applied sciences

MDPI

Article

A Multi-Attention Approach Using BERT and Stacked Bidirectional LSTM for Improved Dialogue State Tracking

Muhammad Asif Khan [1,†], Yi Huang [2,†], Junlan Feng [2,*], Bhuyan Kaibalya Prasad [3,†], Zafar Ali [1,†], Irfan Ullah [4,†] and Pavlos Kefalas [5,†]

1 School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
2 China Mobile Research Institute, Beijing 100053, China
3 Department of Electronics and Communication Engineering, National Institute of Technology, Rourkela 769008, India
4 Department of Computer Science, Shaheed Benazir Bhutto University, Sheringal 18050, Pakistan
5 Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece
* Correspondence: fengjunlan@chinamobile.com
† These authors contributed equally to this work.

Abstract: The modern digital world and associated innovative and state-of-the-art applications that characterize its presence, render the current digital age a captivating era for many worldwide. These innovations include dialogue systems, such as Apple's Siri, Google Now, and Microsoft's Cortana, that stay on the personal devices of users and assist them in their daily activities. These systems track the intentions of users by analyzing their speech, context by looking at their previous turns, and several other external details, and respond or act in the form of speech output. For these systems to work efficiently, a dialogue state tracking (DST) module is required to infer the current state of the dialogue in a conversation by processing previous states up to the current state. However, developing a DST module that tracks and exploit dialogue states effectively and accurately is challenging. The notable challenges that warrant immediate attention include scalability, handling the unseen slot-value pairs during training, and retraining the model with changes in the domain ontology. In this article, we present a new end-to-end framework by combining BERT, Stacked Bidirectional LSTM (BiLSTM), and a multiple attention mechanism to formalize DST as a classification problem and address the aforementioned issues. The BERT-based module encodes the user's and system's utterances. The Stacked BiLSTM extracts the contextual features and multiple attention mechanisms to calculate the attention between its hidden states and the utterance embeddings. We experimentally evaluated our method against the current approaches over a variety of datasets. The results indicate a significant overall improvement. The proposed model is scalable in terms of sharing the parameters and it considers the unseen instances during training.

Keywords: dialogue state tracking; attention mechanism; stacked BiLSTM; spoken dialogue systems; BERT; classification problem

Citation: Khan, M.A.; Huang, Y.; Feng, J.; Prasad, B.K.; Ali, Z.; Ullah, I.; Kefalas, P. A Multi-Attention Approach Using BERT and Stacked Bidirectional LSTM for Improved Dialogue State Tracking. *Appl. Sci.* **2023**, *13*, 1775. https://doi.org/10.3390/app13031775

Academic Editor: Valentino Santucci

Received: 18 December 2022
Revised: 14 January 2023
Accepted: 17 January 2023
Published: 30 January 2023

1. Introduction

During the last decade, verbal interaction with computing devices grew immensely popular since it provides an effective method of communication without using hands or eye contact with the system. The speech-based systems, which are also known as spoken dialog systems (SDS), give users the opportunity to verbally interact with the system to achieve a goal such as finding restaurants, airline tickets, and geographical locations [1,2]. Among the most popular SDS systems are Apple's Siri, Google Now, and Microsoft's Cortana, which are integrated into mobile products, applications, and services [3]. A spoken dialogue system contains five essential modules that are depicted in Figure 1. These include automatic speech recognition (ASR), natural language understanding (NLU), dialogue manager (DM), natural language generation (NLG), and text-to-speech synthesis [4]. Note

that a DM has two structural components, namely dialogue state tracking (DST) and the dialogue policy, which tracks the state of the dialogue [5–7]. The DST module is affected by the results of NLU to update or track the state. However, the majority of DST mechanisms in the literature ignore the NLU component, and results are retrieved directly from ASR to track the dialogue states [8].

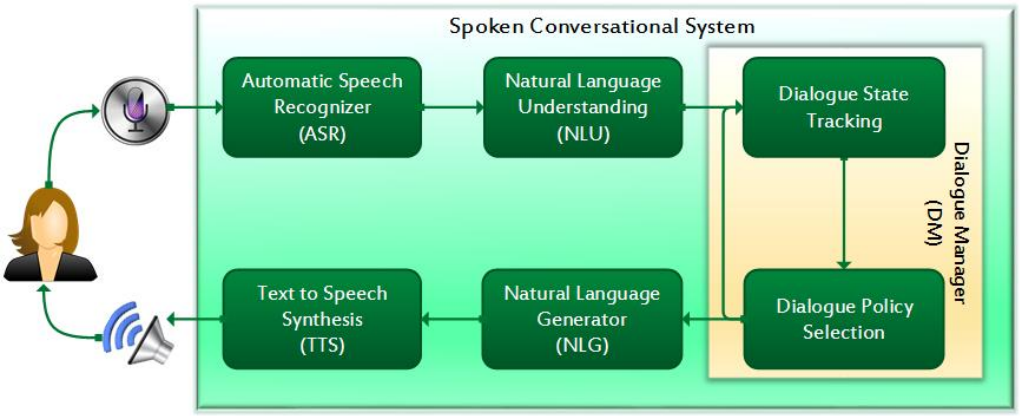

Figure 1. The principal components of a typical conversational system.

1.1. Research Background

Recently, DST has gained popularity in both industry and academia, as it is capable of determining the users' intentional states at each turn of the dialogue [9,10]. A state holds a set of dialogue acts and pairs of predefined slots and their corresponding values [11,12] in the form of dialogue states. Figure 2 presents an example of a dialogue conversation regarding a restaurant reservation taken from the WoZ-2.0 dataset. Here, we can see the dialogue acts (inform or request) and the slot value pairs (price range = expensive, food = Australian, food = Mediterranean). The dialogue state of the conversation is estimated by DST over the conversation history and current utterance of the user [8]. A state's accurate prediction is significant as it enables the system to perform the next action with greater accuracy and efficiency, and produces a personalized response for the target user.

User: I am looking for a restaurant in the expensive price range that serves Australasian food.
State: inform (price range=expensive, food= australasian).

System: There are no restaurants matching your requests. May I help you find a different type of restaurant?
User: I'd like to find some Mediterranean food and need to know their address and phone number?
State: Request (Address), request (phone), Inform (price range=expensive, food= mediterranean)

System: There is an expensive restaurant that serves mediterranean food located at Thompsons Lane Fen Ditton. Their phone number is 01223 362525.
User: Thank you!

Figure 2. A toy example of dialogue conversation based on WoZ-2.0 dataset.

For accurate state prediction, the DST models are either based on fixed or open vocabulary [13]. The fixed vocabulary-based DST models define a collection of slots and values from a predefined ontology [9,13], to extract features or identify patterns and build relations among the entities. The ontologies contain slots with known predefined values [14–16]. Moreover, these methods have demonstrated highly competitive performance over some popular DST datasets, such as DSTC-2 and WoZ-2.0 [17]. However, they have conceptually complex architectures and are heavily engineered [18–20]. Additionally, several models operate on fixed domain ontology and each slot type requires a separate model to train in the corresponding domain [8,12,20], which consequently increases complexity. Thus, the number of parameters and slot types are proportional to each other, rendering these approaches

prone to scalability issues. Although they can predict the values that are not available in history records, defining the possibly large slot value list per domain is quite challenging during the training phase. Formally, they are treated as a classification problem [12,21]. On the other hand, open-vocabulary-based models generate or extract the values for each slot from the context simply by using the encoder–decoder architecture [22,23] or pointer networks [24–26]. These approaches can share the parameters among the slots and domains in multi-domain datasets to overcome scalability issues [27]. They are flexible while extracting the slot values from dialogue history to solve the unseen problems [28]. However, various architectures are inefficient in context encoding from dialogue context because they encode all the previous utterances from history at each dialogue turn to aggregate the information [24,26]. Additionally, they are limited in predicting the slot values that were not detected during training.

To summarize, both DST approaches can handle scalability problems, parameters sharing over slots, and handling the growth of parameters when the domain ontology dynamically changes. However, the fixed-vocabulary-based DST tracker performs better on DSTC-2 and WoZ-2.0 datasets versus the multi-domain datasets [21,29]. They can observe unseen slot values. Therefore, this study focuses on creating a vocabulary-based model and formalizing it as a classification problem. The notion behind considering this as a classification problem is to further improve the performance of DST [19,30–33]. These models use recurrent networks to extract features from the history of the system and the user's sentence pairs. A classifier is employed upon the extracted features to predict the dialogue states. In contrast to [19,30–33], global–local self-attention (GLAD) [8], neural belief tracker (NBT) [12], global conditioned encoder (GCE) [16], and temporal excessive networks (TEN) [34] built a classifier to extract the features at each turn to predict the turn level state from current system action and user utterance pairs. The MDBT model [33] used multiple BiLSTMs to encode utterances of the system and user. Manh et al. [21] applied the classifier upon dialogue context and candidate slot value pairs to determine the relevance score of the candidate at a high level similar to the sentence pair classification task. Additionally, the state operation predictor employed in [35] as a classification task to output each slot representation on top of the encoder and additionally for domain classification task in between dialogue turns to the acquisition of slot operations and domain transformation. Other models [15,36] introduced a semantic parser (a binary classifier) containing a semantic tuple classifier (STC) approach to train on dialogue act consisting of slots and slot-value pairs in a topic to predict the presence of that slot-value pair in the long sentences. This discussion emphasizes choosing a fixed-vocabulary-based model for enhancing the quality of dialogue state prediction where DST exerts on sentence pair as a classification task. Owing to this emphasis, the next section briefly explains the proposed solution and presents our key contributions.

