Combining statistical dialog management and intent recognition for enhanced response selection

DAVID GRIOL*, *Department of Software Engineering, University of Granada, Granada, 18071, Spain.*

ZORAIDA CALLEJAS**, *Department of Software Engineering, University of Granada, Granada, 18071, Spain.*

Abstract

Conversational interfaces are becoming ubiquitous in an increasing number of application domains as Artificial Intelligence, Natural Language Processing and Machine Learning methods associated with the recognition, understanding and generation of natural language advance by leaps and bounds. However, designing the dialog model of these systems is still a very demanding task requiring a great deal of effort given the number of information sources to be considered related to the analysis of user utterances, interaction context, information repositories, etc. In this paper, we present a general framework for increasing the quality of the system responses by combining a statistical dialog management technique and a deep learning-based intention recognizer that allow replacing the system responses initially selected by the statistical dialog model with other presumably better candidates. This approach is portable to different task-oriented domains, a diversity of methodologies for dialog management and intention estimation techniques. We have evaluated our two-step proposal using two conversational systems, assessed several intention recognition methodologies and used the developed modules to dynamically select the system responses. The results of the evaluation show that the proposed framework achieves satisfactory results by making it possible to reduce the number of non-coherent dialog responses by replacing them by more coherent alternatives.

Keywords: Spoken conversational systems, chatbots, dialog management, dialog optimization, intent estimation

1 Introduction

Providing easy and natural access to technical systems is a huge challenge. One common way is to apply graphical user interfaces (GUIs). Spoken Conversational systems (SCSs) surpass conventional GUIs by enabling communication with the systems through various interaction modes, such as speech [1, 17, 35, 36, 62]. These computer programs aim to mimic human communication abilities, encompassing multiple communication modalities. To effectively interact with users, spoken dialog systems typically perform five key tasks: automatic speech recognition (ASR), spoken language understanding (SLU), dialog management (DM), natural language generation (NLG) and text-to-speech synthesis (TTS) [36].

*E-mail: dgriol@ugr.es
**E-mail: zoraida@ugr.es

Vol. 00, No. 0, © The Author(s) 2024. Published by Oxford University Press.

https://doi.org/10.1093/jigpal/jzae045

Over time, the performance of spoken conversational systems has shown improvement, expanding from limited initial applications to more intricate tasks like information retrieval and question answering [3, 29], e-government services [19, 24, 37], e-commerce platforms [26, 44, 55], recommendation and guidance systems [4, 41], e-learning and tutoring systems [12, 18, 23], in-car systems [34, 47], robots and smart environments [10, 28], healthcare [11, 30, 32], security and inspection systems [9, 60] as well as embodied conversational systems [8, 25]. The global conversational systems market is projected to grow from USD 525.4 million in 2021 to USD 3858.5 million by 2030, at a CAGR of 24.8% for the period 2022–2030 [38]. The impact and number of users achieved by recent initiatives such as OpenAI ChatGPT[1] is a visible example of the breakthrough of generative AI systems that can produce content on demand and the expected evolution of the era of conversational systems in the coming years.

However, the majority of commercial task-oriented conversational systems are typically custom-designed using rule-based dialog models and established standards, which demand developers to specify precise steps for the system to follow. As a consequence, adapting these hand-crafted systems to accommodate specific user needs or address new tasks becomes a labor-intensive process [35, 40, 52, 65]. Employing statistical approaches for user modeling and dialog management allows for a broader range of dialog strategies compared to engineered rules [20, 35, 61].

The key advantage of statistical approaches lies in their ability to be trained on real dialogs, capturing the variability in users' different interactions and preferences using the system. Although the model's parameters still require expert knowledge of the specific application domain, the ultimate goal is to develop conversational systems that allow different paths to achieve the objectives, improve portability and provide an easier adaptation [16, 65].

In statistical conversational systems, the DM component is typically divided into two parts: Dialog State Tracking and Dialog Policy. These components serve the purpose of updating the dialog state and determining the appropriate actions to be taken based on the current state [35, 65]. Both of these models are learned from real human-computer dialog data, which means that statistical DMs consider multiple hypotheses for the correct dialog state instead of relying on a single hypothesis [35, 62].

Given that system actions directly impact user experience, the DM plays a significant role in ensuring user satisfaction [20, 35]. However, the success of statistical approaches greatly hinges on the excellence and comprehensiveness of the models and datasets utilized during the training process. Additionally, the size of annotated dialog corpora currently available is often insufficient for many task-oriented systems to adequately explore the wide range of potential dialog states and policies. Another critical challenge is dealing with unseen situations not considered during training. To tackle this issue, it is essential to employ models that are capable of generating suitable system responses in a generalizable manner, allowing the dialog to proceed satisfactorily.

