The embedded tokens are passed through a conditional random field (CRF) layer. It assigns labels by identifying the entities within the input text to each token based on the IOB (Inside, Outside, Beginning) format.

The training strategy of the multi-task is using an end-to-end approach in which both tasks are optimized simultaneously within the same training cycle. Total loss is a combination of the loss value from both tasks, i.e., intent classification and NER. Different error values are used for two different tasks; for intent classification loss is calculated using categorical cross-entropy, while the NER loss calculated using the loss of the CRF layer. The model can efficiently learn shared representations by combining these losses into a single objective function. The combined loss value is minimized through backpropagation that allows the model to generalize effectively across both tasks while maintaining task-specific outputs. This multitasks learning strategy facilitates the model to leverage common information between two tasks.

G. Training and Validation

Trainer was utilized for the multitask learning process that was provided by Hugging Face Transformers to produce the final model. The multitask model is designated to handle both intent classification and NER tasks simultaneously. The IndoBERT encodes the input data, while separate output layers handle the intent classification and NER tasks. The combination of the intent classification loss (categorical cross-entropy) and the NER loss (CRF-based) into the total loss allows for joint optimization during training.

The Trainer helps to automate both the forward and backward propagation during the training process. The neural network architecture attempts to minimize the total loss by updating its parameters through backpropagation and leveraging the combined loss from both tasks. The multi-task learning setup enables the model to learn from both tasks simultaneously and improves the ability to generalize across different types of tasks.

The implementation of the proposed architecture is using PyTorch, a deep learning framework that is compatible with GPU-accelerated computation. More detailed components of the model architecture are presented in Table 3. The training was performed on an NVIDA DGX A100 GPU server. The Adam optimizer was used to update the model weights, with an initial learning rate of 1e−3. A linear decay learning rate schedule was employed, with a final learning rate of 1e−5, along with a warm-up technique. The model was trained with a mini-batch size of 64, and the training process was repeated for multiple epochs until the early-stopping criteria were met.

It is important to clarify that the data preprocessing and training stages are distinct. The preprocessing phase involves data extraction, transformation, tokenization, and loading. During this stage, the text data is first tokenized and transformed into a format suitable for input into the model. This step is carried out on the CPU and includes tasks such as splitting the data into training and validation sets, applying tokenization, and converting the raw text into input embeddings.

Once preprocessing is completed, the training stage begins, where the actual model training occurs on the GPU. During training, the pre-processed data is p. assed through the shared IndoBERT encoder and the task-specific output layers for both intent classification and NER. The average training time per iteration is 29.56 milliseconds, and the total training time for the entire model is approximately 290 seconds. The time required for training may vary depending on the values of the hyperparameters.

TABLE 3
THE COMPONENTS AND DIMENSIONS OF THE BUILDING BLOCK FOR THE CONFIGURATION OF A DEEP NEURAL NETWORK IN CONSTRUCTING THE MODEL ARCHITECTURE.

Block	Component Description
Input Layer	T-length word
Dense Layer Linear layer	Output: 512
Embedding Layer Word-level	Output: 512
Transformers	A transformer with two blocks and four multi-head attention mechanisms with a dimension of 256.

H. Evaluation Metrics

To evaluate the performance of the proposed model, we employ several metrics including loss and F1-score. The loss function utilized includes a combination of loss for both Named Entity Recognition (NER) and Intent

Classification, calculated as described in equations X and Y. To evaluate the model precisely, this research uses the F1-score, the representation of precision and recall value, used to evaluate both NER and intent classification performance. It is crucial to balance the trade-off between precision and recall, particularly in imbalanced datasets. Previous research in multitask learning used various evaluation methods to assess the effectiveness of the architecture. A multi-class confusion matrix is one of the assessment methods that provides a comparison of the actual class and the predicted class in binary classification. There are four main elements used for evaluating the model, i.e., true negative (TN), true positive (TP), false positive (FP), and false negative (FN). In a multi-class classification confusion matrix, the four elements of the confusion matrix are calculated over all classes. Precision refers to the probability that a model accurately predicts a specific class among other predictions made for that class. Equation 6 used to calculate precision refers to the probability that a model accurately predicts a specific class among other predictions made for that class.