1.2. Key Contributions

We present the BERT-Stacked BiLSTM-Multiple-Attention (BLA) model, which aims to track the dialogue over a fixed vocabulary. The model combines BERT [37], Stacked BiLSTM (SBL) [38], along with multiple-attention. In addition, we have implemented a pre-trained language model for word embedding in our framework. With the success of pre-trained language models such as ELMO [39] and BERT [37] over the last few years, only a few researchers had used these models in the domain of DST upon the datasets WoZ-2.0 and DSTC-2 to tackle the classification problem. As stated earlier, the classification problem lies within a special category where distinguished frameworks apply sentence classification to themselves. To this point, we extended the BERT network by incorporating SBL and multi-attention to learn the contextualized word representation, which increased the robustness of the model in terms of predicting the slots and their corresponding values. Our contributions are summarized below:

- We present BLA to tackle the sentence classification problem in DST by taking a dialogue context and candidate slot value pair to accurately track slots and their corresponding values.
- We propose a model that exploits the sequential and overall features encoded in BERT to improve the performance of neural networks.
- We extract the contextual word features from a dialogue context using the BERT pre-training language model along with Stacked BiLSTM, and multi-attentions.
- Upon evaluating our model on two real-world datasets, we achieved state-of-the-art performance over the current baseline models regarding turn request and joint goal accuracies.
- With detailed experiments in the ablation study on distinguishing variants of neural networks, we have confirmed the importance of feature extraction from Stacked BiLSTM and Multiple Attention Mechanism for the proposed model in terms of the improvement over turn goal, turn request, and joint goal accuracies.

1.3. Organization of the Paper

The rest of the paper is organized as follows. Section 2 presents the latest related work. Section 3 presents the proposed methodology. Section 4 experimentally evaluates our model against baselines over two popular datasets. Section 5 concludes the paper with findings and future directions.

2. Related Work

A large number of models focus on DST aiming to scale over realistically sized dialogue problems and perform in real-time. During those years, DST has shown improvement in the overall performance of spoken dialog systems. This section examines related methods and techniques used to exploit information and generate DST. The DST approaches can be categorized into two types: (a) traditional approaches, i.e., rule-based, SVM, Bayesian, maximum entropy and delexicalized, (b) deep-learning (DL)-based approaches, including deep neural networks and BERT, discussed in the following subsections.

2.1. Traditional DST Approaches

Most of the DST detection methods in the literature employ rule-based and machine-learning approaches. A spoken dialogue system (SDS) can be imitated as a decision process where the uncertainty in tracking its dialogue state occurs due to errors that pervade through ASR and SLU/NLU [40–42]. These impairments were resolved by [43] through a simple and generic rule-based DST mechanism called GENR. It is based on simple domain-independent rules that use basic probability operations without requiring external knowledge. It infers the immediate information gained from observable system actions and partially observed user acts. The observation is SLU n-best list and normalized confidence scores, which are similar to those proposed in [3,44,45]. However, GENR targets miss the intermediate analysis of the information gained from SLU n-best lists and the confidence scores are produced without learning prior knowledge. Additionally, GENR uses data-driven methods or is designed based on domain-specific strategies beyond considering only the SLU hypothesis.

Another popular approach to predict labels, annotations, or slot-value pairs in DST is the use of maximum margin classifiers, such as support vector machines (SVMs). For instance, [36] proposed a hybrid-based architecture consisting of traditional and deep-learning-based approaches (see next section for DL-based approaches of the model). They used a semantic parser that requires a set of SVMs trained on n-gram features to predict the dialog act type by multi-class SVM, and a binary SVM to predict the existence of each slot-value pair from an utterance. The dialogue context features and the n-gram features of the original STC parser were exploited to constrain the semantic parser. The final vector is obtained by combining the original n-gram feature with the reciprocal of the turn index and the context features from the last system action. Finally, normalization via a semantic

parser is applied to each of the top ASR hypotheses to guarantee the sum for generating confidence scores to test and predict the dialogue act.

Few other approaches [46–48] assume that dialogue can be modeled as a probabilistic model such as Bayesian network that relates dialogue state s to the system action a, the true or unobserved user action u, and ASR or SLU result $\hat{u}$ as cited in Figure 3. When the ASR/SLU and the system action are observed, Bayesian inference is applied to compute the distribution over possible dialogue states. Several probabilistic formulations have been researched for relating these quantities [46,47]. For example, [48] employed a dynamic Bayesian network to learn the user action for a given text directly from the machine-transcribed corpus. In another work, Yu et al. [49] used constrained Markov Bayesian polynomial (CMBP) based on Bayes theorem to enhance the power of the rule-based model. The primary idea was to demolish the necessary probabilistic conditions and prior knowledge to overcome the optimization problem for the DST model during training.

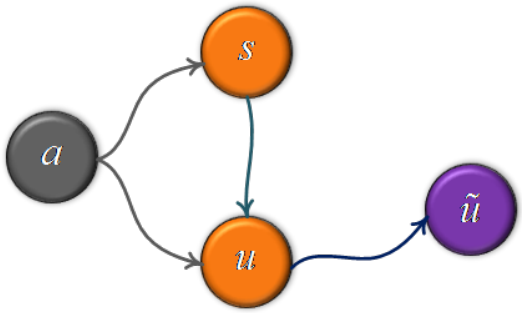

Figure 3. A typical Bayesian network for DST.

Some other opportunities such as maximum entropy (MaxEnt) used in belief tracking mechanism to calculate beliefs [50] upon a state through parametric models, which directly present the belief $b(s_{t+1}) = p(s_{t+1}|s_t, z_t)$ in connection to maximum entropy that has been widely implemented as discriminative approach [51] in variety of models. It creates the belief methodically as $P(s|x) = \eta.e^{w^T \varnothing(x,s)}$; where η represents normalizing constant, $x = (d_1^u), d_1^m), s_1, \ldots, d_t^u), d_t^m), s_t)$ denotes the history of user dialogue acts, $d_i^u \epsilon \{1, \ldots t\}$ are system dialog acts, $d_i^m \epsilon \{1, \ldots t\}$ presents the sequence of beliefs unto the current dialogue at time $t, \phi(.)$ is vector features on x and s, and w denotes the set of model parameters learned from the annotated dataset.

Sun et al. [36] designed six models of MaxEnt where they parted one model for method, one for the requested slot, and four for joint goals. The models for the joint goals were applied on four informable slots (price range, food, area, name), separately. For each informable slot-value pair in SLU in k-th turn, the MaxEnt for the corresponding slot was employed to see whether the slot value in the user goad is right or not. The input of the model consists of 160 features extracted from the feature functions of joint goals and the model passed the confidence score as output. The MaxEnt model for the method was implemented to determine the user interaction with the system. The input comprises 97 features extracted from feature functions of the method and the output contains five confidence scores. The model for the requested slots determined the true or false requests of a user in the SLU. The input is comprised of 10 features extracted from feature functions of requested slots and a true confidence score is the output.

Poor generalization regarding morphological or lexical variations in the training data, presents a constant key challenge in the DST domain. This can be counteracted using various techniques [47] such as delexicalization features. The systems using delexicalization pose exact matching of words for slot-value pairs generation with the construction of semantic dictionaries to identify alternative mentions of ontology [12,31]. For example, "*I am looking for a cheaper restaurant in the downtown area.*" where the words cheaper and downtown rephrase into Cheap and Centre, respectively. This type of word-rephrasing occurs with the adoption of a defined semantic dictionary with rephrasing for ontology val-

ues Food = Cheaper: (affordable, low cost, inexpensive, low budget, cheaper, cheaply, ...), Area = Centre: (central, center, downtown, midtown, center of the city, ...). The delexicalized features for RNN word-based tracking can be used to share training data across disparate domains and improve performance in each individual domain [32].

Unfortunately, all the aforementioned models are typically heavily engineered with rules, and generative and discriminative approaches such as delexicalization, Bayesian networks, maximum entropy, and SVM. Thus, the extraction of features relevant to each state and the lack of learning the semantic information is challenging for these approaches. These models require separate training for each slot value. Additionally, the word embedding techniques are mostly dense-based, that cannot extract words features in a contextual way. Therefore, downstream modules are prone to morphological and lexical errors. Furthermore, the parameters grow with the ontology size and are proportional to the number of slots [21]. To overcome these issues, the model requires extra techniques to reduce the issues that increase the complexity of the model architecture.

2.2. Deep-Learning-Based Approaches

In recent years, numerous studies applied DL-based approaches (DLA) to natural language processing (NLP) problems [52,53] including DST in spoken dialogue systems [21,29]. DLA is a set of algorithms, i.e., deep neural networks (DNNs) such as convolutional neural network (CNN), gated recurrent unit (GRU), long short-term memory (LSTM), attention mechanism, and BERT, that can learn a complex mapping between input and output space [54]. These approaches demonstrated excellent performance on DST datasets such as WoZ-2.0 [17] and DSTC-2 [18]. Moreover, DLA has produced promising results in the field of DST for learning semantic neural representations of words [55,56], point slot values in conversation [24], neural dialogue generation [57], and sharing parameters among estimators for various dialogue states [16,58] as well outlined in Table 1.

In connection with the above discussion, Ref [59] proposed a deep neural network as a classifier for the DST mechanism to train the entry of the DST challenge dataset. The aim of deploying the DNN was for improving the probability distribution over possible values by learning the tied weights and use of sliding windows. Sun et al. [36] designed four DNNs for joint goals (one for each informable slot), one for method, and one for requested slots. These DNNs have a similar structure with softmax for output layer activation and sigmoid for hidden layer activation. Each DNN takes the feature set of a certain value of the goal, method, or requested slots as the input and outputs two values through the hidden layer to predict the confidence score as output. Mrkšić et al. [12] presented a neural belief tracker (NBT) that employed multi-layers of neural networks including CNN. It uses the previous system output, viz., the last utterance takes the user utterance as the current utterance and parses the system output and current utterance to come up with the current state of dialogue. Another framework, GLAD [8] designed LSTM with the combination of attention mechanism to estimate each slot value pair independently. It uses global modules to share parameters between estimators for each slot and local modules to learn slot-specific feature representations. Kumar et al. proposed a multi-attention-based scalable DST (MA-DST) framework [60] to tackle DST as a classification problem. The model contains GRU, and multi-attention to encode the dialogue context and slot semantics that enhance the performance of DST robustly in a scalable manner and perceive the unseen words during training.

Table 1. Comparison of latest studies exploring the DST problem, where TG, TR, and JGA, denote Turn Goal, Turn Request, and Joint Goal Accuracy, respectively.