In this paper, we present a general framework focused on solving these problems and increasing the coherence of the system responses selected by statistical dialog management models in task-oriented conversational systems (although it can be also beneficial for rule-based dialog managers). To do this, we propose a two-step approach which aims at replacing the system action initially selected by the SDM with a more coherent one.

Our proposal integrates a statistical deep learning-based model that allows to estimate users' intentions by comparing their utterances with the ones annotated as more coherent for each of the system responses of the conversational system. The output of the intent estimator is considered to

[1] https://openai.com/blog/chatgpt/

select the initial dialog response initially selected by the SDM or replace it by a more coherent candidate.

This module is based on the use of intents and entities by the SLU component of task-oriented conversational systems. Intents represent the underlying purpose or goal behind a user's utterance. They capture the user's intention or the action they want to perform (e.g. for the utterance *I want to rent a house in Granada*, the intent could be labeled as *rent_house*). Generally, developers define the set of intents to be used by the system and supplies a list of examples of utterances that can express each intent. Intent recognition involves classifying user utterances into predefined categories or labels that correspond to the desired actions or outcomes. On the other hand, entities are specific pieces of information or objects that are relevant to the context of a conversation and need to be extracted from users' utterances input. They can be concrete things like names, dates, locations or more abstract concepts like product names, categories or user preferences (e.g. the names of the *cities* provided in the previous example).

After this introduction, the remainder of the paper is as follows. In Section 2, related work on quality of dialog management, intent estimation and user-adapted conversational systems is briefly described. The main contributions of this paper are the two-step approach for increasing coherence of statistical dialog management and intent estimation, presented in Section 3. While the framework is designed to be independent of the actual statistical dialog management approach, we present in Section 4 a showcase for two practical systems acting in different application domains to evaluate our proposal and discuss the experimental results obtained. Finally, the conclusions derived and future research work are presented in Section 5.

2 Related work

Dialog management can be regarded as one of the most challenging tasks for developing spoken conversational systems, since this module encapsulates the logic of the speech application [35, 56]. Consequently, the core of dialog system engineering centers around crafting a fitting dialog management strategy. Diverse surveys offer a thorough examination of dialog management methodologies and architectures [2, 35, 36].

The optimization and adaptation of dialogs can significantly impact speech applications [35, 36]. Specifically, our focus lies in delving into techniques that assess the suitability of the dialog manager's decisions to fine-tune its performance and adopt an improved strategy. To achieve this, it becomes essential to identify whether there exist better system responses than the ones selected by the SDM (Statistical Dialog Manager).

The most widespread strategies to evaluate the quality of the system responses are performance benchmarks and user questionnaires. This way, some authors have used system performance as an indicator for problematic dialogs. As an illustration, Litman and Pan [31] detected challenging scenarios in dialogs by examining the performance of the speech recognizer. They use this data to modify the dialog approach. At the start of each conversation, an initial user-initiated strategy is employed, bypassing the need for confirmations. However, based on the ASR performance, there might be a transition to a system-directed strategy that incorporates explicit confirmations

Other authors focus more on the user perception by assessing users' satisfaction. In the dialog system community, the most relevant ways to measure user satisfaction are the questionnaires proposed in the models PARADISE [54], SASSI [21] and ITU-t Rec. P8.51, which are discussed in detail in [13]. For instance, Gašić *et al.* [14] employ user ratings to improve the dialog performance. In the case of a POMDP-based dialog manager, they trained the optimal policy using a reward

function based on users' ratings collected with Amazon Mechanical Turk. The results demonstrate that their method achieves convergence much more rapidly compared to traditional approaches that rely on a user simulator.

Nevertheless, these methods do not permit real-time adaptation of the dialog flow; instead, they depend on pre-optimized dialog strategies. To address this limitation, it is feasible to predict user judgments by analyzing data that describes user-system interactions. For example, the PARADISE framework assumes that user satisfaction is dependent on task success and dialog costs (e.g. time to complete the task, number of turns, speech recognition accuracy, percentage of semantic errors corrected, frequency of user interruptions to the system, etc.), so that it is possible to predict user judgments based on quantifiable interaction parameters computed from system logs. Similarly, [49] propose the 'interaction quality' metric, which makes it possible to estimate quality exchange-wise, i.e. not for the whole dialog but up to any point in the interaction. The metric depends on assessments provided by a group of expert raters about how the user would feel about the interaction regarding different aspects related to system performance, type of system responses and even whether the user is acoustically annoyed. Here, it has been shown that using expert raters instead of asking the actual users provides valid results [53].