$$Precision = \frac{TP}{FP+TP} \tag{6}$$

Recall is the number of times a correct prediction for a class occurs among all the data in that class. This value is used to infer the effectiveness of a model to predict that class. Equation 7 used to calculate recall is presented as follows:

$$Recall = \frac{TP}{FN+TP} \tag{7}$$

The F1-score can be defined as the harmonic mean of precision and recall as shown in Equation 8. The aforementioned sentence explains the correspondence between the class observed in the dataset and the class predicted by the model's prediction.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

IV. Results

In this study, a series of experiments is conducted with the objective of simulating an expected result. The experiment that has been done is to address the question of how the proposed architecture blocks can be optimized in order to achieve a contextualized understanding of human language. Furthermore, the experiment seeks to provide evidence of this optimization through the use of several evaluation mechanisms.

A. Hyperparameter Tuning

The experimental scenarios conducted have produced several results. In this section, we will present the results of several hyperparameters used in the training phases. The hyperparameter that evaluate in this research such as early stopping patience, warm-up step, weight decay, and weight loss for NER and intent classification loss value.

Warming-up step evaluation was carried out using 3 values, such as 65, 70 and 75. The warm-up step value was chosen based on 10% of the total steps in model training using the hyperparameter of training set ratio of 0.95, maximum epoch of 100, batch size of 8, weight decay of 0.01, early stopping patience of 15, early stopping threshold of 0.0001, NER loss weight of 0.1 and intention classification loss weight of 0.9. The evaluation result in Table 4 showed that models with warm-up step values of 65 and 75 can classify intentions and NER. These two hyperparameter settings do not provide a F1-score as high as those with a warm-up step of 70. A relatively little change in F1-score is produced by hyperparameters 70 and 75. The warm-up step value for this test depends on the number of steps the model takes during training, which depends on the quantity of data used. With the highest F1-score, warm-up step hyperparameter 70 was chosen.

TABLE 4
WARM-UP STEP VARIATION FOR HYPERPARAMETER TUNING EXPERIMENT

Warm-up Step	Early stopping patience	Weight loss NER	Weight loss intent	Weight decay	F1-score intent	F1-score NER
65	15	0.1	0.9	0.01	0.938	0.981
70	15	0.1	0.9	0.01	0.947	0.984
75	15	0.1	0.9	0.01	0.947	0.982

Early stopping patience hyperparameter tuning evaluation was carried out using 3 values, i.e., 5, 10 and 15. Hyperparameter tuning was carried out using the maximum epoch of 100, batch size of 8, weight decay of 0.01, warm-up step of 70 (consider the best value from previous experiments), early stopping threshold of 0.0001, NER loss weight

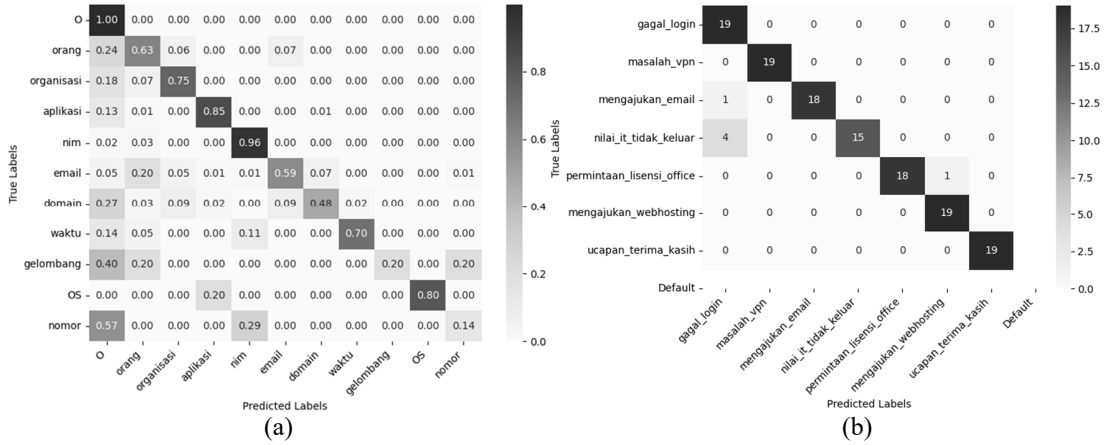
of 0.1 and intention classification loss weight of 0.9. The application of the early stopping hyperparameter idea aims to halt the training process before to the occurrence of overfitting. Increasing the value of early stopping patience typically leads to a higher number of epochs and a longer training time for the model to reach convergence. Results from the experimentation conducted on early stopping patience, as presented in Table 5, indicate a positive correlation between the value of early stopping and the model's proficiency in conducting Named Entity Recognition (NER) and intention categorization. An additional experiment was then carried out utilizing early stopping patience ratings of 18 and 20. The F1-score of the model, when considering these two values, indicates that the combination with an early starting patience value of 15 remains the most appropriate model capability.