Model	Methodology	Strengths	Limitations	Performance (TG, TR, JGA)
GLAD [8]	• Encodes each utterance and previous system actions by modeling slot-value pairs using BiLSTM and learns slot specific features. • The attention mechanism computes the slot-specific temporal and context summary	• Estimates each slot-value pair independently. • Learns feature representations via parameters sharing. • Generalization on rare slot-value pairs.	• Each slot type requires a separate model to train. • Pretrained Glove embeddings and character n-gram for words representation. • Deterministic rule propagate errors to future turns, may lead to wrong state aggregation.	TG, TR, JGA
TEN [34]	• Captures temporal dependences across dialogue turns using GRU. • The classifier extracts feature at each turn to predict the turn level state from current system action and user utterance. • The graph factor aggregates state dependencies. • Belief propagation handles the uncertainties in state aggregation.	• Improves turn level state prediction. • Reduces the state aggregation errors. • Requires less parameter estimation for the state aggregator. • Robust for the state aggregation.	• Needs to be improved with respect to graph factor with another graph neural network for state aggregation. • Pretrained Glove embeddings and character n-gram for words representation. • Conceptually complex framework due to feature engineering.	JGA
Seq2Seq-DU [61]	• BERT-based encoder for the utterances in sequence-to-sequence fashion. • Schema descriptions deal with unseen domains. • State representation based on LSTM and pointer network. • Attention between utterance embeddings and schema descriptions.	• Sequence-to-sequence framework to model intents, slots and slot-values as global. • Learns representations of current and previous utterances during encoding and schema descriptions. • Scalable.	• Closely related to specific dialogue contexts. • Ignores slot domain membership relations. • Low prediction accuracy.	JGA
MSP [56]	• Slot specific memory holds slots and values to handle state update. • BERT for dialogue contextual representation. • Hit type prediction layer maps representations of memory and dialogue contexts.	• Update dialogue state accurately. • Previous wrong predicted values may correct in new value prediction	• The framework update state may reduce the performance of general tracking the slot-value pairs. • Increases the usage of system's memory	JGA
Trippy-r [62]	• Fills slots with values copied from the context, predictions from previous turns or system informs. • Feed forward network for classification. • Tracks constraints via intent detection. • Triple copy strategy to copy values from the context, predictions, previous turns and system information. • Slot attention layer to token representations of dialogue turn.	• Handles unseen values during training robustly. • Economically alternative to data augmentation to prevent memorization and over-fitting. • Domain agnostic to facilitate transfer to unseen slots and d • Capable to track new domains by learning from non-dialogue data.	• Requires intent detection module. • Weak supervision and sparsity of data may render model prone to errors in the prediction of slot-value pairs.	JGA

Several studies rely on hand-crafted rules to extract the features and delexicalization for lexical errors [31,43,63] that rise the scalability challenge in the framework during the classification task for DST. It has been noticed that stacked BiLSTM [38] is a better alternative for features extraction [64,65]. Stacked BiLSTM is the extension of LSTM and a deep bidirectional neural network that learns deeper representations up to the two or more layers [38]. It performs better on classification tasks for speech recognition and prediction-based management systems such as domain and intent detection of dialogue systems [38,64].

In another study, [66] introduced a model to deal with DST as a sequence-to-sequence problem and mitigate other problems including scalability and unseen schema in new domains on multi-domain datasets. The model employed BERT to learn and utilize a better representation of the current and previous utterances. Moreover, a pointer network was utilized to build the representations of intents, slots, and values among their corresponding values. Tian et al. [66] employed to take the previous dialogue state and the dialogue of the current turn to produce a primitive dialogue state. Later, the current turn of the dialogue and primitive dialogue state is passed to the amendable generator to generate the amendable dialogue state that contains the mistake-free dialogue state for further predictions of dialogue states. Wang et al. [28] developed a stack-propagation framework to jointly process the slot filling and come up with end-to-end DST by exploiting a BERT-based dialogue history encoder to procure the dialogue history representation and initiate the slot value decoder. They employed a slot mask attention mechanism to get key slot information detected by slot filling to improve the updated dialogue state. They used a slot-value softcopy mechanism to exploit words marked by slot filling.

Zhang et al. [58] used two BERT-based encoders and designed a hybrid approach for fixed-ontology- and open-vocabulary-based DST. They defined pick list-based slots for classification and span-based slots for span extraction as DST readers. Shan et al. [67] employed a contextual hierarchical attention network based on BERT and an adaptive objective to alleviate the slot imbalance problem by dynamically adjusting the weights of slots during training. To overcome the substantial noise in the state annotation, Ye et al. [38] used a pre-trained language model BERT such as [11,15,55] that was fine-tuned during training and encoded dialogue context into contextual semantic representations. Ye, Feng, and Yilmaz [68] worked on the noise labels in the state annotations of datasets to increase the efficiency of DST. They employed one BERT to encode the dialogue context into contextual semantic representations and another one to encode the candidate slot-value pairs into semantic vectors. Multi-head slot attention was used to obtain the information relevant to all slots from the same dialogue context. Hu et al. [69] developed IC-DST, a text-to-SQL approach to define each domain as a table and each slot as a column for dialogue states representations. The aim was to overcome the limitations of fine-tuning pre-trained language models, retrain the systems again when a new slot or domain is added, and achieve reasonable performance on few-shot settings.

In summary, by analyzing the related works presented in this section, we suspect that DST needs to focus on managing scalability problems, unseen values observed during training, and retrain the model when new values appear. This conclusion formalizes DST as a sentence classification task, that can be processed by generating slots and their corresponding values from the learning through the current turn and dialogue history to recognize the dialogue state in fixed domain-ontology-based frameworks. Therefore, we devised a model that is capable of sharing parameters across the slot types and scalable on both datasets for addressing the scalability issue. The model takes candidate slots and value pairs as input to preserve them in the sequence of words. This way, the model can be applied directly to the newly occurred slot values that were unseen during training. The Stacked BiLSTM should be used to exploit features from the last hidden state of BERT to capture the contextual and semantic features of words. This will improve the classification of slot-value pairs from user and system utterances. In addition, parameters of the model does not grow using a BERT-based uncased model with fixed parameters of 110 M. The proposed model should not require a retraining model when domain ontology changes dynamically. The self-attention mechanism and a combination of mask-input IDs attention and self-attention can be used to focus more precisely on the features extraction and sentence classification to enhance the performance of DST. We present this vision in the next section.

3. Methodology

This section presents an overview of the proposed model in detail as cited in Figure 4. The model has three modules, namely BERT, Stacked BiLSTM (SBL), and multiple attention mechanism. The BERT module takes the input of words from the dialogue context and candidate. The contextual embeddings assign each word a representation of its context obtained by the BERT pre-training language model. The vector representation of each word in the text is passed to the BERT module to obtain the sequential and overall features. The sequential features are input into SBL to obtain the hidden features and actual output. The hidden features of SBL are fed into global attention. The local attention is applied to the non-zero words generated by the BERT tokenizer in the form of input ids. Finally, the overall features from BERT, the actual output from LSTM, and the output of multiple attention mechanism are concatenated for final recognition.

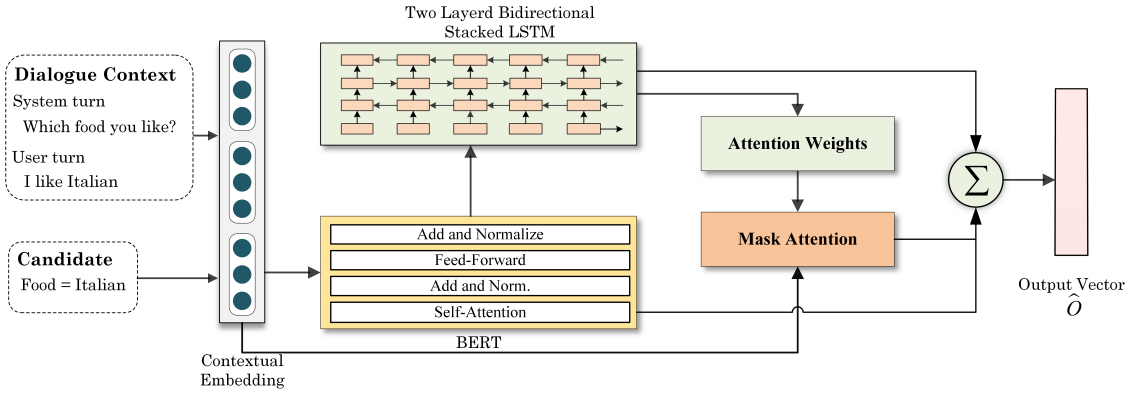

Figure 4. A schematic view of the proposed model.

3.1. Contextual Word Representation

Many pre-trained contextual language models such as ELMO [39] and BERT [37] have gained popularity in NLP tasks. The BERT language model is used to extract semantic features of the tokens by creating relationships among tokens [70] with different contexts in sentence pair classification problems. In view of this, we designed BERT as a contextual word representation encoder to get semantic feature vector representations of dialogue contexts and candidates. The use of BERT is inspired by the fact that it is a powerful deep bidirectional language representation learning model with multiple transformer layers [37], each having 12 self-attention heads and a hidden size of 768 units [71]. We used BERT to produce not token-specific vector representation for each taken in a sentence as well as an aggregated vector representation for the whole sentence.

3.2. Dialogue Context Encoding

In order to encode the dialogue context, which holds utterances of the user and system (from its previous turn), it is fed to BERT. The candidate that contains the slot-value pair is fed to BERT. The two inputs pass through the tokenizer layer to split into a list of tokens. Next, the model produces an output fixed-length vector by concatenating the token, position, and segment embedding. Next, at the start of each utterance, a special classification [CLS] token is inserted and a special token [SEP] is utilized to separate two utterances and candidates. The output vector, i.e., the hidden states U_h^m of each token is then computed by BERT as shown in Equation (1).

$$U_h^m = BERT(S_{[CLS]}, S_1, S_2, \ldots, S_n, S_{[SEP]}) \tag{1}$$

where U_h^m denotes $h_{[CLS]}, h_1^m, h_2^m, \ldots, h_{[SEP]}^{[m]}$, m represents the number of BERT encoder layers, and h_t is the embedding of token t. The BERT model predicts the probability score for candidate slot value pairs that are only equal to or greater than zero at each turn. To

obtain the predicted slot value pairs at the current turn, we used the newly predicted slot-value pairs to update previously predicted values. For example, a user enquired about food = Italian at the current turn. However, if the cuisine does not exist then it will be added as newly corresponding values. Likewise, if the user modified cuisine Italian into food = Turkish then the corresponding values will be updated. Finally, we pass the last hidden state $U_h = U_h^L (L = 12)$ to SBL and attention to predict the slot value pairs.