Noh *et al.* [39] followed a similar idea to sort the dialog responses according to their impact on the dialog history. In order to do that, they measure the similarity between the interaction histories (as sequences of dialog responses) weighting each response according to the impact on its neighboring dialog responses. This measure, that they called Discourse Coherence Indicator (DHI), allows to account for how much the given dialog act restricts the range of possible successors. Li *et al.* [27] have recently proposed a dialog-adaptive language model based on the definition of dialog-adaptive pre-training objectives (DAPO) that uses a simulation of dialog-specific features related to coherence, specificity and informativeness. A recent proposal to improve response selection by means of subjective knowledge-seeking dialog contexts and manually annotated responses grounded in subjective knowledge sources is also described in [66].

Differing from the typical concept of discourse coherence, which assesses the overall coherence of an entire dialog as coherent or non-coherent [15, 27, 42], our focus is on examining the coherence of individual system responses independently. This turn-based approach is a requirement for coherence to be used by the dialog manager for action selection. One challenge here is to compute more coherent system responses dynamically during the ongoing interaction instead of rating coherence for a whole dialog in a static and off-line setting.

Another challenge is to find appropriate ways to integrate coherence in the decision making component. Named Entity Recognition (NER), Intention Classification (IC) are two of the most important tasks in many information retrieval related tasks applications [22, 35, 63]. A prominent example is the development of intelligent conversational systems, for which it is essential to correctly understand the users' utterances by recognizing the main concepts and values that they provide (entities) and the tasks or actions they want to perform (intents).

IC can be performed by measuring the similarity between the present user utterance and the collections of example utterances addressing the diverse intents established within the conversational system. The results of this comparison can be used to then select the response associated to the most similar system intent. This problem can treated as a specific classification task that must deal with colloquial utterances with slangs, ellipses, abbreviated words, anaphoras, interjections, etc. RNN-based, attention-based, CNN-based, GNN-based and Transformer models are currently employed given the accuracy of intent recognition results they provide [22, 46, 58, 63, 64].

The main objective of the problem of measuring semantic similarity between sentences is to compare two texts and conclude whether they have a similar meaning [33]. This is a problem of

great complexity due to the ambiguity of language and the fact that sentences that share many words in common may have very different semantic contents and vice versa. Although this is a historical problem within the field of PLN, recent advances in the fields of Artificial Intelligence, Machine Learning and Natural Language Processing have devised new algorithms that are providing very good results. Some of these algorithms receive as input the texts to be compared and provide as output a numerical measure indicating semantic similarity. Other methods use embeddings to represent the words by numerical vectors and measure the semantic similarity through the distance (e.g. row-wise cosine similarity) between the vectors that represent them.

The methods for computing semantic similarity can be classified into those classical methods that do not take into account the context in which each word appears (e.g. Jaccard similarity, the Count vectorizer and TF-IDF Vectorizer) and more recent contextual deep learning methods such as Bidirectional Encoder Representations from Transformers (BERT),[2] XLNET [59] or Generative Pre-trained Transformers (GPT).[3]

Bag-of-words methods are based on extracting text features and representing them by numerical embedding vectors, for which cosine similarity is computed. Jaccard Similarity takes into account the number of unique words in common in the texts being compared. The measure is normalized by taking into account the total number of unique words in the combination of the texts that are compared. Among these methods, Count Vectorizer[4] and TF-DIF Vectorizer[5] stand out. Their main disadvantages are associated with the dimensions of the generated vectors and the number of not relevant components since most of the words do not appear in most documents.

The Count Vectorizer algorithm represents each unique word in the entire corpus by a single vector index. The vector values for each document are the number of times each specific word appears in the text (i.e. integer values including 0). The TF-IDF algorithm tries to overcome the main disadvantage of Count Vectorizer in considering all words equally important without taking into account their semantic content. In this algorithm, each unique word in the corpus is also represented by a single vector index, but the vector components for each word are calculated by the product of two values. Term Frequency (TF) corresponds to the frequency of occurrences of a word within a document, denoting the importance of each word. The Inverse Document Frequency (IDF) is the inverse logarithm of the fraction of documents in which the word appears, representing how frequent the word is in the entire corpus.

The Word Movers Distance (WMD) method [48] uses word embeddings to try to overcome the disadvantages described for the two previous methods. There are different ways to generate these embeddings, the most notable being Word2Vec, Glove and FastText. WMD represents the smallest distance that separates the word embeddings of a given document from the word embeddings of the document to be compared.

Contextual methods usually provide higher accuracy in the similarity measure. The Universal Sentence Encoder (USE) method is based on a model based on Transformers pre-trained by Google. It is available in open source at Tensorflow. This model can be used to compute the contextual word embeddings for each word in a sentence. The element-wise sum of all word vectors is then used to compute the sentence embedding. The semantic similarity is given by computing the cosine similarity of the sentence embeddings.

