Neural Computing and Applications (2024) 36:16331–16344
https://doi.org/10.1007/s00521-024-09927-0

ORIGINAL ARTICLE

Leveraging intent–entity relationships to enhance semantic accuracy in NLU models

Romina Soledad Albornoz-De Luise[1] · Miguel Arevalillo-Herráez[1] · Yuyan Wu[1]

Received: 21 July 2023 / Accepted: 28 April 2024 / Published online: 26 May 2024

Abstract
Natural Language Understanding (NLU) components are used in Dialog Systems (DS) to perform intent detection and entity extraction. In this work, we introduce a technique that exploits the inherent relationships between intents and entities to enhance the performance of NLU systems. The proposed method involves the utilization of a carefully crafted set of rules that formally express these relationships. By utilizing these rules, we effectively address inconsistencies within the NLU output, leading to improved accuracy and reliability. We implemented the proposed method using the Rasa framework as an NLU component and used our own conversational dataset AWPS to evaluate the improvement. Then, we validated the results in other three commonly used datasets: ATIS, SNIPS, and NLU-Benchmark. The experimental results show that the proposed method has a positive impact on the semantic accuracy metric, reaching an improvement of 12.6% in AWPS when training with a small amount of data. Furthermore, the practical application of the proposed method can easily be extended to other Task-Oriented Dialog Systems (T-ODS) to boost their performance and enhance user satisfaction.

Keywords Natural language understanding · Semantic accuracy · Task-oriented dialog systems · Conversational agents · Intent detection · Slot filling

1 Introduction

As Dialog Systems (DS) gain popularity, it becomes increasingly important to enhance their performance. Task-Oriented Dialog Systems (T-ODS) are a kind of DS that aim to assist users in completing specific tasks across diverse domains [1], such as e-commerce, help desk or customer care, website navigation, personalized service, and training or education [2, 3]. These systems play a valuable role in real-world scenarios, facilitating tasks like restaurant booking, weather queries, flight booking, traffic information, technical problem solving, and providing access to educational material, among other functionalities.

In T-ODS, more structured conversations and advanced reasoning abilities are required to dynamically generate knowledge presented to the user. The ability to accurately understand users' requirements is a crucial aspect of these systems. This involves using Natural Language Understanding (NLU) techniques to understand and extract relevant information from text, enabling effective communication with humans.

Intent detection and entity extraction are two crucial tasks for NLU systems. Intent detection involves determining the purpose or goal behind a user's input, while entity extraction (also known as slot filling) involves extracting and organizing specific pieces of information into predefined entity categories or slots. Traditionally, these tasks have been handled independently [4]. However, they suffered from error propagation. To address this problem, researchers began to solve the two tasks jointly [5–7]. Results showed that there is a strong interdependency between the intent and the entities, as the specific

✉ Romina Soledad Albornoz-De Luise
romina.albornoz@uv.es

Miguel Arevalillo-Herráez
miguel.arevalillo@uv.es

Yuyan Wu
yuyan.wu@uv.es

[1] Departament d'Informática, Universitat de València, Avinguda de la Universitat s/n, 46100 Burjassot, Valencia, Spain

entities present within a sentence are greatly influenced by the underlying intent of that sentence. Other recent studies have focused on enhancing NLU performance by improving the internal architecture [8], deploying new models for different domains and varying amounts of labeled data [9–12] and refining prediction stages by, e.g., detecting examples that have been misclassified, handling out-of-distribution examples [13], and reducing the error rate [14, 15].

To assess the performance of an NLU component, Exact Match Accuracy (EMA) is frequently employed, which measures the proportion of sentences in which both the intent and all entities are predicted correctly. A low EMA means an inaccurate identification of the entities and intent conveyed in the user's message. This, in turn, can lead to misunderstandings and incorrect responses, negatively impacting the user's experience and impairing the system's ability to perform its intended tasks. To boost performance and enhance user satisfaction, these systems can incorporate an extra validation check to ensure the accuracy of the message and generate an appropriate response. This validation step prevents the system from solely relying on the output of the NLU component.

In this work, our contribution is the development of a post-processing technique designed to enhance the EMA of NLU components by using a new paradigm that leverages knowledge injected by domain experts. This technique identifies inconsistencies in the system's predictions and rectifies them by searching for the most probable intent that aligns with the detected entities. This is done by combining the ranking of intents returned by the NLU system with a set of manually crafted rules. The proposed approach can be seamlessly used with any existing model that simultaneously predicts a ranking of intents and a set of entities. Its effectiveness has been evaluated by using the Rasa toolkit to deploy a representative example of an NLU component, taking advantage of its availability as an open-source tool. The validation of the proposed approach relies mainly on the AWPS dataset, due to its unique attributes and relevance to our research. However, our method has also been evaluated using other widely employed datasets for NLU in T-ODS, including ATIS, SNIPS, and NLU-Benchmark. The results indicated that implementing our method led to an increase in the EMA, demonstrating its effectiveness in improving the accuracy of these types of DS.

To provide a comprehensive understanding of our research, the remainder of the paper has been organized as follows. Section 2 provides background information on the problem and explains why this research is important and relevant. Section 3 presents the proposed method to check and attempt to correct any incorrect predictions and also provides a detailed explanation of its operation and purpose. Section 4 explains the datasets used in the experiments and describes the experimental setting employed to test the trained models. Section 5 provides an analysis of the results, including some high-level observations. Finally, Sect. 6 summarizes the main findings of the study and suggests possible directions for future research.

2 Background and related work

Our goal is to present a post-processing approach to enhance NLU prediction within T-ODS in the educational domain. Accordingly, the first subsection offers a general overview of T-ODS. Subsequently, we explore various approaches for enhancing NLU predictions, and finally, we describe an specific T-ODS in the educational domain, which is the main focus of our enhancement.

2.1 Task-oriented dialog systems

Task-oriented dialog systems are a branch of dialog systems that are designed to help users to accomplish a certain task such making restaurant reservations or providing assistance to users. Unlike open-domain DS or chat-bots that prioritize user engagement [16] and allow for more generic conversations such as chit-chat, these systems focus on achieving specific tasks within one or multiple domains [17]. They are typically built on top of a structured ontology, which defines the domain knowledge necessary for carrying out these tasks [18].

T-ODS have the potential to play a significant role in a wide range of domains, enhancing human interaction with technology. In educational contexts, intelligent tutoring systems have integrated this technology to help students learn by engaging in natural language conversations. For example, AutoTutor [19] is a pedagogical agent that simulates the dialog and strategies of a human tutor and uses natural language to interact with learners. Another example is the math problem-solving DS [20, 21] that was integrated into the ITS called Hypergraph-Based Problem Solver (HBPS) [22–24]. The commercial sector widely employs T-ODS for various purposes, such as assisting customers with purchase-related tasks with online shopping [25] or for airline ticket booking [26]. In the tourism domain, similar systems help users to complete a travel plan [27], while in healthcare T-ODS like Watson Health [28] support patient diagnosis and provide links to access medical literature. However, challenges remain in areas such as effective communication, efficiency, security, and privacy.