TABLE 5
EARLY STOPPING PATIENCE VARIATION FOR HYPERPARAMETER TUNING EXPERIMENT

Warm-up Step	Early stopping patience	Weight loss NER	Weight loss intent	Weight decay	F1-score intent	F1-score NER
70	5	0.1	0.9	0.01	0.098	0.981
70	10	0.1	0.9	0.01	0.944	0.983
70	15	0.1	0.9	0.01	0.947	0.947

Weight decay hyperparameter tuning was carried out using 3 values, namely 0.1, 0.01 and 0.0001. Weight decay is used to change the weights used in the learning rate model at each epoch as a form of regularization. With weight decay, the model can use smaller weights and improve the generalization of the model. Based on the test results in Table 6, it was found that the model with a weight decay value of 0.1 could not perform NER or intention classification. Through these results, it can be concluded that regularization using a weight decay value of 0.1 can be considered too strong for the data used today, resulting in an underfitting model.

Weight loss NER and intent classification evaluation calculate the total loss of the model was carried out using 11 combinations of values in the range of 0 to 1 for both weights. Experiments conducted using hyperparameters of maximum epoch of 100, batch size of 8, warm-up step of 70, early stopping patience of 15, weight decay of 0.01, and early stopping threshold of 0.0001. With the macro average score of the NER f1-score and the highest classification, the best combination was selected, the NER loss weight of 0.6 and the intention classification loss weight of 0.4 was selected for the best combination. The macro average value was chosen because there is a combination of loss weights that only depend on the NER loss, as well as the intention classification weight loss.



(a) (b)

Fig. 3 Confusion matrix of the (a) entity recognition task and (b) intent classification task results.

V. DISCUSSION

The optimal hyperparameter combination, as determined by the conducted tests, yields a f1-score NER value of 0.9330 and an intention classification f1-score value of 0.9621. This combination is achieved by employing an early stopping patience of 10, a warm-up step of 60, a weight decay of 0.001, an intention classification loss weight of 0.42, and a NER loss weight of 0.5. This hyperparameter combination produces a model that goes through a training process of 17 epochs using a dynamic learning rate that changes at each epoch until it reaches a value of 4.19E-05. The proposed model can be used for simultaneous implementation of intention and NER classification, as evidenced by

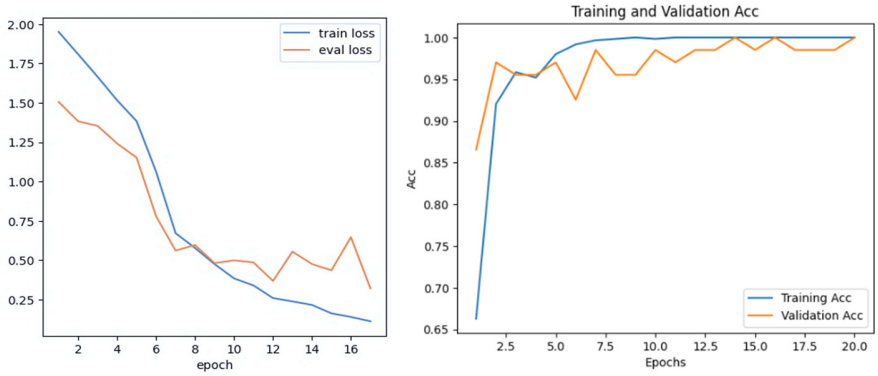
the confusion matrix generated by the model in Fig. 3a and Fig. 3b The model can perform intention classification well, with errors that can be caused by the data context of one intention being the same as another intention such as self-introduction. In contrast to the intention classification confusion matrix, the NER confusion matrix shows a more dispersed prediction capability and a bias towards non-entity words.