[2] https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html
[3] https://paperswithcode.com/paper/improving-language-understanding-by
[4] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
[5] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

In 2018, Google introduced BERT as a neural network-driven approach for pre-training models [6]. Unlike methods like word2vec or GloVe, which create individual word representations, BERT considers the contextual nuances of each instance of a particular word through pre-training. To do so, it uses Masked Language Modeling (MLM) to allow the BERT model to learn the existing relationships between words in the language. BERT can be used as a Cross Encoder by incorporating a classification head to the output of the BERT model. The Cross Encoder model has the documents to be compared as input and generates the similarity probability as output.

Metric Learning can also generate embeddings for applications related to semantic similarity. In this method, a neural network, such as BERT, is used to convert texts into embeddings. The embeddings are constructed in such a way that those representing similar words are close in vector space. After training the model, the similarity between texts is calculated by computing the cosine similarity between their corresponding vectors.

Sentence Transformers (SBERT) [45] use BERT and its variants as a base model. They are pre-trained using the contrastive learning metric, which measures the similarity between two embeddings (0 or 1). Since these models require a large number of samples for training, for the work presented in this paper we have pre-trained a model using the sentence-transformers library. For each sentence in the training set we compute its contextual word embeddings using a pre-trained BERT model as an encoder. Then, Mean Pooling process is performed to calculate the element-wise average of all token embeddings and obtain a one-dimensional sentence embedding for the complete text. The model is trained with a Siamese Network architecture with contrastive loss. The cosine similarity between the embeddings representing the texts is computed.

In the proposal presented in this paper, we have completed a comparative assessment of the main methodologies previously described for intent classification. We have selected the methodology that provides the best results for the specific corpora used for the evaluation, so that the result provided by the intent recognition model can be used to modify the response initially selected by the SDM trained from the dialogs of the task in those cases in which its coherence can be increased.

3 General description of the proposal

The main objective of our proposal is to enhance the results of a dialog manager by allowing to choose a possibly better system response when the one selected with highest probability by the DM is lower than a specific threshold. With this aim, we propose to carry out dialog management in two steps as shown in Figure 1.

In the first step of our proposal, the statistical DM produces an n-best list containing possibly suitable next system intents (i.e. responses), expressed in terms of dialog acts (*DA*) [51]. A dialog act refers to the intention or function behind a particular utterance or speech act in a dialog. It represents the communicative purpose of a speaker's turn (e.g. making requests, giving information, asking questions, expressing opinions, acceptance or rejection, etc.).

The statistical dialog management methodology that we have used to evaluate our proposal is explained in [7, 20]. This methodology uses a data structure, called *Dialog Register* (*DR*), for Dialog State Tracking. The *DR* contains information about the entities provided by the user during the dialog and represents the current dialog state. This data is encoded by only considering whether the user has supplied a value for a particular intent or entity, without considering the specific values of the entities itself. The value '0' is used when the entity or intent is unknown or the value is not given. To make decisions regarding whether the state of a particular value in the *DR* is '1' or '2', the system

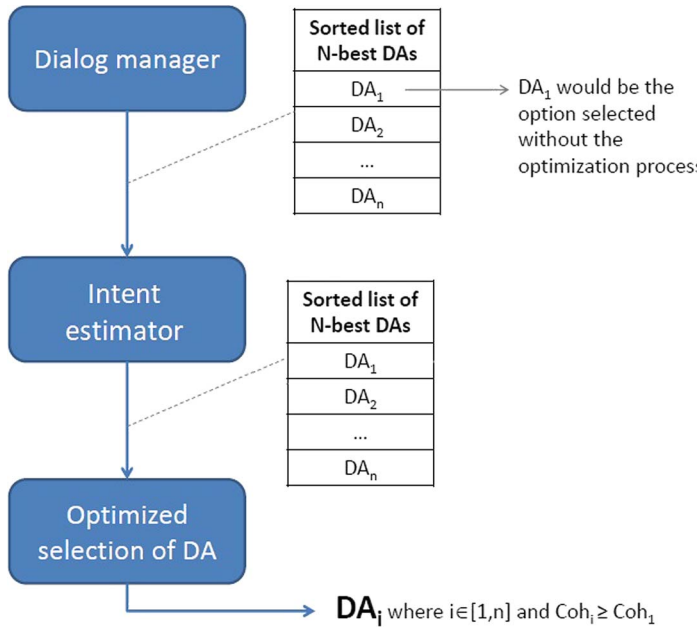

FIGURE 1. Schematic overview of our proposal.

relies on confidence scores supplied by the ASR and SLU modules to use a '1' when the value of the associated score is higher than a given threshold and a '2' if it is lower.