In general, T-ODS contain three main components, namely NLU, Dialog Manager (DM) and Natural Language Generation (NLG). The NLU component generates a

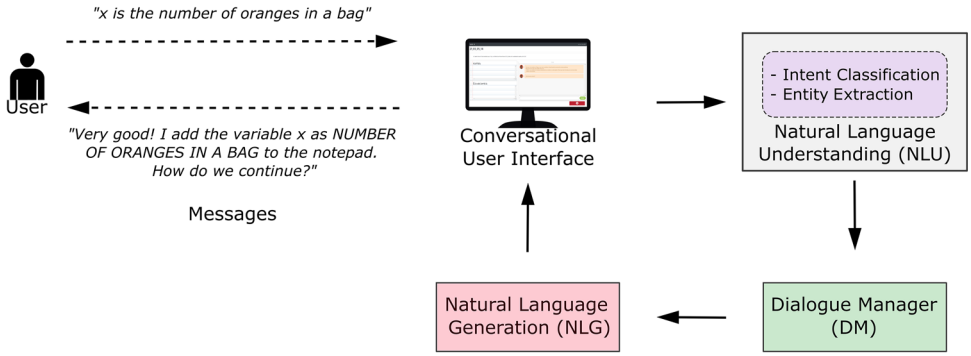


Fig. 1 The main components and workflow of a T-ODS

representation of the user's input which can be used by the other components of the system; the DM keeps track of the dialog history and runs dialog strategies; and the NLG module generates natural language responses based on the system's actions. The workflow of a T-ODS is illustrated in Fig. 1. As depicted, user inputs received through the GUI are sent to the NLU component to extract the intent and entities from the sentence. The extracted information is then used by other components to construct an appropriate response. It is important to note that errors in intent and entity detection can propagate to other components and impact the overall performance of the system.

Existing methods for developing NLU can be categorized into four classes: rule-based, statistical, machine learning techniques and hybrid methods [29, 30]. Rule-based approaches involve creating a set of handcrafted IF ... THEN rules, which are used to allow reasoning by inference. The preconditions of such rules may be triggered by the context or state of the dialog or the user's input via pattern matching [30–35]. Statistical methods base their processing on a mathematical analysis of the text corpora. Many T-ODS developed to date in this category rely on topic modeling and combine a Bag of Words representation with Latent Semantic Analysis (LSA) to analyze the conceptual similarity of the input utterance to a set of representative training inputs [36, 37]. Machine learning approaches, such as supervised learning and deep learning, have gained popularity in NLU. These techniques involve training models on annotated data to learn patterns and make predictions. In the context of NLU, these models automatically extract one or more possible interpretations from a single input [38, 39]. Moreover, the use of NLU services or frameworks, such as DialogFlow[1], Amazon Lex[2], IBM Watson[3], and Rasa[4], support the construction of these systems. Two common tasks performed by NLU services are intent detection and entity extraction [40].

[1] https://dialogflow.com/.

[2] https://aws.amazon.com/lex/.

[3] https://www.ibm.com/watson/.

[4] https://rasa.com/.

These methods are designed to accurately identify the intent and entities within sentences, enabling them to carry out specific tasks with high accuracy. The primary goal of this work is to enhance the exact match accuracy, which considers both tasks simultaneously, during a post-processing stage designed to be universally adaptable to any model.

2.2 Approaches for enhancing NLU prediction

Recent studies have focused on enhancing performance through the deployment of deep learning methods and hybrid approaches [41, 42]. Many neural network architectures have been applied for intent detection and entity extraction in various domains [27, 41, 43–47], using different amounts of labeled data [9, 11, 12]. For instance, [8] proposed a model that combines different neural network architectures with regular expression rules to encode domain knowledge. They also used a pre-trained language model to generate contextual representations of user sentences.

Also, some efforts have been made to refine different stages of the prediction process. One such stage involves detecting examples that are misclassified. For example, [13] used softmax prediction probabilities to detect whether an example is misclassified or belongs to an out-of-distribution category, which refers to a different distribution from the training data. In [14], authors proposed a selective classifier with a confidence estimator to address this problem. Their approach involves a simple error regularization trick that allows the classifier to abstain from making predictions on low-confidence examples, aiming to reduce the error rate and enhance selective prediction performance. In [15], the authors proposed training a confidence estimator that assigns higher scores to correctly predicted instances, by annotating a held-out dataset conditioned on the model's predictive correctness. The annotated dataset is then used to train a calibrator, which serves as the confidence estimator for selective prediction. These collective efforts aim to advance the performance and capabilities of NLU models, ultimately enabling more accurate and robust NLU. Our work seeks to build upon these advancements by

implementing a unique post-processing approach that accounts for the intricate relationships between intents and entities. This strategy diverges from the conventional approaches of design new models or utilizing and integrating different neural network architectures to enhance the various stages of the prediction process.

2.3 A particular T-ODS in the educational domain

In previous studies [20, 21, 48, 49], the Rasa open-source framework [50] was utilized to develop NLU models that interact with an existing intelligent tutoring system called HBPS [22–24]. This system focuses entirely on the translation stage of the algebraic/arithmetic word problem-solving process and adopts an unguided approach that does not impose any restrictions on the solution path. HBPS is able to check the validity of user entries, provide aids that are consistent with the student's reasoning, and simultaneously monitor multiple alternative solution paths.

The T-ODS in HBPS use the NLU Rasa service to extract the intent from the user utterances received through the Graphical User Interface (GUI). The prediction model's output is then forwarded to other components for further processing and used to construct an appropriate response [20]. Despite the high performance exhibited by the Dual Intent and Entity Transformer (DIET) [43] in intent detection and entity extraction, there is still potential for enhancing classification results. This improvement can be achieved by taking into careful consideration the inter-dependencies that exist between intents and entities. This analysis can help identifying invalid intent detections or reinforce decisions made by the NLU component.

Our goal is to develop a robust and effective approach to enhance the exact match accuracy of T-ODS. This involves systematically verifying the output of NLU prediction models and employing an intent corrector while considering the inter-dependencies between intents and entities. By incorporating this algorithm, we can prevent the system from solely relying on the NLU component's output and mitigate the propagation of errors to other system components. The adoption of this approach aims to promote a seamless and intuitive interaction experience for users, leading to improved overall system performance, effectiveness, and better communication between users and the system.

3 Proposed method

Let's assume a NLU system that considers a set of m intents and r entity types. Given an utterance $S = [s_1, s_2, \ldots, s_n]$, which represents a sequence of words or tokens of length n. Intent detection is defined as a classification task over utterances, where the system has to assign the correct intent label y_i^I from a set of predefined intent classes $y^I = \{y_1^I, y_2^I, ..., y_m^I\}$ to the whole utterance S. On the other hand, entity extraction can be considered a token-level sequence tagging problem, where the system has to assign a corresponding entity label $y_j^E \in \{y_1^E, y_2^E, \ldots, y_r^E\}$ to each token s_j of the utterance.

In most conversational systems, there is a strong correlation between the intent and the entities, which can be expressed in terms of the number of entities of each type that may appear with a particular intent. When a system encounters difficulties in comprehending a legitimate request, it may be due to inaccurate detection of intent, entity extraction, or both occurring simultaneously. If the system incorrectly identifies the user's intent, it may respond inappropriately or fail to execute the desired action. Similarly, errors in entity extraction can lead to misunderstandings about the specific details of the request. These inaccuracies can significantly impact the effectiveness of the conversational system, leading to user frustration and potential failure to meet their needs. Ensuring simultaneous accuracy in both intent detection and entity extraction is crucial for the efficient processing of requests and their correct fulfillment by conversational systems.