3.3. Stacked BiLSTM for Feature Extraction

To further learn better contextual and semantic features we deploy SBL on the top of BERT's last hidden state. As stated earlier, SBL is the extension of LSTM stacked with multiple layers of neural networks to learn the features from dialogue context deeply. It has been perceived that multiple layers of LSTM reduce the validation loss more than single layer of LSTM to enhance the accuracies of DST (Section 5.3). For better understanding of multiple layers of Stacked BiLSTM, we will investigate the intuition of LSTM. LSTM is used to solve the long-range dependency of the sequence and vanishing the gradient problem [72]. The vanishing gradient problem occurs when the gradients are back propagated through the network then the network can vastly rot or grow [73]. For example, when multiple layers using the activation function are added to a network then the gradients of the loss function approach toward zero, making its training a challenging task. The purpose of LSTM is to add, delete or update dialogue context and candidate information in cells organized by three gating mechanisms. Gating mechanisms at time step s renowned as input gate x_s, forget gate y_s and output gate z_s. These gating mechanisms with forward and backward layers control the flow of information for reading, writing, and updating in the last hidden states. The first step of the LSTM block is to decide which features from dialogue context have to be excluded from the cell status, equipped by forget gate as shown in Equation (2).

$$y_s = \sigma(V^f I_s + h_{s-1} P^f + b_f) \tag{2}$$

Here, I_s is the input V^f and P^f are the weight matrices, b_f bias vector parameters, h_{s-1} is the hidden state of the previous state at the time step s. Then the next step for the LSTM block is to decide what batch of new features from dialogue context and candidates has to be stored in the cell state calculated by the input gate x_s about the corresponding values update and tanh layer that creates a vector of a new candidate $\hat{C}_s$ as shown in Equations (3) and (4), respectively. Then, the status of the previously stored utterances and values are updated and calculated by element-wise multiplication $\times$ between its inputs as shown in Equation (5).

$$x_s = \sigma(V^x I_s + h_{s-1} P^x + b_x) \tag{3}$$

$$\hat{C}_s = \sigma(V^C I_s + h_{s-1} P^C + b_c) \tag{4}$$

$$C_s = (C_{s-1} \times x_s \times y_s \times \hat{C}_s) \tag{5}$$

Lastly, the output gate z_s decides what output from which part of the cell status must be generated. This output is the updated version and mathematically represented in Equation (6). At each turn, if the new corresponding value against slot occurred during the current utterance then the input gate extracts the new value. Afterwards, the previous state is forgotten by forget gate and new state is updated by C_s to produce hidden state as shown in Equation (7).

$$z_s = \sigma(V^z + h_{s-1} P^z + b_z \tag{6}$$

$$h_s = z_s \times tanh(C_s) \tag{7}$$

To enhance the performance of typical LSTM, we have implemented forward and backward layers of LSTM to encode past and future information of dialogue context and candidate. The input I_t, which holds the last hidden state of BERT, is fed to the first BiLSTM

block with the previous hidden state $h_{t-1}^{(1)}$ in the Stacked BiLSTM. We have used two stacked layers of BiLSTM. The hidden state at the time step s is calculated as mentioned earlier. Later, it moves to the next time step while also shifting up to the second block of Stacked BiLSTM. The forward and backward stacked layers of LSTM read all the words in the last hidden state $U_h = U_h^L(L = 12)$ generated by BERT to output the actual output r_s, which contains the whole hidden states. Meanwhile, the second output for hidden features of the LSTM H_s is generated. Here, both r_s and H_s contain the words with information of contextual features in opposite directions at each time step. The two opposite direction-hidden state $\overrightarrow{H}$ and $\overleftarrow{H}$ generated by forward and backward LSTM are concatenated as shown in Equation (8).

$$\begin{aligned} \overrightarrow{H} &= h_s(U_1^L, U_2^L, \ldots, U_h^L) \\ \overleftarrow{H} &= h_s(U_1^L, U_2^L, \ldots, U_h^L) \\ H_s &= [\overrightarrow{H}\,\overleftarrow{H}] \end{aligned} \tag{8}$$

3.4. Multiple Attention

The attention mechanism has been successful in various domains including machine translation, NLP, and image processing. It analyzes a given input sentence to obtain relevant contextual information about each word. In particular, the attention mechanism is applied to the sentences to focus on information that is more important. We have applied multiple attention to different parts of the sentence from different modules of the model. In the first stage, we have used attention upon the whole hidden states $r_s = (h_1, h_2, \ldots, h_s)$ produced by Stacked BiLSTM. The function of the attention block is to compute the context vector at each time step l_t using the weighted sum of the annotations α_{mn} and hidden states as described in Equation (9).

$$\begin{aligned} l_t &= \sum_{k=1} r_s \alpha_{mn} \\ \overleftarrow{\alpha}_{mn} &= \frac{exp(S^T S_n)}{\sum_{k=1} exp(S^T S_n)} \\ s &= tanh(W_n p_{jk} + b_n) \end{aligned} \tag{9}$$

where α_{mn} is calculated by the dot product between S^T and high-level representation of node-level context vector s_n using softmax activation function to generate normalized adaptive weights. W_n and b_n are trainable weight matrices and trainable bias, respectively. Moreover, s represents and p_{jk} denotes the input of multi-layer perceptron (MLP) which is used to scale the values of hidden states s. In the second stage, we used the input IDs as tokens instead of padding generated by the BERT tokenizer for masking to apply another attention, i.e., the masking on non-zero elements of the sentence. At the same time, the results of the first attention as context vector at each time step l_t are concatenated $\bigotimes$ with second attention, as described in Equation (10).

$$ATT = software(l_t m) \tag{10}$$

Afterward, we concatenate the output of BERT, Stacked BiLSTM hidden features, and multiple attention to calculate the probability value for the dialogue state as described in Equation (11). Then, the classifier is built upon $\hat{O}$ for classification. Finally, the categorical cross-entropy function is designed to calculate and minimize the loss between the actual dialogue state and predicted dialogue state as described in Equation (12), where i_s is the number of dialogue states, O_i is the actual dialogue state and $\hat{O}_i$ is the predicted dialogue state.

$$\hat{O} = U_h^m \otimes H_s \otimes Att \tag{11}$$

$$L(O, \hat{O}) = -\sum_{i-1} O_i \hat{O}_i \quad (12)$$

4. Experiments

This section presents the experimental and evaluation details including the datasets, metrics, and baseline models used in the comparative analysis.

4.1. Datasets

We have used two datasets namely DSTC-2 and WoZ-2.0, which were provided by thankful to Junfan Chen [34]. The motivation behind the selection of the dataset for this study is to improve the transformer-based dialogue state-tracking performance on a single-domain dataset. Currently, DSTC-2 and WoZ 2.0 are the two famous datasets utilized by numerous researchers. Furthermore, they are easily publicly accessible and more suitable for single-domain DST. Moreover, the aim behind the selection of single-domain datasets is to provide a fundamental study in the domain of DST. The DSTC-2 dataset belongs to the dialog system technology challenge (DSTC), which is an ongoing series of research community challenge tasks. It is used in human-to-computer dialogues in the domain of information about restaurants. Henderson, Thomson, and Williams organized DSTC-2 with the DSTC-3 dataset and the produced results were presented in sessions at SIDIAL 2014 and IEEE SLT 2014 (https://www.microsoft.com/en-us/research/event/dialog-state-tracking-challenge/, accessed on 12 November 2022). The WoZ-2.0 dataset is also known as the Cambridge Restaurant Corpus dataset, collected for developing neural network-based dialogue systems. It was collected through the Wizard of Oz experiment on Amazon MTurk similar to MultiWoZ corpus [12,35]. The WoZ-2.0 includes single-domain dialogues. Each dialogue holds a goal label and several exchanges between the system and its customer. Each user turn was labeled by a set of slot-value pairs that represent a coarse representation of the dialogue state. Most of the dialogues are finished but some of the dialogues were not in WoZ-2.0 dataset [74]. Table 2 presents the statistics of datasets as per domains associated with the dataset in column 2, and the slots which are available for the assessment of dialogue state trackers are shown in column 3. Moreover, in the following columns, we have demonstrated the total utterances, average turns, total dialogues, total words, and key points in the description column.

Table 2. Statistics of Used Datasets.

S. No.	Name	Topic	Slots	Total Utterances	Average Turns	Total Dialogues	Total Words
1	DSTC-2	Restaurant	area, food, name, price range, address, phone, postcode, signature	24,049	7.88	3000	432 K
2	WoZ-2.0	Restaurant	food, price range, area	3452	4.24	1200	22,347

4.2. Training Setup

We used BERT-based-Uncased model (https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-vocab.txt, accessed on 12 November 2022) for the DST classification with 12 transformer layers, 12 attention heads, 768 hidden states, and 110 M parameters. We utilize linear learning rate scheduler method with initial fixed learning rate of 2×10^{-5} to optimize the lose function. That regulate the batch size at each iteration while moving to process the loss function. Moreover, a warm-up proportion of 0.1 is set to reduce the primary effect of early training. Our proposed model trained on AdamW and SGDW optimizer [75], used for optimizing the loss with linear learning rate. The batch size is 16 with BiLSTM 256 hidden dimensional layers for contextual representation of words features extraction. Furthermore, a dropout of 0.1 was used to avoid overfitting in the model for BERT's all fully connected layers and attention probabilities. The dropout for other layers of the model is set to 0.25. In addition, 22 epochs have been used for training. Both two datasets have negative examples greater than positive examples. To handle this issue, we

utilize the conversion of negative examples into positive examples for data imbalance. Class imbalance problem is a common issue faced by researchers in classification tasks during the machine learning algorithms [76,77] that lead to reduced accuracies such as precision, recall, turn request, turn goal, and joint goal. For example, the sentences from corpus have been taken as positive examples; however, segments from Bert (i.e., produced wrong slot-values or failed to update slots) have been taken as negative examples or false class labels. Thus, we use random oversampling to solve the class imbalance problem. Furthermore, We populated the existing data to keep the ratio the same during training. A summary of the hyper-parameters used for our experiments is given in Table 3.