For the LEGO corpus (see Section 4.1), the entities considered for the *DR* are *arrival*, *departure*, *busRoute*, *time* and the concepts are *yes*, *no*, *help*, *repeat*, *start*, *goodbye*. Table 1 shows the state of the *DR* with a sample dialog from the corpus. In the *DR*, each slot has the option to be marked with either '1' or '0' to indicate the active or inactive status of the corresponding intent. The same rule applies to entities. However, entities have an additional option to use the value '2' when the attribute value provided to the system requires clarification or validation from the user.

Using this representation of the dialog state by means of the *DR*, the following equation is used to select the next system action by the SDM:

$$\hat{DA}_i = \underset{DA_i \in \mathcal{DA}}{\operatorname{argmax}} P(DA_i | DR_i, DA_{i-1}) \tag{1}$$

where the updated *DR* at time i and the intent selected by the DM in the previous turn (DA_{i-1}) are considered to decide the next system intent (DA_i). This equation can be solved by means of a classification process, which takes the codification of the current dialog state as input (denoted by the pair DR_i, DA_{i-1}) and the output of the classifier is the probability of selecting each of the system intents given this dialog state.

The classification function can be defined in several ways [7]. We have used a wide variety of traditional machine learning models and architectures: K-Nearest Neighbors (KNN), logistic regression, decision tree, gradient boosting and Multilayer Perceptron (MLP). Five Multilayer Perceptron architectures have been considered, varying in the number of hidden layers from one to five. In each of these architectures, the hidden layers use a linear activation function, while the output layer adopts a sigmoid activation function.

TABLE 1. State of the dialog register for a sample dialog of the LEGO task

System output	User input	Arrival	Departure	Route	Time	Yes	No	Help	Repeat	Start	Goodbye
Welcome to the CMU Let's Go bus information system. To get help at any time, just say Help or press zero. What can I do for you?	I need information on the 54C	0	0	1	0	0	0	0	0	0	0
Where are you leaving from?	From EAST OHIO STREET	0	1	1	0	0	0	0	0	0	0
Where do you want to go?	To POLISH HILL	2	1	1	0	0	0	0	0	0	0
Going to POLISH HILL. Is this correct?	Yes	1	1	1	0	1	0	0	0	0	0
When do you wanna travel?	WHEN IS THE NEXT BUS	1	1	1	1	0	0	0	0	0	0
Alright. Hold on. Let me check that for you. There is a 54C leaving EAST OHIO STREET AT NASH ISLAND at 8 50 p.m. It will arrive at DOBSON STREET AT HERRON at 9 11 p.m. If you are finished, you can say goodbye.	Goodbye	1	1	1	1	0	0	0	0	0	1

Deep learning architectures utilizing convolutions were initially considered, but eventually rejected as they proved unsuitable for the dataset format used in this task. While Convolutional Neural Networks (CNNs) are adept at creating high-level data representations with a spatial inductive bias, the 1-D vector coding representation we designed does not align well with such architectures and, consequently, offers limited potential for their application.

In the second step of our proposal, an intent recognition model is used to decide whether to select as system response the intent selected by the DM in the first step or an alternative one. To develop this model, we have evaluated the main methods described in Section 2. With respect to non-contextual methods, we have evaluated Jaccard Similarity, Count Vectorizer and TF-IDF algorithms. Jaccard Similarity has been computed for N-grams (w-shingling) using the Python textdistance library.[6] The Count Vectorizer and TF-IDF algorithms have been implemented using the sklearn library.

The Fast WMD algorithm of the gensim library[7] has been used to efficiently calculate the distance between one word and all the others. In our work we have used a model of FastText word embeddings pre-trained with Wikipedia texts, in order to avoid Out of vocabulary problems using Word2Vec or Grove methods.

Regarding contextual methods, we have evaluated the Universal Sentence Encoder (USE) method, the use of Bidirectional Encoder Representations from Transformers (BERT), the Metric Learning methodology, Sentence Transformers (SBERT) and SVM classifiers. The USE pre-trained model is available in open source at Tensorflow.[8] The sentence:transformers library has been used to use the open-source Cross Encoder BERT and SBER Bi-Encoder pre-trained models.

Finally, our proposal uses these two steps to select the system intent according to an estimation of the coherence provided by the SDM regarding the current dialog situation and the estimation provided by the intent estimator according to the results of the semantic similarity between the utterances. We propose to use the procedure described by the following if-clause to select the next system dialog act:

$DAs_1 \leftarrow statical\_dialog\_management()$
$DAs_2 \leftarrow intent\_estimator(DAs)$
if $(DAs_1[0] = DAs_2[0] \;\; OR \;\; Prob(DAs_1[0]) \geq \alpha)$ **then return** $DAs_1[0]$
else return $DAs_2[0]$
end if

If the best dialog act chosen by the SDM is the same that the best one provided by the intent estimator or its probability is equal or higher than a given threshold, then this dialog act is selected as the next system response, in other case, it considers a more coherent alternative from the n-best list provided by the intent estimator.