In the hyperparameter tuning test, the validation loss value was chosen as the reference for the best performance of the model. This selection is based on data limitations and simple model architecture, so it is prone to overfitting of training data. This test was conducted 100 times. From the 100 trials, one hyperparameter combination was obtained that produced the lowest validation loss value listed in Table 7.

TABLE 6
WEIGHT DECAY VARIATION FOR HYPERPARAMETER TUNING EXPERIMENT

Warm-up Step	Early stopping patience	Weight loss NER	Weight loss intent	Weight decay	F1-score intent	F1-score NER
70	15	0.1	0.9	0.1	0.923	0.989
70	15	0.1	0.9	0.01	0.947	0.984
70	15	0.1	0.9	0.001	0.981	0.981

The results of the training using the three different embeddings can be seen in TABLE 7. Testing was assessed based on the accuracy Fig. 4b and loss of the training and test data Fig. 4a. Based on the results of testing the three embeddings on model performance, which can be seen in TABLE 8, the 'bert-multilingual-base-cased' embedding produced the best model performance after training for 20 epochs. The 'bert-multilingual-base-cased' embedding successfully outperformed the other two embeddings from all test metric values, namely accuracy and loss in training and testing. This means that the 'bert-multilingual-base-cased' embedding can produce a better contextual representation of the data, compared to the other two embeddings. Therefore, the 'bert-multilingual-base-cased' embedding will be used in future tests.



(a) (b)

Fig. 4 (a) Loss value during training and evaluation phases and (b) accuracy during the hyperparameter tuning

TABLE 7
HYPERPARAMETERS CONFIGURATION IN EXPERIMENT

Hyperparameter	Initial Value	Best Combination
Optimizer	Adam	Adam
Learning rate	0.007	0.016
Weight decay	0.00017	0.0001
Batch size	32	32
Hidden size	64	96

TABLE 8
EMBEDDING TESTING RESULTS

Pre-trained Embedding	*Train acc*	*Train loss*	*Val acc*	*Val loss*
bert-base-uncased	0.97	0.07	0.93	0.15
bert-multilingual-base-cased	1.00	0.06	0.97	0.08
indolem/indobert-base-uncased	1.00	0.06	0.96	0.15

The model initially uses a combination of intention classification of 0.9 and NER loss weight of 0.1, so that loss intention plays a greater role in calculating total loss. For further analysis, all possible combinations between 0 and 1 were tested sequentially. The NER f1-score and intent classification using this model were then compared using macro averages to select the best combination.

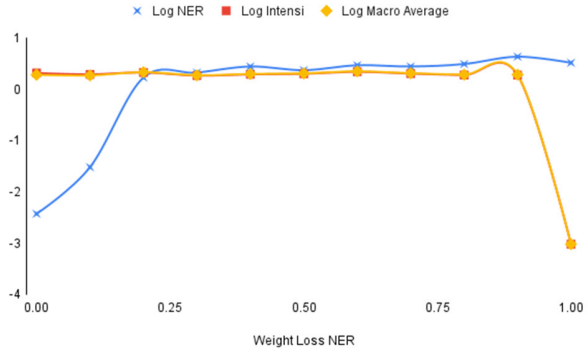


Fig. 5 The utilization of logarithmic representation in relationship between NER and purpose categorization

Based on the test results in Fig. 5, the increase in NER loss weight is directly proportional to the f1-score NER value of the model, and vice versa. Therefore, the model with a combination of NER loss weight and intention classification of 5 produces f1-score NER and intention classification with the least difference. The model that incorporates a weight of 0 for NER loss and a weight of 1 for intention classification loss only depends on intention classification loss for calculating the overall loss. However, this approach still yields a high f1-score value for the NER process. This is because the majority of the training data used is not part of the list of named entities and the model with this combination of weight values can only produce predictions of sub words that are not entities. Models that incorporate a weight of 1 for NER loss and 0 for intention classification loss are solely reliant on NER loss, hence limiting their ability to perform intention classification. The model predicts the default intentions for each test data used.