Table 3. Hyper-Parameters Used in Experiments.

Layer	Hyper-Parameter	Size
BERT	dimension	786
LSTM	hidden Layer	256
	number of Layers	2
Learning Rate	learning rate	2×10^{-5}
LSTM Dropout	dropout	0.25
Number of Labels	output for input sample	2
Dropout	dropout rate	0.25
Warmup-Proportion	warmup steps	0.1

4.3. Evaluation Metrics

This section presents the metrics used in evaluating the results of the experiments. These metrics include the following.

4.3.1. Turn Goal Accuracy

A user may inform the system about certain goals (e.g., Inform (food = Indian), (area = center)) given during each turn. For instance, food and area are examples of the informable slots in the WoZ-2.0 and DSTC-2 datasets. On the other hand, Indian and center are examples of values corresponding to their slots. DST uses context from previous turns, goals from user utterances at each turn, other external details, and the system's output at every turn. The turn goal accuracy [8] is the fraction of user turns for which goals have been correctly predicted by the model. To incorporate information regarding each slot, there are BERT, BiLSTM, and self-attention networks for each slot. In this way, to pay attention to the slot-only information in the input sequence X, the computed attention is conditioned on the slot embedding.

4.3.2. Turn Request Accuracy

The turn request slots used by [8,16,21,78] refer to requests and its value is the category that the user demands, e.g., phone, area. The turn request accuracy is the accuracy of the information requested by the system or user in each requestable turn that arises during the dialog conversation [16]. It calculates the percentage of turns in a dialogue where the user's requests were correctly identified [79]. The turns with no requested slots in ground truth and predictions are skipped [80].

4.3.3. Joint Goal Accuracy

The joint goal accuracy evaluates the effectiveness of DST [8,21,34,55]. It compares the predicted dialogue states with the ground truth at each dialogue turn [81]. It is the percentage of turns where the user's informed joint goals are considered correct if all the predicted values exactly match the ground truth [79]. The joint goals are the set of accumulated turn goals up to the current turn [8].

4.4. Baseline Models

This section discusses and compares the following baseline models to assess the performance of our proposed model.

- **NBT** Neural Belief Tracker (NBT) [20] updates its internal representation of the states of conversion at each turn in data driver fashion. It was the first neural-based approach for DST, and used word embedding performing on par with the models developing engineered lexical rules.
- **GLAD** Global Locally Self Attentive Dialogue State Tracker (GLAD) [8] consists of two modules. The global module shares parameters among slots through self-attentive RNNs. The local module learns features that consider specific slots. The model uses the previous actions of the system and the current utterance of the user as input to compute meaningful similarities with predefined ontological terms.
- **Statenet** Universal dialogue state tracker (Statenet) [16] generates a fixed length representation of the dialogue history and compares the distance between the representation and the value vectors in the candidate sets. It uses the original ASR information of the user utterances, information about the machine's acts, and literal names of slots and their corresponding values. The values from candidate sets are set to be dynamically changed in the model.
- **GSAT** Global encoder and Slot-Attentive decoders (GSAT) [82] is a highly robust dialogue state tracker that predicts the dialogue states with or without using pre-trained embeddings. In addition, GSAT consists of a recurrent neural network-based single global encoder and slot-attentive decoders as classifiers. The encoder module encodes the current user utterance and previous system action from history to output the context vector and hidden representation for each token. The classifier module processes the hidden representations of the encoder and possible slot values to provide the probability distribution for each possible slot value.
- **COMER** Conditional Memory Relation Network (COMER) [25] uses two sequential decoders to formulate dialogue state as a sequence generation problem instead of a pair-wise prediction problem. First, its encoder–decoder network generates the slot sequences in the dialogue state and then for each slot generates the corresponding value sequences by conditioning on the dialogue history. The parameters are shared across all the decoders to overcome the scalability of the hierarchical structure of the dialogue state.
- **Full Bert** is a simple but effective BERT-based model [21] proposed for recourse-limited systems to track the dialogue states. The full BERT approach uses BERT to control the parameters to not grow when the ontology changes. It takes candidate and dialogue contexts and produces a score that indicates the relevance of the candidate by considering a dialogue context and a candidate slot-value pair.
- **TEN** Neural dialogue state tracking with temporally expressive networks (TEN) [34] maintains the state of the system for tracking the progress of dialogue. In this approach, two aspects of state tracking were iterated to improve results. These include (a) temporal feature dependencies in model design and (b) uncertainties in state aggregation more expressively modeled.
- **SUMBT** Slot-utterance matching for universal and scalable belief tracking [55] focuses on developing a scalable and universal DST. It uses an encoder to encode the system and user utterances. It provides the contextualized semantic representation of sentences. The encoder also encodes the slots and their corresponding values. It also learns the slot value relationships that appear in dialogues.
- **Seq2Seq-DU** A sequence-to-sequence approach to dialogue state tracking (Seq2Seq-DU) [61] transforms all the utterances in dialogue into semantic frames. It employs encoding of utterances and schema descriptions sequentially to generate pointers in decoding of dialogue states.
- **AG-DST** Amendable generation for DST [66] takes the dialogue of the current and previous turns as input. This same process of taking input is in a two-pass process as

basic generation and amending generation to output the primitive dialogue state for basic generation and amended dialogue state for amending generation.

- **BLA** implements SBL to exploit contextual information from the last hidden state of BERT. The global attention is used over the overall output of SBL, and local attention on the non-zero words from the BERT tokenizer input IDs to extract the words features. Finally, the overall outputs from three modules BERT, SBL, and multiple attentions are concatenated to generate the dialogue state.

5. Results and Discussion

This section presents the comparative analysis of the experimental results to understand how the proposed system improved performance compared to the selected baselines. It also presents an ablation study to understand the impact of different constituent components on performance.

5.1. Comparative Analysis of Experimental Results

The experimental results in Table 4 show the performance of proposed model in this study against the selected baselines on DSTC-2.0 and WoZ-2 datasets. It can be observed that the proposed model outperforms the baseline models on DSTC-2 in terms of joint goal and request accuracies. NBT performed poorer due to its use of dense embeddings and mapping directly from turns to states. Furthermore, it suffers from scalability problems to large output unbound space and requires a separate model to train for each slot type in the domain ontology. These limitations led to the model not sharing parameters across slots. Additionally, the slots are trained independently without sharing information among them. The second best results in Table 4 on the DSTC-2 dataset are produced by Seq2Seq-DU, bearing a sequence-to-sequence scalable architecture that can share the parameters across all slots and domains for single and multi-domain datasets. It uses contextual embeddings that produce better results in DST. However, it is unable to focus more on the feature extraction over the BERT output and attention over all the non-zero words of input IDs from pre-trained contextual embeddings. However, the proposed model performed 2.2% better than Seq2Seq-DU in terms of joint goal accuracy. This is because our model is more prominent in the feature extraction from the dialogue context through Stacked BiLSTM. Additionally, it considers the significance of relations among the values to their corresponding slots. The results produced on WoZ-2.0 in Table 4 are better than the DSTC-2 dataset. It is because the WoZ-2.0 dataset has less ambiguity with additional annotation corrections. The second-best results among the all other baseline models are produced by AG-DST in terms of joint goal accuracy. The reason behind the great impact on the results by AG-DST is that it utilizes the input sequence to compose the current turn dialogue from the previous dialogue state. Our proposed framework is scalable to share the parameters and create relations between values and their corresponding values. In contrast, NBT and GLAD produced poor results as compared to other models in respect of turn request and joint goal accuracies. NBT and GLAD have some limitations as compared to other latest frameworks in the way that the number of parameters is proportional to the number of slot types. Furthermore, they operate on fixed domain ontology and the neural architecture is heavily engineered and conceptually complex. However, our model produced high turn request and joint goal accuracies in comparison to the aforementioned baselines by 0.13% and 1.67%, respectively. Our model performed well because it can learn better contextual and semantic features as well as slot types, and possible values defined in advance and do not change dynamically. Furthermore, we convert the negative examples into positive examples to balance the datasets by random oversampling that decreases the errors. Additionally, the model does not require feature engineering such as lexicon and delexicalization due to the deployed Stacked BiLSTM and multiple attention mechanism. Another strength of the model is that it does not depend upon ontology size increase, given that the vocabulary does not change. Thus, the model is robust in terms of less computation

as a consequence of the BiLSTM and multiple attention instead of complex architecture with feature engineering.

Table 4. Joint Goal Accuracy and Turn Request Accuracy on DSTC-2 and WoZ-2.0. † is used for reproduced results from official codes. ‡ is used for the reported results in respective papers.

Models	DSTC-2		WoZ-2.0	
	Turn Request	Joint Goal	Turn Request	Joint Goal
NBT †	97.50	73.40	91.60	84.20
GLAD †	-	74.50	97.10	88.10
Statenet ‡	-	75.50	-	88.90
GSAT †	96.50	84.81	96.74	90.48
COMER ‡	-	-	97.10	88.60
Full Bert †	-	74.50	97.70	90.40
TEN †	-	77.30	97.10	90.80
SUMBT ‡	-	-	97.10	91.00
Seq2Seq-DU ‡	85.00	74.50	-	91.2
AG-DST ‡	-	-	-	91.37
BLA	**99.56**	**87.31**	**97.83**	**93.73**

5.2. Ablation Study

This section discusses the variants of neural networks to show the effectiveness of models on DSTC-2 and WoZ-2.0 datasets in Table 5 to compare the results with the proposed BLA model. We compose BSL with the last hidden state of BERT for input to SBL and then concatenate the outputs of BERT and SBL. It decreases the joint goal accuracies on both datasets as well as the turn requests on the DSTC-2. In contrast, SBL has no effect on both datasets regarding turn goal accuracies. However, the turn request accuracy on WoZ-2.0 dataset produces the highest results among all models. Later we used the combination of input to Stacked BiGRU (SBG) from the last hidden state of BERT, and then BERT and SBG output concatenation to formalize BSG. It produced the same results on WoZ-2.0 dataset for turn goal accuracy but improves the turn goal accuracy as higher than all models. However, the accuracies for both datasets' joint goal and DSTC-2 turn goal decreased.