This approach can be seamlessly applied to various application domains and allows for the utilization of a wide range of dialog managers and automatic coherence estimation techniques. As a result, it can be readily incorporated into existing systems with one simple requirement: both the DM and the intent estimator must generate an n-best list of system responses.

[6] https://pypi.org/project/textdistance/
[7] https://radimrehurek.com/gensim/auto_examples/tutorials/run_wmd.html
[8] https://www.tensorflow.org/hub/tutorials/semantic_similarity_with_tf_hub_universal_encoder

4 Experiments and evaluation results

In this section we describe the different dialog systems for which our proposal has been applied and evaluated (Section 4.1), the process and measures defined for the evaluation (Section 4.2) and the results of the evaluation of the statistical dialog manager, the intent recognizer and the overall proposals (Sections 4.3, 4.4 and 4.5).

4.1 Experimental dialog systems

The LEGO corpus originates from interactions with the 'Let's Go Bus Information System' at Carnegie Mellon University in Pittsburgh [43]. This spoken dialog system, which offers bus schedule details for Pittsburgh, was made available for the public in 2005, achieving 20,000 calls from March to December 2005 [43].

We have chosen the Let's Go task for assessing our proposal, driven by two key considerations. Firstly, the corpus was acquired through an operational dialog system serving actual users. This presents a challenge in devising novel dialog strategies capable of surpassing the efficacy of the manually crafted dialog model. Secondly, Let's Go stands as a widely acknowledged benchmark in the dialog system community for experimentation and evaluation [50, 57]. For evaluating the suitability of our approach, we utilized the LEGO corpus [50]. This dataset encompasses 200 calls, comprising 4,885 exchanges between users and the 'Let's Go Bus Information System', all recorded in 2006.

On the contrary, DI@L-log serves as a spoken conversational system explicitly crafted for obtaining at-home monitored data from individuals with type 2 diabetes [5]. Users provide their weight, blood pressure (both systolic and diastolic measurements) and blood sugar levels. The system then verifies and assesses this data, promptly providing patients with feedback regarding their present health condition. Furthermore, it transmits the outcomes to medical professionals, enabling them to track the patient's advancement and address system-generated notifications concerning unusual variations. The entities defined for this task are related to the specific information pieces that the system requires (e.g. Weight, Sugar, Systolic Pressure and Diastolic Pressure). The set of system intents encompasses those necessary for asking and confirming this data, providing information on the values provided, conveying possible alerts and handling welcoming and farewell interactions. Our experiments utilized a corpus of 300 dialogs to evaluate our proposal.

We have chosen the DI@L-log task to evaluate our proposal since these dialogs were acquired following different strategies: a system-driven strategy in which the dialog system queries the user entity by entity; a mixed-initiative strategy in which the user can provide additional entities not explicitly requested in the current query; and a user-initiative strategy in which the user can provide more than one entity at any time and in any order. Thus, it is possible to use this task to evaluate our proposal with the full set of strategies defined for this system.

4.2 Process followed for the evaluation

We split the LEGO and DI@L-log corpus for training and testing performing a random 80/20 split. Subsequently, we conducted a 5-fold cross-validation phase for each task, using the cross-entropy loss to identify the optimal values for the optimizer and model hyperparameters. Specifically, the following settings were established:

- For the KNN algorithm, we considered five neighbors.
- The function used to assess the quality of decision tree splits was entropy.

TABLE 2. Results of the SDM for the LEGO and DI@L-log corpora

	LEGO task		DI@L-log task	
Model	**Accuracy**	**p-value against best**	**Accuracy**	**p-value against best**
Logistic regression	0.796	$1.6 \cdot 10^{-15}$	0.873	$1.5 \cdot 10^{-15}$
KNN	0.786	$1.2 \cdot 10^{-15}$	0.871	$1.9 \cdot 10^{-15}$
Decision Tree	**0.828**	-	**0.909**	-
Gradient Boosting	0.819	$1.3 \cdot 10^{-15}$	0.908	$1.4 \cdot 10^{-15}$
MLP 1 hidden	0.793	$1.3 \cdot 10^{-15}$	0.888	$1.5 \cdot 10^{-15}$
MLP 2 hidden	0.783	$1.3 \cdot 10^{-15}$	0.873	$1.5 \cdot 10^{-15}$
MLP 3 hidden	0.791	$2.6 \cdot 10^{-7}$	0.875	$2.1 \cdot 10^{-7}$
MLP 4 hidden	0.793	$7.9 \cdot 10^{-10}$	0.876	$6.5 \cdot 10^{-10}$
MLP 5 hidden	0.794	$8.7 \cdot 10^{-14}$	0.876	$6.6 \cdot 10^{-14}$

- Each MLP model was configured with the ADAM optimizer, no L2 regularization penalty and 256 neurons per hidden layer.
- Learning rates were set as follows: 0.0005 for MLPs with 1 and 2 hidden layers, 0.0001 for MLPs with 3 and 4 hidden layers and 0.00005 for the one with 5 hidden layers.
- The number of training epochs was set to 200 for MLPs with 1 and 2 hidden layers, and 300 for the rest.