TABLE 9
COMPARISON OF THE BASELINE MODEL PERFORMANCE WITH THE PROPOSED MODEL

Model Architecture	*F1-Score*	*Precision*	*Recall*
Fine-tuned BERT [47]	85.73	84.71	86.78
ConveRT [48]	86.04	86.13	85.95
DIET [49]	88.98	87.17	84.30
Proposed Architecture	90.37	89.98	85.89

In order to demonstrate the validity of our proposed architecture, we conduct a comparative analysis of standard evaluation metrics with previous study, perform as the foundational framework for this study. The experiment was reproduced in order to utilize the dataset that was acquired, so enhancing the fairness of this comparison. We employed three evaluation metrics, i.e., F1-score, precision, and recall. The proposed architecture distinguishes itself from the other models mentioned in Table 9 by specifically addressing the challenges of understanding human language input, particularly in the context of the Indonesian language, which is considered a low-resource language. While the Fine-tuned BERT model leverages pre-trained knowledge, it may not fully adapt to domain-specific nuances. Model ConveRT [48] develops for conversational tasks that outperform the context but lack broader contextual understanding. DIET aims to balance intent classification and named entity recognition, but it has difficulty for low-resource language settings and specific domains. Our proposed architecture integrates a multi-task learning strategy with a pre-trained language model tailored for Indonesian. It combines intent classification and NER using transformers to ensure efficient handling of low-resource languages. The proposed design enables it to achieve superior performance, as demonstrated by the F1-score, precision, and recall metrics demonstrated in the target domain.

There are several limitations of the proposed architecture, although it demonstrated significant improvements in intent classification and NER for Indonesian language. First, the confidence in the IndoBERT pre-trained model limits its ability to capture the nuance of different dialects and informal languages. Secondly, linear layers for classification might not be an optimal choice compared with recurrent neural networks, which could potentially offer better

performance. Lastly, the dataset might not cover all possible variations that represent real-word applications, so it needs larger and more diverse datasets.

VI. CONCLUSIONS

This research objective is to utilize multi-task learning and transfer learning techniques to enhance intent classification and NER, particularly for Indonesian. The main objective was to demonstrate the effectiveness of integrating IndoBERT to improve the contextual understanding of Indonesian text. The proposed model utilizes the PyTorch and Transformer libraries. Our comprehensive testing and hyperparameter optimization led to identifying optimal settings, i.e., early stopping patience at 10, warm-up steps at 60, NER loss at 0.58, weight decay at 0.001, and intent classification loss at 0.42. These parameters generate an NER F1-score of 0.9330 and an intention classification f1-score of 0.9621. Although the proposed approach obtains superior performance, there is some bias from non-entity sub words, with our approach achieving an F1-score of 90.37, precision of 89.98, and recall of 85.89. This result outperforms traditional models in handling low-resource language, i.e., Indonesian. In summary, the objective to develop a contextual and optimized model for Indonesian was successfully met. It attempts to enhance the NLU component of chatbots through intent classification and NER; future development will explore how this improvement can be integrated into a fully functional chatbot system. Finally, this study establishes the foundation by focusing on the NLU components; the next phase of development will integrate these components into a full conversational flow, including the response generation from the chatbot.

Author Contributions: *Rizal Setya Perdana*: Conceptualization, Funding acquisition, Investigation, Methodology, Writing – original draft, Supervision, Validation. *Putra Pandu Adikara*: Data curation, Formal Analysis, Project administration, Resources, Software, Writing – review & editing, Visualization.

Funding: This work was supported by Universitas Brawijaya under Hibah Penelitian Pemula (HPP) Grand number 611.37/UN10.C200/2023.

Acknowledgments: This work is supported by Artificial Intelligence Center (AI-Center) Universitas Brawijaya https://aicenter.ub.ac.id/.

Conflicts of Interest: The authors declare no conflict of interest.