Figure 2 shows a typical output of an NLU component, which includes the user utterance, the identified entities, the predicted intent, and an intent ranking. The proposed method carefully examines this information to determine whether the identified intent and entities are compatible. If they are, the response is forwarded to the other components for further processing. However, if a discrepancy is detected, the method thoroughly iterates through the intent ranking to determine the most appropriate intent that aligns with the extracted entities. If none of the intents yield a correct prediction, the method returns a "nlu_fallback" response so that the conversational system can prompt the user for further clarification or adopt an alternative strategy to address the situation. In practice, we do not need to consider the entire ranking of intents as, in our experience, considering lengths above half the number of intents has rarely benefited the EMA. Instead, considering intents that were initially judged as highly unlikely implies a higher processing time and often leads to a wrong interpretation of the user message.

The proposed approach involves comparing a prediction against a list of valid combinations, by using the predicted intent along with the number of entities of each type that are present in the NLU response. To effectively make this comparison we use an internal vector representation, modeling both valid combinations and the content of an NLU response by using a vector $[p_0, p_1, \ldots, p_r]$ of size $r + 1$. This representation uses two dictionaries to encode

```
{
  "text": "U",
  "intent": {
    "name": "intent_1",
    "confidence": conf_1
  },
  "entities": [
    {
      "entity": "entity_1",
      "start": star_1,
      "end": end_1,
      "confidence_entity": conf_ent_1,
      "value": "value_1"
    },
    {
      "entity": "entity_2",
      "start": star_2,
      "end": end_2,
      "confidence_entity": conf_ent_2,
      "value": "value_2"
    },
    ...,
    {
      "entity": "entity_n",
      "start": star_n,
      "end": end_n,
      "confidence_entity": conf_ent_n,
      "value": "value_n"
    }
  ],
  "intent_ranking": [
    {
      "name": "intent_1",
      "confidence": conf_1
    },
    {
      "name": "intent_2",
      "confidence": conf_2
    },
    ...
    {
      "name": "intent_N",
      "confidence": conf_N
    }
  ]
}
```

Fig. 2 Output of the NLU component given an utterance U

intents and entities using numerical values. The intent dictionary, $D_I = \{y_1^I : 1, \ldots, y_m^I : m\}$, maps each of the m intents to a numerical value. Likewise, the entities dictionary, $D_E = \{y_1^E : 1, \ldots, y_r^E : r\}$, links each of the r predefined entities to a unique number.

At design time, all valid combinations are specified as a set of vectors $P = \{P_1, \ldots, P_n\}$. A single intent may be associated with multiple vectors in set P. Each vector P_i represents a potentially valid combination and encodes the number of instances for each entity type that could be associated with one particular intent. The first component of each vector P_i takes the value associated with the intent in the dictionary D_I. The subsequent components indicate the number of instances of each entity type that would make a response be judged as valid, arranged according to the sequence defined in the dictionary D_E. A -1 value can be used to specify that the number of entities of that type is irrelevant and does not affect the validity of the response.

At inference time, the NLU response is transformed into a response vector P_R in the same format as the vectors in P. Again, the first element of the vector P_R indicates the index of the intent as stored in D_I. The rest of the elements refer to the number of instances of each entity type that appear in the response, in the same order as in the vectors in P. The vector P_R is then compared against the vectors in the set P, and the response is considered consistent only if it matches at least one of the vectors in P. A vector P_R is considered to match a vector $P_i \in P$ if they store identical values in all positions, except for those marked with -1 in vector P_i.

To provide a more comprehensive understanding of the proposed method, we provide two illustrative examples within the context of the HBPS dialog system. Table 1 shows the most frequent intents in this T-ODS, along with how many instances of each entity type should appear when an utterance is classified under each intent. When several rows are specified for the same intent, the output is considered consistent as long as it matches one of them. The remaining 17 intents considered in this system are all expected to appear with no entities.

In HBPS, the dictionaries D_I for intents and D_E for entities are defined as specified as follows, where ellipses have been used for brevity reasons to replace some intents that do not accept entities of any kind.

$$
\begin{aligned}
D_I = \{\ & \text{``affirm''} : 1,\\
& \ldots,\\
& \text{``define letter''} : 5,\\
& \text{``define letter missing description''} : 6,\\
& \ldots,\\
& \text{``equation''} : 10,\\
& \text{``equivalence''} : 11,\\
& \text{``expression''} : 12,\\
& \text{``get letter meaning''} : 13,\\
& \ldots,\\
& \text{``number''} : 20,\\
& \ldots,\\
& \text{``quantity description''} : 22,\\
& \ldots,\\
& \text{``word operation''} : 25\},
\end{aligned}
$$

$$
\begin{aligned}
D_E = \{&\text{``description''} : 1,\\
&\text{``equation''} : 2,\\
&\text{``expression''} : 3,\\
&\text{``number''} : 4,\\
&\text{``variable''} : 5\}.
\end{aligned}
$$

Table 1 Intents with Entities: Details of potentially valid combinations of intents and entities in HBPS

Intent	Total samples	Entity				
		Description	Equation	Expression	Number	Letter
Define letter	1199	1	0	0	0	1
Define letter missing description	148	1	0	0	0	0
Equation	848	0	1	0	0	0
Equivalence		0	0	0	1	1
		0	0	1	1	0
	475	0	0	2	0	0
		0	0	0	0	2
		0	0	1	0	1
Expression	1 958	1	0	1	0	0
		0	0	1	0	0
Get letter meaning	107	0	0	0	0	1
Number	629	0	0	0	1	0
		1	0	0	1	0
Quantity description	664	1	0	0	0	0

These dictionaries are used to generate the set P of valid combinations of intents and entities, which is shown in Table 2.

Consider the scenario where the system interprets the user's utterance "x = age of Anna" as the intent "define letter," recognizing the entities [variable: x] and [description: age of Anna]. In this case, the response would be encoded as $P_R = [5, 1, 0, 0, 0, 1]$. The first component indicates that the predicted intent, "define letter," corresponds to index 5 in the intent dictionary. The second component and last components indicate that the output contains one entity of type "description" and another of type "letter." The remaining components represent the count of entities of types "equation," "expression," and "number," respectively, all of which are zero in this case. This vector matches vector P_5 in Table 2, therefore classifying the NLU output as valid.

As an example of an inconsistent scenario, let's consider a user who sends the message "Andrea is 9x." The system generates a vector representation for this input as $P_R = [5, 1, 0, 1, 0, 0]$, indicating that the system detects the intent "define letter", with the entities [description: Andrea] and [expression: 9x]. This encoded representation of the answer does not match any entry in P. Hence, we can infer that the system misidentified the user's input as falling under the "define letter" intent. To address this misclassification, the method replaces the intent encoding with that of the next one in the ranking. For the sake of our example, let's assume is "expression." This replacement results in an updated vector representation of $P_R = [12, 1, 0, 1, 0, 0]$. As this new vector successfully matches P_{16}, the method corrects the prediction by substituting the original intent, "define letter," with the intent "expression." This corrected prediction is then sent to the system for further processing.

We shall note that the proposed approach is only able to correct potential mistakes in the identification of the intent, but is not able to address potential errors in entity extraction. The entity list originally returned by the NLU system is always left untouched, meaning that the proposed method neither positively nor negatively affects metrics that focus exclusively on the extracted entities, such as the F_1 score for entity extraction. Rather, the method is designed to improve exact match accuracy by ensuring that the chosen intent aligns with the entities identified.

4 Experimental evaluation

4.1 Datasets

To evaluate the effectiveness of the proposed method, we conducted a series of experiments on our dataset AWPS, as well as on three other publicly available datasets, namely ATIS, SNIPS, and NLU-Benchmark. The major characteristics of all 4 datasets employed in the evaluation are described below.