To evaluate the performance of the statistical dialog manager and the intent estimator, we utilized two metrics: accuracy and F-score. The accuracy metric indicates the percentage of turns in which the DM or intent estimator's top hypothesis matches the reference answer in the corpus. To assess the statistical significance of the evaluation metrics, we conducted a paired t-test, comparing the results of different models. This analysis allows us to determine whether any observed differences in performance are statistically significant.

4.3 First step. Evaluation of the statistical dialog manager

Table 2 describes the results obtained for the different classification functions evaluated for the statistical dialog manager. For the LEGO task, we can observe that the best performing model is the decision tree. We report a 1-best guess rate of 82.8%, which is statistically supported by the paired t-test. This model's micro-averaged precision, recall and F1-score are 82.20%, 82.76% and 82.50%, respectively.

For the DI@L-log task, the best performing model is also the decision tree, with slightly differences with the results obtained using Gradient Boosting. We report a 1-best guess rate of 90.9%, which is also statistically supported by the paired t-test. The main errors in the selection of the system responses have been detected given the difficulties found in the statistical DM to successfully distinguish among the previously described range of strategies used by the users to provide their responses in this dialog system.

4.4 Second step. Evaluation of the intent estimator

Table 3 describes the results obtained for the different methods evaluated for the intent estimator. We can observe that the SBERT Cross Encoder has the best performance for both dialog tasks, followed closely by SBERT Bi-Encoder. The results obtained are better than the ones provided by Jaccard,

TABLE 3. Results of the intent estimator for the LEGO and DI@L-log corpora

	LEGO task		DI@L-log task	
Model	**Accuracy**	**F-score**	**Accuracy**	**F-score**
Jaccard_score	0.709	0.708	0.823	0.821
TFIDF_cosine score	0.649	0.645	0.776	0.774
NegWMD_score	0.702	0.701	0.814	0.813
USE_cosine score	0.701	0.700	0.812	0.812
SBERT CrossEncoder_score	**0.798**	0.797	**0.869**	0.868
SBERT BiEncoder_score	0.779	0.777	0.854	0.852

TABLE 4. Results of the intent estimator for the LEGO and DI@L-log corpora

Combination of the SDM and intent estimator	**Accuracy**	**F-score**
LEGO task	**0.864**	0.862
DI@L-log task	**0.953**	0.952

TFIDF, WMD and USE methods. These results show that considering the context of the words is very important for the two dialog tasks and the domain of the words generally employed by the users are common English language.

4.5 Overall evaluation of the proposal

Finally, we have assessed our proposal according to Figure 1, selecting from Tables 2 and 3 the best options for the implementation of the statistical dialog manager and the intent estimator. Table 4 shows the result obtained for the assessment of our proposal. As it can be observed, the combination of the SDM and the intent estimator following our proposal allows for improving the accuracy results achieving an accuracy value of 86.4% for the LEGO task and 95.3% for the DI@L-log task, improving the best results provided by both modules. A manual review of the selected system responses in both dialog tasks allowed us to see that also many of the 1-best non-coherent dialog acts first selected by the dialog manager were corrected.

In more complex domains, our proposal is expected to have a greater impact on dialog systems. Such systems often face challenges in defining a well-performing set of rules, leading to potentially lower overall coherence. In contrast, systems utilizing statistical methods for policy creation tend to perform better in complex domains. To illustrate this, consider an example dialog from the LEGO task (see Figure 2), where the SDM selects a non-coherent system response. However, through the re-ranking process based on our two-step dialog management approach, a more suitable system response is chosen. This showcases the potential of the re-ranking process to improve the selection of the system responses.