Next, we used the last hidden state of BERT for SBL, then the hidden state of SBL to SBG to concatenate the outputs of BERT and SBG to design BLG. It produced the highest turn goal on DSTC-2; however, it lowered the accuracies of turn request and joint goal on DSTC-2, and turn goal, turn request, and joint goal on WoZ-2.0. In BLC, we used the last hidden state of BERT to give input in SBL, and then all the hidden states of SBL for the conditional random field (CRF) to concatenate outputs of BERT and CRF. Next, we deployed BLCA similar to BLC with slightly difference in addition of multiple attentions over the overall hidden states of SBL and then the concatenation of BERT, CRF, and multiple attentions. We observed that the performance on all accuracies across both datasets poorly decreased for BLC and BLCA. The second last model BCLA in Table 5 produced insignificant results upon all accuracies than other models on WoZ-2.0 datasets and decreased all accuracies for DSTC-2 as compared to the proposed model. BCLA is composed with the last hidden state of BERT as input to CNN, then the output of CNN embeds to SBL. Later, self-attention applies to the overall hidden states of SBL. Finally, BERT, SBL, and self-attention outputs were concatenated.

- **Effect of Stacked BiLSTM Network** We used variants of neural networks to investigate the performance of BLA in the ablation study. We replaced Stacked BiLSTM with Stacked BiGRU and run the model on Dev and Test parts of the WoZ-2.0 datasets. It clearly indicates that the joint goal accuracy on Dev, and joint goal, turn goal, and turn request accuracies on the Test part improve and utilize more accurate representations regarding SBL.
- **Effect of Attention Mechanism** To investigate the effect of attention, we conduct multiple attention and self-attention variants with different models. In Table 5, we

remove multiple attention from BERT and SBL that lessens the performance on joint goal accuracy on Dev, and on turn request and joint goal accuracies on Test in respect of BLA. Later, multiple attention eliminates from CRF which indicates the insignificant results over turn goal and joint goal Dev accuracies, and upon all three accuracies on the Test dataset. We use self-attention instead of multiple attention in BCLA, which generates the poorest results among all models. It can be perceived clearly from the ablation study that multiple attention enhances the performance of all models wherever in practice.

Table 5. Performance Evaluation on DSTC-2 and WoZ-2.0 Datasets.

Models	DSTC-2 Accuracies			WoZ-2.0 Accuracies		
	Joint Goal	Turn Request	Turn Goal	Joint Goal	Turn Request	Turn Goal
BSL	86.95	99.33	99.11	92.48	**98.19**	95.54
BSG	86.55	99.38	99.16	92.65	97.83	**95.66**
BLG	87.13	99.31	**99.18**	92.16	97.59	94.57
BLC	81.74	98.55	96.92	85.66	96.86	92.28
BLCA	82.96	98.93	97.3	88.07	97.22	91.73
BCLA	87.01	99.03	97.48	37	74.11	72.23
BLA	**87.31**	**99.56**	99.11	**93.73**	97.83	95.54

5.3. Hyper-Parameters

In this section, the results produced by hyper-parameters are examined. The settings for hyper-parameters used in this research study are shown in Table 4. The most important parameters of the proposed architecture are learning rate, hidden states, Stacked BiLSTM, and dropout rate. We have used a learning rate from 2.5×10^{-5} to 0.00005 as cited in Figure 5 to optimize the model. It indicates that the model produces poor results on a learning rate of 0.00005. Therefore, it is evident that the model is unable to converge on high learning rates. In contrast, the model produced joint goal accuracy of more than 87.20% and 93.70% on a low learning rate of 2×10^{-5} for DSTC-2 and WoZ-2.0, respectively, which are the highest among other learning rates. Similarly, the results on DSTC-2 for turn request accuracy are higher on 2×10^{-5} except the results for turn request on WoZ-2.0 are increased only by 0.02%. Thus, it is better to elect an adequate small learning rate of 2×10^{-5} in BLA.

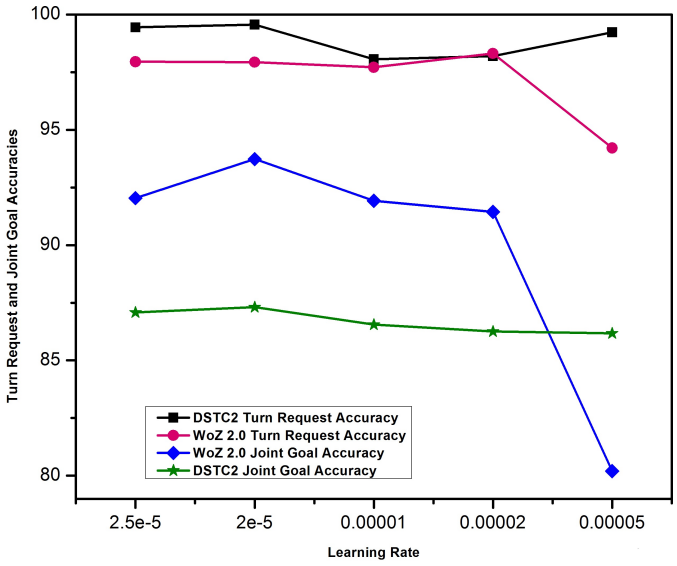

Figure 5. Impact of Learning Rate.

Furthermore, we used the different numbers of layers of Stacked BiLSTM, i.e., 2, 3, 4, 5, and 6 to find the best suitable number of layers for BLA. We noticed that when the neural network increased in depth, it reduced the results of the model as cited in Figure 6. When the number of layers is more than two, the results for joint goal accuracy are produced at less than 93.73% for WoZ-2.0 dataset and less than 87.31% for DSTC-2. Similarly, the performance of the model is decreased on turn request accuracy for the DSTC-2 dataset. However, the performance on the WoZ-2.0 dataset increased by 0.02% for turn request accuracy after setting up the number of layers to 4 and 6. Thus, it is perceived from the experiments that keeping the number of layers smaller is sufficient. We attained better results after setting up the two layers for the Stacked BiLSTM layers module of the model.

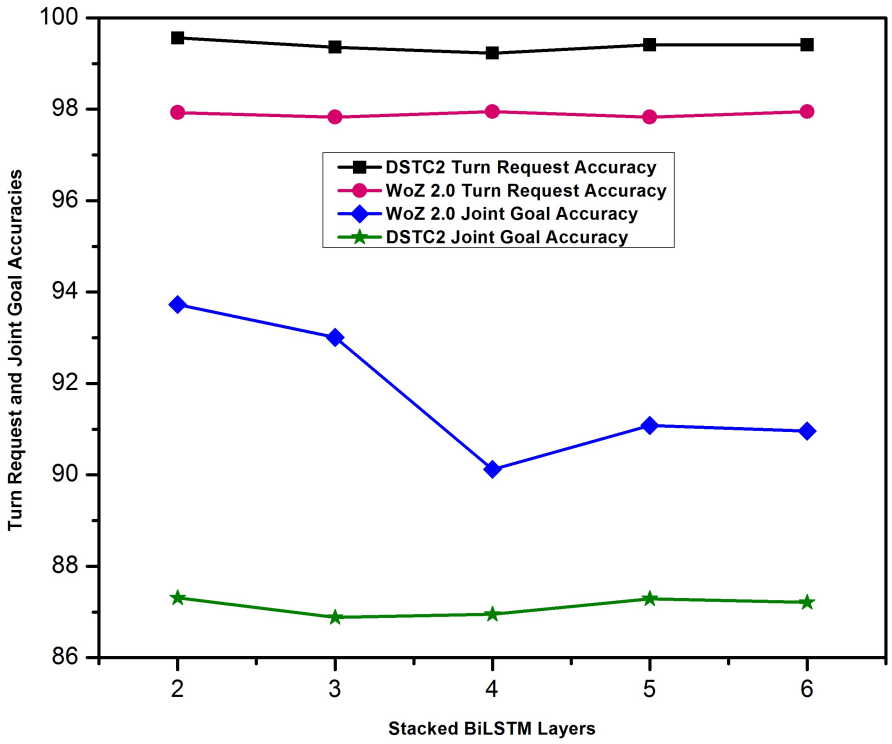

Figure 6. Impact of Stacked BiLSTM.

We used the dropout layer as cited in Figure 7 to analyze the effect on neural network regularization. We observed that the model significantly improved the results between 0.2 and 0.4. On the other hand, when the dropout rate is first decreased and then increased, the model performed poorly among the joint goal accuracies and turn request accuracies on both datasets. The model produced the best results when the dropout is set up to 0.25. We picked values that are beneficial for the model to produce better results upon joint goal accuracy. It demonstrated that Stacked BiLSTM layers and learning rate have a significant impact on the model during the learning of features. We also observed that when the learning rate is larger, i.e., 0.001, 0.01, and 0.1, the model was unable to produce joint goal accuracy and remained at 0%. Meanwhile, the turn request accuracies are less than 70% for both datasets on a large learning rate.