Data Availability: Dataset link will be available upon request

Informed Consent: There were no human subjects.

Animal Subjects: There were no animal subjects.

ORCID:
Rizal Setya Perdana: https://orcid.org/0000-0001-5420-1252
Putra Pandu Adikara: https://orcid.org/0000-0001-8849-1833

REFERENCES

[1] A. Agarwal, S. Maiya, and S. Aggarwal, "Evaluating Empathetic Chatbots in Customer Service Settings," *arXiv e-prints*, p. arXiv:2101.01334, Jan. 2021, doi: 10.48550/arXiv.2101.01334.

[2] E. W. T. Ngai, M. C. M. Lee, M. Luo, P. S. L. Chan, and T. Liang, "An intelligent knowledge-based chatbot for customer service," *Electron Commer Res Appl*, vol. 50, p. 101098, 2021, doi: https://doi.org/10.1016/j.elerap.2021.101098.

[3] L. Nicolescu and M. T. Tudorache, "Human-Computer Interaction in Customer Service: The Experience with AI Chatbots—A Systematic Literature Review," *Electronics (Basel)*, vol. 11, no. 10, 2022, doi: 10.3390/electronics11101579.

[4] L. Athota, V. K. Shukla, N. Pandey, and A. Rana, "Chatbot for Healthcare System Using Artificial Intelligence," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020, pp. 619–622. doi: 10.1109/ICRITO48877.2020.9197833.

[5] L. Xu, L. Sanders, K. Li, and J. C. L. Chow, "Chatbot for health care and oncology applications using artificial intelligence and machine learning: Systematic review," *JMIR Cancer*, vol. 7, no. 4, p. e27850, Nov. 2021.

[6] P. Kandpal, K. Jasnani, R. Raut, and S. Bhorge, "Contextual Chatbot for Healthcare Purposes (using Deep Learning)," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020, pp. 625–634. doi: 10.1109/WorldS450073.2020.9210351.