4.1.1 AWPS

The Algebraic Word Problem Solving Dataset [20] contains annotations of intents and entities. It includes

Table 2 Set of vectors P, expressing admitted combinations of intent and entities

Vector index	Vector components
P_1	[1, 0, 0, 0, 0, 0]
P_2	[2, 0, 0, 0, 0, 0]
P_3	[3, 0, 0, 0, 0, 0]
P_4	[4, 0, 0, 0, 0, 0]
P_5	[5, 1, 0, 0, 0, 1]
P_6	[6, 1, 0, 0, 0, 0]
P_7	[7, 0, 0, 0, 0, 0]
P_8	[8, 0, 0, 0, 0, 0]
P_9	[9, 0, 0, 0, 0, 0]
P_{10}	[10, 0, 1, 0, 0, 0]
P_{11}	[11, 0, 0, 0, 1, 1]
P_{12}	[11, 0, 0, 1, 1, 0]
P_{13}	[11, 0, 0, 2, 0, 0]
P_{14}	[11, 0, 0, 0, 0, 2]
P_{15}	[11, 0, 0, 1, 0, 1]
P_{16}	[12, 1, 0, 1, 0, 0]
P_{17}	[12, 0, 0, 1, 0, 0]
P_{18}	[13, 0, 0, 0, 0, 1]
P_{19}	[14, 0, 0, 0, 0, 0]
P_{20}	[15, 0, 0, 0, 0, 0]
P_{21}	[16, 0, 0, 0, 0, 0]
P_{22}	[17, 0, 0, 0, 0, 0]
P_{23}	[18, 0, 0, 0, 0, 0]
P_{24}	[19, 0, 0, 0, 0, 0]
P_{25}	[20, 0, 0, 0, 1, 0]
P_{26}	[20, 1, 0, 0, 1, 0]
P_{27}	[21, 0, 0, 0, 0, 0]
P_{28}	[22, 1, 0, 0, 0, 0]
P_{29}	[23, 0, 0, 0, 0, 0]
P_{30}	[24, 0, 0, 0, 0, 0]
P_{31}	[25, 0, 0, 0, 0, 0]

examples like "y represents the number of basketball students" labeled with the intent "define letter", and the entities [variable: y] and [description: number of basketball students]. The dataset consists of 6 293 training and 1 973 test utterances that cover various resolution steps of algebraic math word problems. These steps include defining letters, equations, and expressions, as well as seeking help or guidance, and more. In total, the dataset includes 25 different intents and 5 different entity types.

4.1.2 ATIS

The ATIS [51] dataset comprises transcripts of audio recordings of people making flight reservations and has been extensively studied over the years. The dataset used for our experiments includes 4 978 training and 893 test utterances, each annotated with intents and entities. For example, the phrase "how many airports does oakland have" is labeled with the intent "atis_quantity" and the entity [city_name: oakland]. The ATIS training set contains 21 intents and 79 entities.

4.1.3 SNIPS

The SNIPS dataset is collected from the Snips personal voice assistant [52]. This dataset includes 13 784 training utterances and 700 test utterances. Like ATIS, the SNIPS dataset used is annotated with intents and entities. For example, the phrase "play music from 1996" is labeled with the intent "PlayMusic" and the entity [year: 1996]. It comprises 7 intents and 39 entities.

4.1.4 NLU-Benchmark dataset

The NLU-Benchmark dataset [53] consists of 25 716 utterances. This dataset has annotations for intents and entities. As an example, the phrase "is there any alarm after five am" is labeled with the intent "alarm_query" and the entity [time: five am]. This dataset consists of 10-folds, each with its own train and test sets. The train and test sets for each fold contain 9960 and 1076 utterances, respectively. Overall, there are 64 intents and 54 entity types present in the dataset.

4.2 Experiments on AWPS dataset

To replicate the performance level demonstrated in the study described in [20], we followed the same training approach outlined in that paper. Specifically, we used the "No Unigrams" pipeline and trained our models using Rasa version 3.6.0. During training, the ranking length parameter was set to 0, to ensure that the model considered all possible intents and provided accurate responses in various situations.

We created 10 different models using the training corpus $\mathcal{C}_1$, progressively increasing the train set with samples that were not included in the previous model. For each new model, 10% of the corpus data were added, until using the entire repository. We repeated this process 10 times (10-folds). The results reported are the ones obtained for the 1973 sentences contained in the test corpus $\mathcal{C}_2$, reserving 10% of the train data for validation in all cases.

4.3 Experiments on ATIS and SNIPS

To achieve a performance level similar to that demonstrated in the study by [43], we followed the training

approach detailed in their paper. Specifically, we used the "sparse*" pipeline and trained our models using Rasa version 1.8.0. For each dataset, we employed 10-folds. Each fold comprised its own train set of either 4978 or 13784 utterances, and a corresponding test set of either 893 or 700, respectively. Within each fold, we created 10 different models using the original training datasets, progressively adding 10% more train data for each new model until the entire repository was utilized. The newly added samples were not included in the previous model. Throughout the model-building process, care was taken to maintain a consistent distribution of utterances from each intent in every fold.

4.4 Experiments on NLU-Benchmark dataset

In our study, we attempted to replicate the highest level of performance shown in [53]. However, due to the unavailability of PolyAI's ConveRT models, we were unable to reproduce their original experiment. As an alternative, we focused on reproducing the results using the pipeline named as "sparse-GloVe-mask-loss" in their paper and trained our models using Rasa version 1.8.0. For each fold, we utilized a separate train set consisting of 9960 samples and a test set containing 1076 samples. In order to explore the model's performance across different amounts of training data, we generated 10 distinct models for each fold. With each new model, we gradually increased the size of the training data by 10% until we utilized the entire repository. It is important to note that the newly added samples were not included in the previous model to maintain consistency. Throughout the training process, we ensured that the distribution of utterances from each intent remained consistent across all folds. This approach allowed us to have a balanced representation of intents in each fold, ensuring that the models were trained and evaluated on similar intent distributions.

4.5 Evaluation metrics of NLU performance

Several metrics can be employed to evaluate the performance of different tasks in NLU models. Precision, recall, and F-measure are commonly used metrics that assess the quality of the model's predictions. For a given class C_i, let's denote the number of samples that were correctly classified as members of this class as TP_i (true positives); the number of samples that were incorrectly assigned to the class as FP_i (false positives); the number of correctly classified samples into a class other than C_i as TN_i (true negatives); and the number of samples of class C_i that were assigned to a different class as FN_i (false negatives). Assuming K different classes, the micro-average precision measure is defined as

$$Precision = \frac{\sum_{i=1}^{K} TP_i}{\sum_{i=1}^{K} TP_i + \sum_{i=1}^{K} FP_i}$$

and the micro-average recall measure is defined as

$$Recall = \frac{\sum_{i=1}^{K} TP_i}{\sum_{i=1}^{K} TP_i + \sum_{i=1}^{K} FN_i}$$

The micro-averaged F_1 score is a widely accepted evaluation measure that provides a good compromise between the two metrics and is computed as:

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

To comprehensively assess a model's performance in both intent detection and entity extraction tasks, the EMA metric is a commonly accepted metric. EMA is also known as overall accuracy or sentence-level semantic accuracy and measures the number of sentences where both the intent and all slots are predicted correctly, divided by the total number of sentences. It provides a comprehensive assessment of the NLU model's performance in both tasks simultaneously and hence evaluates the overall effectiveness of the model.