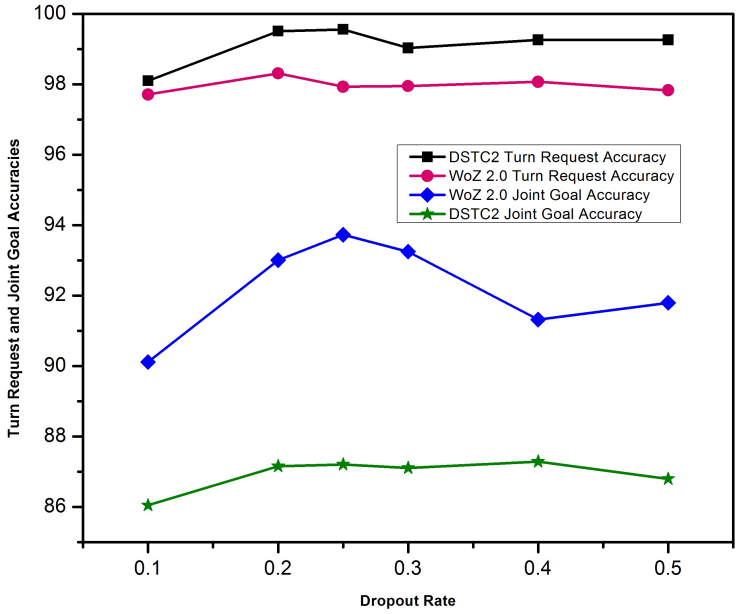

Figure 7. Impact of Dropout Rate.

6. Conclusions and Future Work

In this study, we have proposed a new framework BLA to find the accurate states for dialogue state tracking. The model is composed of BERT, Stacked BiLSTM, and multiple attention mechanisms to encode the dialogue context and candidate for feature extraction from utterances of the user and system to predict the dialogue state. Additionally, BLA allows parameters to share across the slots and slot-specific feature learning. BLA improves model performance by minimizing the mistakes from examples loaded from datasets after the conversion of negative examples into positive examples. To evaluate the performance of the model, we conducted experiments on two publicly available datasets WoZ-2.0 and DSTC-2, and analyzed the experiments with variants of the proposed model and baseline model. The result shows that BLA achieves state-of-the-art results on both datasets. The main findings and observations of the study include the following:

- The proposed model, upon evaluation on two datasets, revealed that it extracts features more effectively due to the use of Stacked BiLSTM as compared to other DL-based and traditional approaches.
- We experimented with variants of the attention mechanism. The weaker performance of the model on the self-attention layer suggests that the multi-attention mechanism is helpful for understanding the features and phrases for dialogue state tracking.
- During hyper-parameter selection, we noticed that the learning rate was the most important parameter for the proposed model because the model regulates the number of allocated errors for updating weights to perfectly calibrate the accuracies.

Besides these findings, the proposed framework bears some limitations. These include the need to be more flexible, scalable, and simple in order to work faster on multi-domain datasets. The parameter sharing can be further developed to render it more efficient for multi-domain datasets. Thus, we intend to incorporate a graph attention network to build the relationship between slots and their values that can enhance the performance of BLA. The decrement of parameters with the usage of BERT variants such as RoBERTa, DistilBERT, and ELECTRA may increase the robustness in DST, which is also under consideration. We plan to extend our model and conduct experiments on publicly available multi-domain datasets in the near future.

Author Contributions: M.A.K. contributed to conceptualization, methodology, software, data curation, writing original draft. Y.H. was involved in resources and analysis with constructive discussions. J.F. supervised and conceptualized the study. B.K.P. contributed to software and methodology. Z.A. was involved in analysis and manuscript preparation. I.U. helped with conceptualization, methodology, writing—review and editing. P.K. contributed to formal analysis, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The research presented in this paper was partially supported by the Natural Science Foundation Program of China under grant number U21A20488.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this research can be found in [34]. Readers may request the code of the proposed model from the corresponding authors.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Abro, W.A.; Qi, G.; Gao, H.; Khan, M.A.; Ali, Z. Multi-turn intent determination for goal-oriented dialogue systems. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
2. Abro, W.A.; Qi, G.; Aamir, M.; Ali, Z. Joint intent detection and slot filling using weighted finite state transducer and BERT. *Appl. Intell.* **2022**, *52*, 17356–17370. [CrossRef]
3. Young, S.; Gašić, M.; Thomson, B.; Williams, J.D. Pomdp-based statistical spoken dialog systems: A review. *Proc. IEEE* **2013**, *101*, 1160–1179. [CrossRef]
4. Abro, W.A.; Aicher, A.; Rach, N.; Ultes, S.; Minker, W.; Qi, G. Natural language understanding for argumentative dialogue systems in the opinion building domain. *Knowl.-Based Syst.* **2022**, *242*, 108318. [CrossRef]
5. Chen, H.; Liu, X.; Yin, D.; Tang, J. A survey on dialogue systems: Recent advances and new frontiers. *ACM Sigkdd Explor. Newsl.* **2017**, *19*, 25–35. [CrossRef]
6. Xiang, L.; Zhao, Y.; Zhu, J.; Zhou, Y.; Zong, C. Zero-shot language extension for dialogue state tracking via pre-trained models and multi-auxiliary-tasks fine-tuning. *Knowl.-Based Syst.* **2023**, *259*, 110015. [CrossRef]
7. Hong, T.; Cho, J.; Yu, H.; Ko, Y.; Seo, J. Knowledge-grounded dialogue modelling with dialogue-state tracking, domain tracking, and entity extraction. *Comput. Speech Lang.* **2023**, *78*, 101460. [CrossRef]
8. Zhong, V.; Xiong, C.; Socher, R. Global-Locally Self-Attentive Encoder for Dialogue State Tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 1458–1467. [CrossRef]
9. Liao, L.; Long, L.H.; Ma, Y.; Lei, W.; Chua, T.S. Dialogue state tracking with incremental reasoning. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 557–569. [CrossRef]
10. Lee, H.; Jeong, O. A Knowledge-Grounded Task-Oriented Dialogue System with Hierarchical Structure for Enhancing Knowledge Selection. *Sensors* **2023**, *23*, 685. [CrossRef]
11. Ye, F.; Manotumruksa, J.; Zhang, Q.; Li, S.; Yilmaz, E. Slot self-attentive dialogue state tracking. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 1598–1608.
12. Mrkšić, N.; Ó Séaghdha, D.; Wen, T.H.; Thomson, B.; Young, S. Neural Belief Tracker: Data-Driven Dialogue State Tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 1777–1788. [CrossRef]
13. Eric, M.; Goel, R.; Paul, S.; Sethi, A.; Agarwal, S.; Gao, S.; Kumar, A.; Goyal, A.; Ku, P.; Hakkani-Tur, D. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines. In *Proceedings of the 12th Language Resources and Evaluation Conference*; European Language Resources Association: Marseille, France, 2020; pp. 422–428.
14. Mao, M.; Liu, J.; Zhou, J.; Wu, H. Efficient Dialogue State Tracking by Masked Hierarchical Transformer *arXiv* **2021**, arXiv:2106.14433.
15. Kim, S.; Yang, S.; Kim, G.; Lee, S.W. Efficient Dialogue State Tracking by Selectively Overwriting Memory. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 567–582.
16. Nouri, E.; Hosseini-Asl, E. Toward scalable neural dialogue state tracking model. In Proceedings of the In 2nd Conversational AI workshop on NeurIPS, Montréal, QC, Canada, 7 December 2018.
17. Wen, T.H.; Vandyke, D.; Mrkšić, N.; Gašić, M.; Rojas-Barahona, L.M.; Su, P.H.; Ultes, S.; Young, S. A Network-based End-to-End Trainable Task-oriented Dialogue System. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*; Association for Computational Linguistics: Valencia, Spain, 2017; pp. 438–449.

18. Henderson, M.; Thomson, B.; Williams, J.D. The second dialog state tracking challenge. In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), Philadelphia, PA, USA, 18–20 June 2014; pp. 263–272.
19. Ren, L.; Xie, K.; Chen, L.; Yu, K. Towards universal dialogue state tracking. *arXiv* **2018**, arXiv:1810.09587.
20. Mrkšić, N.; Vulić, I. Fully statistical neural belief tracking. *arXiv* **2018**, arXiv:1805.11350.
21. Lai, T.M.; Tran, Q.H.; Bui, T.; Kihara, D. A simple but effective bert model for dialog state tracking on resource-limited systems. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Online, 4–8 May 2020; pp. 8034–8038.
22. Wu, C.S.; Madotto, A.; Hosseini-Asl, E.; Xiong, C.; Socher, R.; Fung, P. Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv* **2019**, arXiv:1905.08743.
23. Zhu, S.; Li, J.; Chen, L.; Yu, K. Efficient Context and Schema Fusion Networks for Multi-Domain Dialogue State Tracking. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Stroudsburg, PA, USA, 16–20 November 2020. [CrossRef]
24. Gao, S.; Sethi, A.; Agarwal, S.; Chung, T.; Hakkani-Tür, D.Z. Dialog State Tracking: A Neural Reading Comprehension Approach. In Proceedings of the 20th Annual SIGDIAL Meeting on Discourse and Dialogue, Stockholm, Sweden, 11–13 September 2019; pp. 1876–1885.
25. Ren, L.; Ni, J.; McAuley, J. Scalable and Accurate Dialogue State Tracking via Hierarchical Sequence Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 1876–1885. [CrossRef]
26. Le, H.; Socher, R.; Hoi, S.C. Non-Autoregressive Dialog State Tracking. In Proceedings of the International Conference on Learning Representations, Virtual, 26 April–1 May 2020; pp. 199–203.
27. Zhou, H.; Iacobacci, I.; Minervini, P. XQA-DST: Multi-Domain and Multi-Lingual Dialogue State Tracking. *arXiv* **2022**, arXiv:2204.05895.
28. Wang, Y.; He, T.; Mei, J.; Fan, R.; Tu, X. A Stack-Propagation Framework With Slot Filling for Multi-Domain Dialogue State Tracking. *IEEE Trans. Neural Networks Learn. Syst.* **2022**, 1–15. [CrossRef]
29. Liu, B.; Lane, I. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. In Proceedings of the Interspeech 2017, Stockholm, Sweden, 20–24 August 2017; pp. 2506–2510.
30. Henderson, M.; Thomson, B.; Young, S. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In Proceedings of the 2014 IEEE Spoken Language Technology Workshop (SLT), Lake Tahoe, NV, USA, 7–10 December 2014; pp. 360–365.
31. Henderson, M.; Thomson, B.; Young, S. Word-based dialog state tracking with recurrent neural networks. In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), Philadelphia, PA, USA, 18–20 June 2014; pp. 292–299.
32. Mrkšić, N.; Ó Séaghdha, D.; Thomson, B.; Gašić, M.; Su, P.H.; Vandyke, D.; Wen, T.H.; Young, S. Multi-domain Dialog State Tracking using Recurrent Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*; Association for Computational Linguistics: Beijing, China, 2015; pp. 794–799. [CrossRef]
33. Ramadan, O.; Budzianowski, P.; Gašić, M. Large-Scale Multi-Domain Belief Tracking with Knowledge Sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 432–437. [CrossRef]
34. Chen, J.; Zhang, R.; Mao, Y.; Xu, J. Neural Dialogue State Tracking with Temporally Expressive Networks. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP*; Association for Computational Linguistics: Seoul, Korea, 2020; pp. 1570–1579.
35. Budzianowski, P.; Wen, T.H.; Tseng, B.H.; Casanueva, I.; Ultes, S.; Ramadan, O.; Gašić, M. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 5016–5026. [CrossRef]
36. Sun, K.; Chen, L.; Zhu, S.; Yu, K. The SJTU system for dialog state tracking challenge 2. In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), Philadelphia, PA, USA, 18–20 June 2014; pp. 318–326.
37. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [CrossRef]
38. Graves, A.; Jaitly, N.; Mohamed, A. Hybrid speech recognition with Deep Bidirectional LSTM. In Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–12 December 2013; pp. 273–278. [CrossRef]
39. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237. [CrossRef]
40. Williams, J.D. Challenges and opportunities for state tracking in statistical spoken dialog systems: Results from two public deployments. *IEEE J. Sel. Top. Signal Process.* **2012**, *6*, 959–970. [CrossRef]