[7] M. Mateos-Sanchez, A. C. Melo, L. S. Blanco, and A. M. F. García, "Chatbot, as Educational and Inclusive Tool for People with Intellectual Disabilities," *Sustainability*, vol. 14, no. 3, 2022, doi: 10.3390/su14031520.
[8] A. S. Sreelakshmi, S. B. Abhinaya, A. Nair, and S. Jaya Nirmala, "A Question Answering and Quiz Generation Chatbot for Education," in *2019 Grace Hopper Celebration India (GHCI)*, 2019, pp. 1–6. doi: 10.1109/GHCI47972.2019.9071832.
[9] X. Deng and Z. Yu, "A Meta-Analysis and Systematic Review of the Effect of Chatbot Technology Use in Sustainable Education," *Sustainability*, vol. 15, no. 4, 2023, doi: 10.3390/su15042940.
[10] J.-W. Chang, N. Yen, and J. C. Hung, "Design of a NLP-empowered finance fraud awareness model: the anti-fraud chatbot for fraud detection and fraud classification as an instance," *J Ambient Intell Humaniz Comput*, vol. 13, no. 10, pp. 4663–4679, 2022, doi: 10.1007/s12652-021-03512-2.
[11] M. Ridha and K. Haura Maharani, "Implementation of Artificial Intelligence Chatbot in Optimizing Customer Service in Financial Technology Company PT. FinAccel Finance Indonesia," *Proc West Mark Ed Assoc Conf*, vol. 83, no. 1, 2022, doi: 10.3390/proceedings2022083021.
[12] D. Fotheringham and M. A. Wiles, "The effect of implementing chatbot customer service on stock returns: an event study analysis," *J Acad Mark Sci*, vol. 51, no. 4, pp. 802–822, 2023, doi: 10.1007/s11747-022-00841-2.
[13] R. S. Perdana, P. P. Adikara, Indriati, and D. Kurnianingtyas, "Knowledge-Enriched Domain Specific Chatbot on Low-resource Language," in *2022 11th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, IEEE, Aug. 2022, pp. 310–315. doi: 10.1109/EECCIS54468.2022.9902930.
[14] B. Galitsky, "Chatbot Components and Architectures," in *Developing Enterprise Chatbots*, Cham: Springer International Publishing, 2019, pp. 13–51. doi: 10.1007/978-3-030-04299-8_2.
[15] E. Adamopoulou and L. Moussiades, "An Overview of Chatbot Technology," in *AIAI 2020: Artificial Intelligence Applications and Innovations*, 2020, pp. 373–383. doi: 10.1007/978-3-030-49186-4_31.
[16] W. M. A. F. W. Hamzah, M. K. Yusof, I. Ismail, M. Makhtar, H. Nawang, and A. A. Aziz, "Multiclass Intent Classification for Chatbot Based on Machine Learning Algorithm," in *2022 Seventh International Conference on Informatics and Computing (ICIC)*, 2022, pp. 1–6. doi: 10.1109/ICIC56845.2022.10006979.
[17] I. Jauregi Unanue, E. Zare Borzeshi, and M. Piccardi, "Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition," *J Biomed Inform*, vol. 76, pp. 102–109, 2017, doi: https://doi.org/10.1016/j.jbi.2017.11.007.
[18] H. Liang and H. Li, "Towards Standard Criteria for human evaluation of Chatbots: A Survey," May 2021.
[19] R. Caruana, "Multitask Learning," *Mach Learn*, vol. 28, no. 1, pp. 41–75, 1997, doi: 10.1023/A:1007379606734.
[20] R. Hu and A. Singh, "UniT: Multimodal Multitask Learning with a Unified Transformer," Feb. 2021.
[21] L. Saitta, European Coordinating Committee for Artificial Intelligence., and Associazione italiana per l'intelligenza artificiale., *Machine learning : proceedings of the Thirteenth International Conference (ICML '96)*. Morgan Kaufmann Publishers, 1996. Accessed: Jul. 27, 2023. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3091696.3091708
[22] R. Collobert and J. Weston, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," in *Proceedings of the 25th international conference on Machine learning - ICML '08*, New York, New York, USA: ACM Press, 2008, pp. 160–167. doi: 10.1145/1390156.1390177.
[23] S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," Jun. 2017, Accessed: Jul. 27, 2023. [Online]. Available: https://arxiv.org/abs/1706.05098
[24] S. Rizou, A. Paflioti, A. Theofilatos, A. Vakali, G. Sarigiannidis, and K. Ch. Chatzisavvas, "Multilingual Name Entity Recognition and Intent Classification employing Deep Learning architectures," *Simul Model Pract Theory*, vol. 120, p. 102620, Nov. 2022, doi: 10.1016/j.simpat.2022.102620.
[25] G. Di Gennaro, A. Buonanno, A. Di Girolamo, A. Ospedale, and F. A. N. Palmieri, "Intent Classification in Question-Answering Using LSTM Architectures," in *Progresses in Artificial Intelligence and Neural Systems*, 2021, pp. 115–124. doi: 10.1007/978-981-15-5093-5_11.
[26] C. O. Bilah, T. B. Adji, and N. A. Setiawan, "Intent Detection on Indonesian Text Using Convolutional Neural Network," in *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, IEEE, Jun. 2022, pp. 174–178. doi: 10.1109/CyberneticsCom55287.2022.9865291.
[27] H. B. Hashemi, A. Asiaee, and R. Kraft, "Query Intent Detection using Convolutional Neural Networks," in *Proc. Int. Conf. Web Search Data Mining, Workshop Query Understanding*, 2016. doi: 10.1145/1235.
[28] A. Benayas, R. Hashempour, D. Rumble, S. Jameel, and R. C. De Amorim, "Unified Transformer Multi-Task Learning for Intent Classification With Entity Recognition," *IEEE Access*, vol. 9, pp. 147306–147314, 2021, doi: 10.1109/ACCESS.2021.3124268.
[29] S. Surana, J. Chekkala, and P. Bihani, "Chatbot based Crime Registration and Crime Awareness System using a custom Named Entity Recognition Model for Extracting Information from Complaints," *International Research Journal of Engineering and Technology*, 2021, [Online]. Available: www.irjet.net
[30] N. Ali, "Chatbot: A Conversational Agent employed with Named Entity Recognition Model using Artificial Neural Network," Jun. 2020, Accessed: Jul. 28, 2023. [Online]. Available: https://arxiv.org/abs/2007.04248
[31] T. Bauer, E. Devrim, M. Glazunov, W. L. Jaramillo, B. Mohan, and G. Spanakis, "#MeTooMaastricht: Building a Chatbot to Assist Survivors of Sexual Harassment," Springer, Cham, 2020, pp. 503–521. doi: 10.1007/978-3-030-43823-4_41.
[32] D. Christianto, E. Siswanto, and R. Chaniago, "Penggunaan Named Entity Recognition dan Artificial Intelligence Markup Language untuk Penerapan Chatbot Berbasis Teks," *Jurnal Telematika*, vol. 10, no. 2, p. 8, 2015, Accessed: Jul. 28, 2023. [Online]. Available: https://journal.ithb.ac.id/telematika/article/view/130
[33] J. Li, A. Sun, J. Han, and C. Li, "A Survey on Deep Learning for Named Entity Recognition," *IEEE Trans Knowl Data Eng*, vol. 34, no. 1, pp. 50–70, Jan. 2022, doi: 10.1109/TKDE.2020.2981314.
[34] T. Bunk, D. Varshneya, V. Vlasov, and A. Nichol, "DIET: Lightweight Language Understanding for Dialogue Systems," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.09936
[35] J. Thakkar, P. Raut, Y. Doshi, and K. Parekh, "Erasmus-AI Chatbot," 2018, [Online]. Available: www.ijcseonline.org
[36] T. T. Nguyen, A. D. Le, H. T. Hoang, and T. Nguyen, "NEU-chatbot: Chatbot for admission of National Economics University," *Computers and Education: Artificial Intelligence*, vol. 2, Jan. 2021, doi: 10.1016/j.caeai.2021.100036.
[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.04805
[38] R. Caruana, "Multitask Learning," *Mach Learn*, vol. 28, no. 1, pp. 41–75, 1997, doi: 10.1023/A:1007379606734.