4.6 Designed rules

In order to define the rules for the AWPS dataset, we followed a manual approach that involved domain experts carefully crafting each rule. These experts conducted a detailed analysis of the dataset, ensuring that specific rules were formulated to cover all possible valid combinations of intents and entity counts, as shown in Table 2. This manual rule-creation process resulted in rules that were highly precise and relevant to the unique characteristics of the dataset.

On the other hand, for the ATIS, SNIPS, and NLU-Benchmark datasets, we opted for an automated rule generation process. To accomplish this, we developed a Python code solution that leveraged regular expressions applied to the labeled dataset. This code extracted an initial set of rules by identifying patterns related to intents and entity counts within the samples. To ensure the efficiency and consistency of rule generation for these datasets, we incorporated an additional step in the automated process focused on eliminating duplicate rules.

5 Results

In this section, we present an evaluation of the effectiveness of the proposed method in improving the model's performance by utilizing the EMA metric. In particular, we

assess the impact of the proposed method by comparing the model's performance before and after applying the proposed method.

For each dataset and percentage of training data (ranging from 10% to 100%), we trained ten models and constructed two average EMA curves. One curve represents the average performance without the proposed method, while the other illustrates the average performance when using the proposed technique. These two curves allow us to observe the impact of the proposed method on the models' EMA, across different datasets and varying amounts of training data.

In addition to the EMA metric, we also applied the Wilcoxon test, a nonparametric statistical test, to compare two groups of samples. This test is especially useful for small sample sizes or non-normally distributed data. By conducting the Wilcoxon test, we can determine whether the addition of the proposed method has a significant effect on the model's performance. A p value below a pre-defined threshold, typically 0.05, indicates a statistically significant difference in performance.

Table 3 presents an analysis of how performance varies as the size of the training set grows. In this table, the micro-average F_1 scores for both intent classification and entity extraction for each dataset are shown as a function of the percentage of training data used to build the model. Each dataset is presented in a different row, while each column represents different percentages of the total available training data. Both the average and standard deviation for 10 runs are provided.

5.1 Results on AWPS

To analyze the impact of the proposed method, we conducted the described experiments on the AWPS dataset. Figure 4a provides a comparison of the average EMA, clearly showing the positive effect associated with using the proposed method. The x-axis denotes the percentage of training data used, while the y-axis represents the EMA on the test set. It can be easily observed that the average EMA values are higher when utilizing the proposed method, across all percentages of training data.

The highest average EMA was achieved when models were trained with 100% of the training data, regardless of whether the proposed method was applied or not. When the proposed method was applied, the average EMA was 89.26%, reaching a maximum of 90.97% in one of the runs. Notably, there is no overlap between the curves, indicating a clear difference in EMA when the proposed method was used. Furthermore, greater variability in EMA is observed when only 10% of the data was used for training. Moreover, there is a distinct relationship between how accurately models identify intents and entities, as shown by their micro-average F_1 scores, and their success in making

Table 3 Micro-average F_1 scores for intent classification and entity extraction as the size of the training set grows

	Training data percentage									
	Intents 10%	Entities	Intents 20%	Entities	Intents 30%	Entities	Intents 40%	Entities	Intents 50%	Entities
ATIS	87.36 ± 0.46	71.17 ± 1.77	90.40 ± 0.89	81.31 ± 2.44	92.53 ± 0.95	90.23 ± 1.77	94.13 ± 0.74	93.06 ± 0.45	94.07 ± 0.92	93.64 ± 0.32
NLU	81.67 ± 0.96	68.74 ± 1.85	86.81 ± 0.81	73.76 ± 0.88	89.08 ± 0.47	75.71 ± 0.88	89.71 ± 0.93	77.61 ± 1.21	90.60 ± 0.60	78.29 ± 1.24
AWPS	83.45 ± 4.06	93.05 ± 1.03	89.01 ± 1.72	94.63 ± 0.75	93.46 ± 0.91	95.19 ± 0.76	94.46 ± 0.84	96.04 ± 0.26	94.71 ± 1.27	96.14 ± 0.86
SNIPS	96.80 ± 0.51	82.68 ± 1.14	96.78 ± 0.36	88.87 ± 0.53	97.14 ± 0.42	91.49 ± 0.57	97.28 ± 0.36	92.62 ± 0.55	97.17 ± 0.30	93.49 ± 0.42
	60%		70%		80%		90%		100%	
ATIS	95.39 ± 0.45	94.02 ± 0.62	95.95 ± 0.40	94.79 ± 0.24	96.08 ± 0.53	94.92 ± 0.26	96.65 ± 0.42	95.02 ± 0.23	96.46 ± 0.53	95.12 ± 0.26
NLU	91.19 ± 0.81	79.53 ± 1.15	91.38 ± 0.71	80.12 ± 0.64	91.68 ± 0.62	80.85 ± 0.86	91.95 ± 0.50	80.51 ± 0.81	92.06 ± 0.78	81.03 ± 0.82
AWPS	95.07 ± 0.54	96.74 ± 0.59	95.21 ± 0.90	97.03 ± 0.50	95.67 ± 0.62	97.35 ± 0.27	95.78 ± 0.45	97.23 ± 0.38	96.21 ± 0.57	97.27 ± 0.97
SNIPS	97.41 ± 0.59	94.30 ± 0.53	97.55 ± 0.43	94.48 ± 0.44	97.50 ± 0.28	94.74 ± 0.39	97.42 ± 0.46	95.00 ± 0.32	97.74 ± 0.33	95.41 ± 0.37

completely correct predictions for this combined task, especially when utilizing the full training dataset. Table 3 confirms this, indicating that the highest average micro-average F_1 scores for intents are achieved when models are trained with 100% of the training data, followed by the second-highest scores in entity recognition.

To determine the significance of the difference in EMA distributions, we performed Wilcoxon tests for each percentage. The tests aimed to show that the EMA data without using the method are lower than when applying the method. With a p value≈0.0009 for all tests, indicating a very low probability of obtaining such extreme test statistics, we reject the null hypothesis and conclude that the EMA without using the method is lower than when applying the method.

To further evaluate the impact of the proposed method, we selected the model trained with 10% of train set for Fold 2, which exhibited the greatest difference in EMA with and without implementing the method. By applying our method to the predictions of this model, we observed an improvement of 12.6% in the EMA.

Figure 3 presents a histogram analyzing the distribution of the correct intent position within the ranking for incorrect predictions. The red color is used to represent cases in which the list of entities returned was incorrect. The yellow bars indicate cases where intents were corrected by the proposed method. Finally, the green color represents cases in which the method could not repair the original prediction.

Out of 1973 sentences, the trained NLU model accurately recognized all entities, including words not belonging to any entity, in 1417 sentences. Among these, 1144 sentences had both the intent and entities predicted correctly. By implementing our proposed method, we were able to improve the EMA by correcting the intent in 249 sentences. These cases are represented by the yellow bars in Fig. 3 and brought the total number of correctly identified intents and entities to 1393, rising the EMA to 70.6%. It can be observed that, for misclassified instances, the most frequent ranking position for the correct intent is the second. When the correct intent was at this position and the entities had been correctly identified, our method demonstrated a 99.14% success rate at correcting the intent.

Table 4 provides a summary of the 249 intents corrected by the method, along with the corresponding number of sentences for each specific replacement. We can observe that the majority of incorrect classifications (71 intents) were initially classified as “quantity description” intent, even though these utterances do have an entity of type “variable”. Therefore, the method can correct these prediction intents based on the established rules.