41. Serban, I.V.; Lowe, R.; Henderson, P.; Charlin, L.; Pineau, J. A Survey of Available Corpora for Building Data-Driven Dialogue Systems. *arXiv* **2015**, arXiv:1512.05742.
42. Saka, A.B.; Oyedele, L.O.; Akanbi, L.A.; Ganiyu, S.A.; Chan, D.W.; Bello, S.A. Conversational artificial intelligence in the AEC industry: A review of present status, challenges and opportunities. *Adv. Eng. Inf.* **2023**, *55*, 101869. [CrossRef]
43. Wang, Z.; Lemon, O. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In Proceedings of the SIGDIAL 2013 Conference, Metz, France, 22–24 August 2013; pp. 423–432.
44. Henderson, J.; Lemon, O. Mixture Model POMDPs for Efficient Handling of Uncertainty in Dialogue Management. In *Proceedings of the ACL-08: HLT, Short Papers*; Association for Computational Linguistics: Columbus, OH, USA, 2008; pp. 73–76.
45. Williams, J.D.; Young, S. Scaling POMDPs for spoken dialog management. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *15*, 2116–2129. [CrossRef]
46. Williams, J.D.; Young, S. Partially observable Markov decision processes for spoken dialog systems. *Comput. Speech Lang.* **2007**, *21*, 393–422. [CrossRef]
47. Henderson, M. Machine learning for dialog state tracking: A review. In Proceedings of the NIPS 2015 Workshop on Machine Learning for Spoken Language Understanding and Interaction, Montreal, QC, Canada, 11 December 2015.
48. Lee, S.; Eskenazi, M. Exploiting Machine-Transcribed Dialog Corpus to Improve Multiple Dialog States Tracking Methods. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*; Association for Computational Linguistics: Seoul, Republic of Korea, 2012; pp. 189–196.
49. Yu, K.; Sun, K.; Chen, L.; Zhu, S. Constrained markov bayesian polynomial for efficient dialogue state tracking. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 2177–2188.
50. Perez, J.; Liu, F. Dialog state tracking, a machine reading approach using Memory Network. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*; Association for Computational Linguistics: Valencia, Spain, 2017; pp. 305–314.
51. Metallinou, A.; Bohus, D.; Williams, J.D. Discriminative state tracking for spoken dialog systems. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, 4–9 August 2013; pp. 466–475.
52. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
53. Li, C.; Chrysostomou, D.; Pinto, D.; Hansen, A.K.; Bøgh, S.; Madsen, O. Hey Max, can you help me? An Intuitive Virtual Assistant for Industrial Robots. *Appl. Sci.* **2023**, *13*, 205. [CrossRef]
54. Ali, Z.; Kefalas, P.; Muhammad, K.; Ali, B.; Imran, M. Deep learning in citation recommendation models survey. *Expert Syst. Appl.* **2020**, *162*, 113790. [CrossRef]
55. Lee, H.; Lee, J.; Kim, T.Y. SUMBT: Slot-Utterance Matching for Universal and Scalable Belief Tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Florence, Italy, 2019; pp. 5478–5483. [CrossRef]
56. Sun, Z.; Huang, Z.; Ding, N. On Tracking Dialogue State by Inheriting Slot Values in Mentioned Slot Pools. *arXiv* **2022**, arXiv:2202.07156.
57. Jin, X.; Lei, W.; Ren, Z.; Chen, H.; Liang, S.; Zhao, Y.; Yin, D. Explicit state tracking with semi-supervision for neural dialogue generation. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 1403–1412.
58. Zhang, J.; Hashimoto, K.; Wu, C.S.; Wang, Y.; Yu, P.; Socher, R.; Xiong, C. Find or Classify? Dual Strategy for Slot-Value Predictions on Multi-Domain Dialog State Tracking. In *Proceedings of the 9th Joint Conference on Lexical and Computational Semantics*; Association for Computational Linguistics: Barcelona, Spain, 2020; pp. 154–167.
59. Henderson, M.; Thomson, B.; Young, S. Deep neural network approach for the dialog state tracking challenge. In Proceedings of the SIGDIAL 2013 Conference, Metz, France, 22–24 August 2013; pp. 467–471.
60. Kumar, A.; Ku, P.; Goyal, A.; Metallinou, A.; Hakkani-Tur, D. Ma-dst: Multi-attention-based scalable dialog state tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 3, pp. 8107–8114.
61. Feng, Y.; Wang, Y.; Li, H. A Sequence-to-Sequence Approach to Dialogue State Tracking. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, 1– 6 August 2021; pp. 1714–1725. [CrossRef]
62. Heck, M.; Lubis, N.; Niekerk, C.v.; Feng, S.; Geishauser, C.; Lin, H.C.; Gašić, M. Robust Dialogue State Tracking with Weak Supervision and Sparse Data. *Trans. Assoc. Comput. Linguist.* **2022**, *10*, 1175–1192. [CrossRef]
63. Williams, J.D. Web-style ranking and SLU combination for dialog state tracking. In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), Philadelphia, PA, USA, 18–20 June 2014; pp. 282–291.
64. Sebt, M.V.; Ghasemi, S.; Mehrkian, S. Predicting the number of customer transactions using stacked LSTM recurrent neural networks. *Soc. Netw. Anal. Min.* **2021**, *11*, 86. [CrossRef]
65. Cui, Z.; Ke, R.; Pu, Z.; Wang, Y. Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. *Transp. Res. Part Emerg. Technol.* **2020**, *118*, 102674. [CrossRef]

66. Tian, X.; Huang, L.; Lin, Y.; Bao, S.; He, H.; Yang, Y.; Wu, H.; Wang, F.; Sun, S. Amendable Generation for Dialogue State Tracking. In Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI, Online, 10 November 2021; pp. 80–92.
67. Shan, Y.; Li, Z.; Zhang, J.; Meng, F.; Feng, Y.; Niu, C.; Zhou, J. A contextual hierarchical attention network with adaptive objective for dialogue state tracking. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Virtual, 5–10 July 2020; pp. 6322–6333.
68. Ye, F.; Feng, Y.; Yilmaz, E. ASSIST: Towards Label Noise-Robust Dialogue State Tracking. In *Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022*; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 2719–2731. [CrossRef]
69. Hu, Y.; Lee, C.H.; Xie, T.; Yu, T.; Smith, N.A.; Ostendorf, M. In-Context Learning for Few-Shot Dialogue State Tracking. *arXiv* **2022**, arXiv:2203.08568.
70. Uddin, M.N.; Li, B.; Ali, Z.; Kefalas, P.; Khan, I.; Zada, I. Software defect prediction employing BiLSTM and BERT-based semantic feature. *Soft Comput.* **2022**, *26*, 7877–7891. [CrossRef]
71. Abro, W.A.; Qi, G.; Ali, Z.; Feng, Y.; Aamir, M. Multi-turn intent determination and slot filling with neural networks and regular expressions. *Knowl.-Based Syst.* **2020**, *208*, 106428. [CrossRef]
72. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef]
73. Sundermeyer, M.; Schlüter, R.; Ney, H. LSTM neural networks for language modeling. In Proceedings of the 13th Annual Conference of the International Speech Communication Association, Portland, OR, USA, 9–13 September 2012.
74. Wen, T.H.; Gašić, M.; Mrkšić, N.; Rojas-Barahona, L.M.; Su, P.H.; Ultes, S.; Vandyke, D.; Young, S. Conditional Generation and Snapshot Learning in Neural Dialogue Systems. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Austin, TX, USA, 2016; pp. 2153–2162. [CrossRef]
75. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
76. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
77. Arefeen, M.A.; Nimi, S.T.; Rahman, M.S. Neural network-based undersampling techniques. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *52*, 1111–1120. [CrossRef]
78. Zhu, C.; Zeng, M.; Huang, X. SIM: A Slot-Independent Neural Model for Dialogue State Tracking. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*; Association for Computational Linguistics: Stockholm, Sweden, 2019; pp. 40–45. [CrossRef]
79. Sharma, S.; Choubey, P.K.; Huang, R. Improving Dialogue State Tracking by Discerning the Relevant Context. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 576–581. [CrossRef]
80. Rastogi, A.; Zang, X.; Sunkara, S.; Gupta, R.; Khaitan, P. Schema-guided dialogue state tracking task at dstc8. *arXiv* **2020**, arXiv:2002.01359.
81. Dey, S.; Kummara, R.; Desarkar, M. Towards Fair Evaluation of Dialogue State Tracking by Flexible Incorporation of Turn-level Performances. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 318–324. [CrossRef]
82. Balaraman, V.; Magnini, B. Scalable neural dialogue state tracking. In Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Sentosa, Singapore, 14–18 December 2019; pp. 830–837.