[39] Z. Zhang, W. Yu, M. Yu, Z. Guo, and M. Jiang, "A Survey of Multi-task Learning in Natural Language Processing: Regarding Task Relatedness and Training Methods," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2023, pp. 943–956. doi: 10.18653/v1/2023.eacl-main.66.
[40] H. Su *et al.*, "Learning to Generate Prompts for Dialogue Generation through Reinforcement Learning," 2022.
[41] J. Bharadiya, "Transfer Learning in Natural Language Processing (NLP)," *European Journal of Technology*, vol. 7, no. 2, pp. 26–35, Jun. 2023, doi: 10.47672/ejt.1490.
[42] I. T. Aksu, N. F. Chen, L. F. D'Haro, and R. E. Banchs, "Reranking of Responses Using Transfer Learning for a Retrieval-Based Chatbot," 2021, pp. 239–250. doi: 10.1007/978-981-15-9323-9_20.
[43] V. Ilievski, C. Musat, A. Hossman, and M. Baeriswyl, "Goal-Oriented Chatbot Dialog Management Bootstrapping with Transfer Learning," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, California: International Joint Conferences on Artificial Intelligence Organization, Jul. 2018, pp. 4115–4121. doi: 10.24963/ijcai.2018/572.
[44] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," 2020.
[45] B. Galitsky, "Chatbot Components and Architectures," in *Developing Enterprise Chatbots*, Cham: Springer International Publishing, 2019, pp. 13–51. doi: 10.1007/978-3-030-04299-8_2.
[46] C. O. Bilah, T. B. Adji, and N. A. Setiawan, "Intent Detection on Indonesian Text Using Convolutional Neural Network," in *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, IEEE, Jun. 2022, pp. 174–178. doi: 10.1109/CyberneticsCom55287.2022.9865291.
[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.04805
[48] M. Henderson *et al.*, "Training Neural Response Selection for Task-Oriented Dialogue Systems," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 5392–5404. doi: 10.18653/v1/P19-1536.
[49] T. Bunk, D. Varshneya, V. Vlasov, and A. Nichol, "DIET: Lightweight Language Understanding for Dialogue Systems," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.09936