When considering only cases where entities were correctly extracted, our method missed only 24 sentences in achieving the highest attainable performance. Out of these 24 sentences, all true intents did not admit any entities. The true intents were “out of scope,“ “insult,” “greet,“ and “affirm.” Mostly, these intents were incorrectly predicted as another intent that did not either admit associated entities. Therefore, our method considered these predictions as

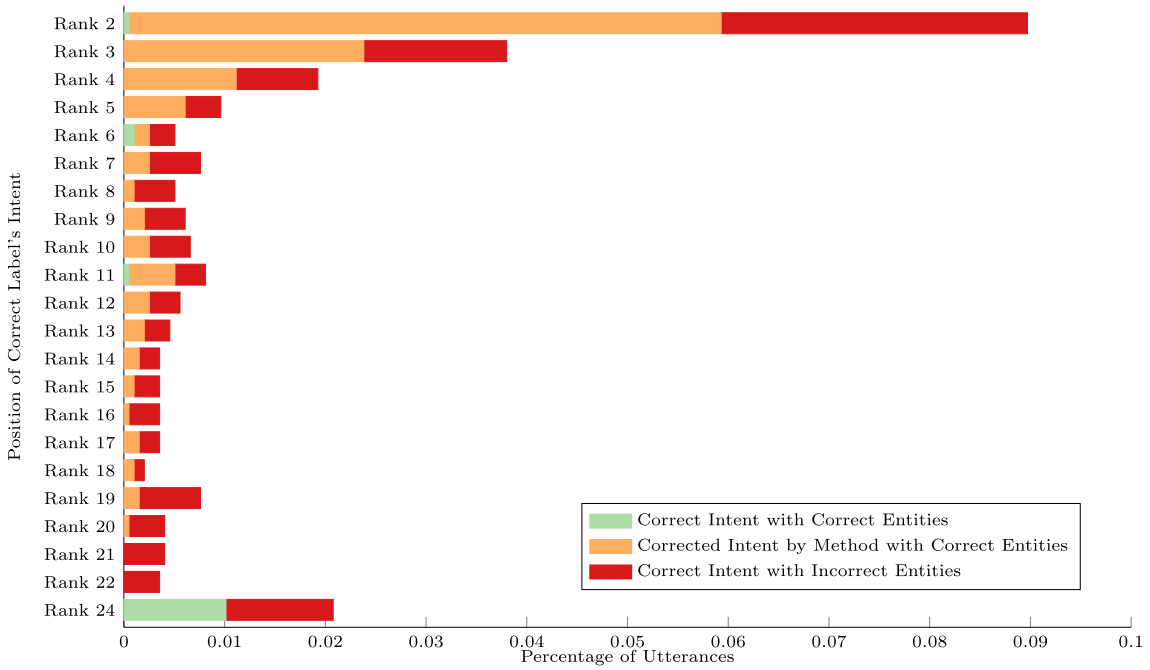


Fig. 3 Distribution of correct intent label in intent ranking for incorrectly predicted intents

Table 4 Summary of corrected intents

Incorrect intent	Corrected intent	Number of sentences
Quantity description	Define letter	71
Equivalence	Define letter	43
Equation	Expression	41
Get letter meaning	Define letter	26
Number	Define letter	18
Equivalence	Expression	13
Multiple intents at once	Expression	10
Quantity description	Expression	8
Multiple intents at once	Define letter	7
Expression	Define letter	5
Number	Expression	4
Define letter	Expression	2
Equation	Define letter	1

correct although they were, in fact, incorrect. In the remaining cases, an intent that required one or more entities was predicted. In these cases, the method recognized the prediction as invalid and selected the highest-ranked intent that allowed for the absence of entities, but this intent did not match the true intent.

5.2 Results on ATIS, SNIPS and NLU-Benchmark datasets

As with AWPS, Fig. 4 shows a comparison of the average EMA when using and when not using the proposed method. This comparison is shown for the ATIS, SNIPS, and NLU-Benchmark datasets, in Fig. 4b–d, respectively.

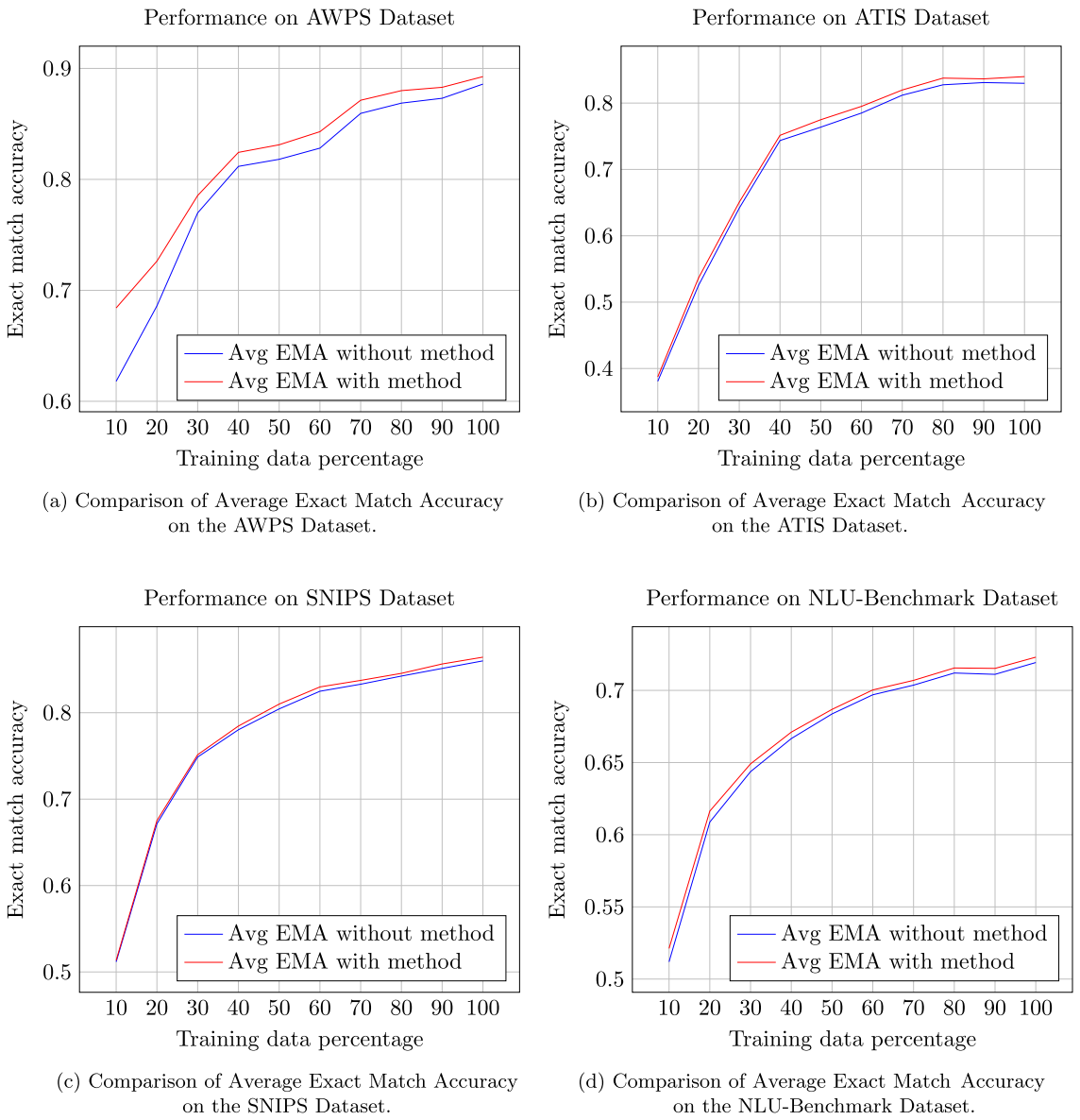
In Fig. 4b and c, a greater improvement of the average EMA can be observed as the percentage of training data increases. Figure 4c reveals only a minimal improvement when the training data percentage is less than 40%. However, as the training percentage increases, the average EMA when using the method consistently outperforms the average EMA when the method was not applied. For the NLU-Benchmark dataset, Fig. 4d shows that the average EMA is higher across all percentages of training data used, compared to not applying the proposed method. In addition, all p values for the Wilcoxon tests conducted on each percentage of training data and for each dataset were consistently below 0.05, providing statistical evidence of a significant difference due to the use of the proposed method. We can hence safely conclude that the proposed method is an effective technique to improve the EMA of models, as it results in better model performance.

6 Conclusions and future work

In this work, we have proposed a simple and effective approach to enhance the EMA of NLU models for T-ODS with relatively low computational cost. The method applies consistency rules to correct invalid outputs that fail to meet the consistency criteria, by iteratively exploring combinations that satisfy a set of constraints specified by using a vector representation. We validated this methodology on AWPS, ATIS, SNIPS, and NLU datasets, demonstrating its effectiveness in improving model performance

However, we shall acknowledge certain limitations associated with the proposed approach. A primary concern is the complexity of the crafted rules, which is heavily influenced by the quantity of entities and intents. As the number increases, the design of rules becomes progressively more challenging. The specification of the set P of valid combinations of intents and entities involves a considerable amount of time and requires updating when the underlying data changes or when the model needs to be retrained to incorporate new intents or entities. We are currently working on alternative less rigid representations that ease the specification of valid intent/entity combinations and provide higher flexibility. In future work, we shall explore the use of proportional logic and/or fuzzy rules to ease the construction of the required specification and simplify compatibility evaluation at inference time.

As T-ODS continue to advance and become more prevalent, it becomes increasingly important to focus not only on accuracy but also on their usability and user experience. In this regard, we shall mention the usability implications of producing a "nlu_fallback" response, requiring asking the user to provide a revised input. The re-prompt indicates a failure to provide the expected answer



(a) Comparison of Average Exact Match Accuracy on the AWPS Dataset.

(b) Comparison of Average Exact Match Accuracy on the ATIS Dataset.

(c) Comparison of Average Exact Match Accuracy on the SNIPS Dataset.

(d) Comparison of Average Exact Match Accuracy on the NLU-Benchmark Dataset.

Fig. 4 Comparison of Average Exact Match Accuracy on the ATIS, SNIPS, and NLU-Benchmark Datasets

to the user but can be considered a minor problem, despite that it prevents the system from providing the service. On the other hand, a more significant negative effect would be caused by responding to an utterance assuming a different intent, as this can mislead or confuse the user and diminish the agent's value. The length of the ranking considered plays a fundamental role in addressing this matter, and an effort should be made to set an optimum value that leads to an adequate balance between avoiding frequent re-prompts and incorrect replacements.

Acknowledgements This research has been supported by project TED2021-129485B-C42 funded by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR; and Grant No. PRE2019-090854, funded by MCIN/AEI/10.13039/501100011033 and "ESF Investing in your future"; and Grant No. ACIF/2021/439 funded by the Valencian Regional Government.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request. During the preparation of this work, the authors used ChatGPT and Grammarly in order to improve language and readability. After using this tools/services, the authors reviewed and edited the content as needed and takes full responsibility for the content of the publication.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Zhang Z, Takanobu R, Zhu Q, Huang M, Zhu X (2020) Recent advances and challenges in task-oriented dialog system
2. Larson S, Leach K (2022) A survey of intent classification and slot-filling datasets for task-oriented dialog
3. He M, Wang J, Ding T, Shen T (2022) Conversation and recommendation: knowledge-enhanced personalized dialog system. Knowl Inf Syst 65(1):261–279. https://doi.org/10.1007/s10115-022-01766-6
4. Han SC, Long S, Li H, Weld H, Poon J (2021) Bi-Directional Joint Neural Networks for Intent Classification and Slot Filling. In: Proc. Interspeech 2021, pp. 4743–4747. https://doi.org/10.21437/Interspeech.2021-2044
5. Zhang C, Li Y, Du N, Fan W, Yu PS (2019) Joint slot filling and intent detection via capsule neural networks
6. E, H., Niu, P., Chen, Z., Song, M.: A novel bi-directional interrelated model for joint intent detection and slot filling (2019)
7. Qin L, Liu T, Che W, Kang B, Zhao S, Liu T (2021) A co-interactive transformer for joint slot filling and intent detection
8. Abro WA, Qi G, Aamir M, Ali Z (2022) Joint intent detection and slot filling using weighted finite state transducer and BERT. Appl Intell 52:17356–17370. https://doi.org/10.1007/s10489-022-03295-9
9. Abro WA, Qi G, Ali Z, Feng Y, Aamir M (2020) Multi-turn intent determination and slot filling with neural networks and regular expressions. Knowl-Based Syst 208:106428. https://doi.org/10.1016/j.knosys.2020.106428
10. Yang P, Ji D, Ai C, Li B (2021) Aise: attending to intent and slots explicitly for better spoken language understanding. Knowl-Based Syst 211:106537. https://doi.org/10.1016/j.knosys.2020.106537
11. Lim J, Son S, Lee S, Chun C, Park S, Hur Y, Lim H (2022) Intent classification and slot filling model for in-vehicle services in Korean. Appl Sci. https://doi.org/10.3390/app122312438
12. Zhang S, Jiang J, He Z, Zhao X, Fang J (2019) A novel slot-gated model combined with a key verb context feature for task request understanding by service robots. IEEE Access 7:105937–105947. https://doi.org/10.1109/ACCESS.2019.2931576
13. Hendrycks D, Gimpel K (2018) A baseline for detecting misclassified and out-of-distribution examples in neural networks
14. Xin J, Tang R, Yu Y, Lin J (2021) The art of abstention: selective prediction and error regularization for natural language processing. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (Volume 1: Long Papers), pp. 1040–1051. https://doi.org/10.18653/v1/2021.acl-long.84
15. Varshney N, Mishra S, Baral C (2022) Towards improving selective prediction ability of NLP systems
16. Huang M, Zhu X, Gao J (2020) Challenges in building intelligent open-domain dialog systems
17. Chen H, Liu X, Yin D, Tang J (2017) A survey on dialogue systems: Recent advances and new frontiers. SIGKDD Explor. Newsl. 19(2):25–35. https://doi.org/10.1145/3166054.3166058
18. Zhang Z, Takanobu R, Zhu Q, Huang M, Zhu X (2020) Recent advances and challenges in task-oriented dialog systems. Science China Technological Sciences, 2011–2027 https://doi.org/10.1007/s11431-020-1692-3
19. Graesser AC, Chipman P, Haynes BC, Olney A (2005) Autotutor: an intelligent tutoring system with mixed-initiative dialogue. IEEE Trans Educ 48(4):612–618. https://doi.org/10.1109/TE.2005.856149
20. Albornoz-De Luise RS, Arevalillo-Herráez M, Arnau D (2023) On using conversational frameworks to support natural language interaction in intelligent tutoring systems. IEEE Trans Learn Technol. https://doi.org/10.1109/TLT.2023.3245121
21. Arnau-González P, Arevalillo-Herráez M, Albornoz-De Luise R, Arnau D (2023) A methodological approach to enable natural language interaction in an intelligent tutoring system. Comput Speech Lang 81:101516. https://doi.org/10.1016/j.csl.2023.101516
22. Arevalillo-Herraez M, Arnau D, Marco-Giménez L (2013) Domain-specific knowledge representation and inference engine for an intelligent tutoring system. Knowl-Based Syst 49:97–105. https://doi.org/10.1016/j.knosys.2013.04.017
23. Arnau D, Arevalillo-Herraez M, Puig L, Gonzéalez-Calero JA (2013) Fundamentals of the design and the operation of an intelligent tutoring system for the learning of the arithmetical and algebraic way of solving word problems. Comput Educat 63:119–130. https://doi.org/10.1016/j.compedu.2012.11.020
24. Arnau D, Arevalillo-Herráez M, González-Calero JA (2014) Emulating human supervision in an intelligent tutoring system for arithmetical problem solving. IEEE Trans Learn Technol 7(2):155–164. https://doi.org/10.1109/TLT.2014.2307306
25. Yan Z, Duan N, Chen P, Zhou M, Zhou J, Li Z (2017) Building task-oriented dialogue systems for online shopping. Proceed AAAI Confer Artif Intell. https://doi.org/10.1609/aaai.v31i1.11182
26. Al-Ajmi A-H, Al-Twairesh N (2021) Building an Arabic flight booking dialogue system using a hybrid rule-based and data driven approach. IEEE Access 9:7043–7053. https://doi.org/10.1109/ACCESS.2021.3049732
27. Li C, Zhou Y, Chao G, Chu D (2022) Understanding users' requirements precisely: a double bi-lstm-crf joint model for detecting user's intentions and slot tags. Neural Comput Appl 34:13639–13648. https://doi.org/10.1007/s00521-022-07171-y
28. Strickland E (2019) IBM Watson, heal thyself: how IBM overpromised and underdelivered on AI health care. IEEE Spectr 56(4):24–31. https://doi.org/10.1109/MSPEC.2019.8678513
29. Leuski A, DeVault D (2012) A study in how nlu performance can affect the choice of dialogue system architecture. In: Proceedings of the 13th annual meeting of the special interest group on discourse and dialogue. SIGDIAL '12, pp. 270–274. Association for Computational Linguistics, USA
30. McTear M (2021) Introducing dialogue systems. Springer, Cham, pp 11–42. https://doi.org/10.1007/978-3-031-02176-3_1
31. Colby KM, Weber S, Hilf FD (1971) Artificial paranoia. Artif Intell 2(1):1–25
32. Colby KM (1974) Ten criticisms of parry. SIGART Bull. 48:5–9. https://doi.org/10.1145/1045200.1045202

33. Wallace RS (2009) In: Epstein, R., Roberts, G., Beber, G. (eds.) The Anatomy of A.L.I.C.E., pp. 181–210. Springer, Dordrecht. https://doi.org/10.1007/978-1-4020-6710-5_13
34. Wei C, Yu Z, Fong S (2018) How to build a chatbot: chatbot framework and its capabilities. In: Proceedings of the 2018 10th international conference on machine learning and computing. ICMLC '18, pp. 369–373. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3195106.3195169
35. Fergus P, Chalmers C (2022) Natural language processing, pp. 217–244. Springer, Cham. https://doi.org/10.1007/978-3-031-04420-5_9
36. Landauer TK, Dumais ST (1997) A solution to plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychol Rev 104:211–240
37. Paladines J, Ramírez J (2020) A systematic literature review of intelligent tutoring systems with dialogue in natural language. IEEE Access 8:164246–164267
38. Hwang M-H, Shin J, Seo H, Im J-S, Cho H (2021) Korasa: pipeline optimization for open-source Korean natural language understanding framework based on deep learning. Mob Inf Syst 2021:1–9. https://doi.org/10.1155/2021/9987462
39. Samant RM, Bachute MR, Gite S, Kotecha K (2022) Framework for deep learning-based language models using multi-task learning in natural language understanding: a systematic literature review and future directions. IEEE Access 10:17078–17097. https://doi.org/10.1109/ACCESS.2022.3149798
40. Zubani M, Sigalini L, Serina I, Gerevini AE (2020) Evaluating different natural language understanding services in a real business case for the Italian language. Procedia Comput Sci 176:995–1004. https://doi.org/10.1016/j.procs.2020.09.095
41. Ni P, Li Y, Li G, Chang V (2020) Natural language understanding approaches based on joint task of intent detection and slot filling for iot voice interaction. Neural Comput Appl 32:16149–16166. https://doi.org/10.1007/s00521-020-04805-x
42. Wang Y, Tang L, He T (2018) Attention-based cnn-blstm networks for joint intent detection and slot filling. In: Sun M, Liu T, Wang X, Liu Z, Liu Y (eds) Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Springer, Cham, pp 250–261
43. Bunk T, Varshneya D, Vlasov V, Nichol A (2020) DIET: lightweight language understanding for dialogue systems
44. Wu C, Luo G, Guo C, Ren Y, Zheng A, Yang C (2020) An attention-based multi-task model for named entity recognition and intent analysis of Chinese online medical questions. J Biomed Inform. https://doi.org/10.1016/j.jbi.2020.103511
45. Zhu S, Cao R, Yu K (2020) Dual learning for semi-supervised natural language understanding. IEEE/ACM Trans Audio, Speech, Lang Process 28:1936–1947. https://doi.org/10.1109/TASLP.2020.3001684
46. Uprety SP, Jeong SR (2022) The impact of semi-supervised learning on the performance of intelligent chatbot system. CMC-Comput, Mater Continua 71(2):3937–3952. https://doi.org/10.32604/cmc.2022.023127
47. Sun R, Rao L, Zhou X (2022) A joint model of natural language understanding for human-computer conversation in iot. Wireless Commun Mobile Comput. https://doi.org/10.1155/2022/2074035
48. Albornoz-De Luise RS, Arnau-González P, Arevalillo-Herráez M (2022) Conversational agent design for algebra tutoring. In: 2022 IEEE International conference on systems, man, and cybernetics (SMC), pp. 604–609. https://doi.org/10.1109/SMC53654.2022.9945524
49. Albornoz-De Luise RS, Arnau-González P, Arevalillo-Herráez M (2022) On providing natural language support for intelligent tutoring systems. In: Rodrigo, M.M., Matsuda, N., Cristea, A.I., Dimitrova, V. (eds.) Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium, pp. 564–568. Springer, Cham. https://doi.org/10.1007/978-3-031-11647-6_116
50. Bocklisch T, Faulkner J, Pawlowski N, Nichol A (2017) Rasa: open source language understanding and dialogue management
51. Hemphill CT, Godfrey JJ, Doddington GR (1990) The ATIS spoken language systems pilot corpus. In: Speech and natural language: proceedings of a workshop Held at Hidden Valley, Pennsylvania, June 24–27
52. Coucke A, Saade A, Ball A, Bluche T, Caulier A, Leroy D, Doumouro C, Gisselbrecht T, Caltagirone F, Lavril T, Primet M, Dureau J (2018) Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces
53. Liu X, Eshghi A, Swietojanski P, Rieser V (2021) In: Marchi, E., Siniscalchi, S.M., Cumani, S., Salerno, V.M., Li, H. (eds.) Benchmarking natural language understanding services for building conversational agents, pp. 165–183. Springer, Singapore. https://doi.org/10.1007/978-981-15-9323-9_15