5 Conclusions and future work

Statistical dialog management poses several advantages over rule-based conversational systems, including portability, flexibility and dynamic adaptability. However, due to their unpredictable nature, they may behave in ways that are unexpected at times. This makes it difficult for the user

System	Welcome to the CMU Let's Go bus information system. What can I do for you? (SDA Welcome)
User	*CAN USE 16B FROM TELL ME TO DOWNTOWN PITTSBURGH*
System	Going to DOWNTOWN PITTSBURGH. Is this correct? (SDA Confirm_destination)
User	*YES*
System	Okay. Where do you wanna leave? (SDA Ask_for_origin)
User	*DOWN YES*
System	For example, you can say, FORBES AND MURRAY, DOWNTOWN, or MCKEESPORT. (SDA Ask_for_origin)
User	*TOWN YES*
Sys. (Non re-ranked)	(SDA Confirm_origin)
Sys. (Re-ranked)	Which neighborhood do you want to leave from? (SDA Ask_for_origin_neighborhood)

FIGURE 2. A dialog example that highlights the contrast between using only the SDM or employing a second step re-ranking with the intent estimation. Towards the end of the dialog, the SDM selects the DA *Confirm_origin*. However, after applying re-ranking based on intent estimation, the DA *Ask_for_origin_neighborhood* is chosen instead.

to anticipate their responses. In this paper, we have presented a general-purpose framework to select the best system response in two steps: first, the dialog manager produces a list of the *n*-best dialog acts for the current dialog state, then an intent estimation module computes the coherence of each alternative. Our proposal uses these two steps to select the most coherent dialog response according to the results provided by these two modules. This approach is independent of the domain, the actual dialog manager used and how coherence is computed.

To evaluate our proposal we have used two task-oriented dialog systems interacting in different application domains. The experimental results using two corpora acquired for these systems show that our technique made it possible to achieve an accuracy over 86% and 95% of correctly provided system responses for the LEGO and DI@L-log dialog systems, respectively, reducing the number of non-coherent dialog responses initially selected by the SDM.

As the approach is designed to apply to all sorts of dialog management systems (which provide an *n*-best list of system dialog acts), for future work we are interested in testing our approach with other dialog managers in different domains. How our proposed method behaves in a live system interacting with real users should also be investigated in future work. Here, we expect our method to have a positive influence on the interaction performance as it has been shown to increase the overall quality of the system responses.

Acknowledgements

The research leading to these results has received funding from 'CONVERSA: Effective and efficient resources and models for transformative conversational AI in Spanish and co-official languages' project with reference TED2021-132470B-I00, funded by MCIN/AEI/10.13039/501100011033 and by the E.U. 'NextGenerationEU/PRTR'. This work was also partially supported by the E.U.'s Horizon 2020 research and innovation programme under grant agreement no. 823907 (MENHIR project: https://menhir-project.eu) and the GOMINOLA project (PID2020-118112RB-C21 and PID2020-118112RB-C22, funded by MCIN/AEI/10.13039/501100011033).

References

[1] E. Adamopoulou and L. Moussiades. Chatbots: history, technology, and applications. *Machine Learning With Applications*, **2**, 100006, 2020.

[2] M. Ahmed, R. Riyaz and S. Afzal. A comparative study of various approaches for dialogue management. *International Journal of Advanced Computer Technology*, **2**, 89–96, 2013.

[3] H. Al-Thani, T. Elsayed and B. J. Jansen. Improving conversational search with query reformulation using selective contextual history. *Data and Information Management*, **7**, 100025, 2023.

[4] Z. K. A. Baizal, D. H. Widyantoro and N. U. Maulidevi. Computational model for generating interactions in conversational recommender system based on product functional requirements. *Data & Knowledge Engineering*, **128**, 101813, 2020.

[5] L. A. Black, M. McTear, N. Black, R. Harper and M. Lemon. Appraisal of a conversational artefact and its utility in remote patient monitoring. In *Proc. of CBMS'05*, pp. 506–508, 2005.

[6] J. Briskilal and C. N. Subalalitha. An ensemble model for classifying idioms and literal texts using bert and roberta. *Information Processing & Management*, **59**, 102756, 2022.

[7] P. Canas, D. Griol and Z. Callejas. Towards versatile conversations with data-driven dialog management and its integration in commercial platforms. *Journal of Computational Science*, **55**, 101443, 2021.

[8] H. Chung, H. Kang and S. Jun. Verbal anthropomorphism design of social robots: investigating users' privacy perception. *Computers in Human Behavior*, **142**, 107640, 2023.

[9] S. Colabianchi, M. Bernabei and F. Costantino. Chatbot for training and assisting operators in inspecting containers in seaports. *Transportation Research Procedia*, **64**, 6–13, 2022.

[10] H. Cuayáhuitl, K. Komatani and G. Skantze. Introduction for speech and language for interactive robots. *Computer Speech & Language*, **34**, 83–86, 2015.

[11] K. Denecke and R. May. Investigating conversational agents in healthcare: application of a technical-oriented taxonomy. In *Proc. of HCist—Int. Conference on Health and Social Care Information Systems and Technologies*, vol. 219, pp. 1289–1296. Lisbon, Portugal, 2023.

[12] D. O. Eke. ChatGPT and the rise of generative AI: threat to academic integrity? *Journal of Responsible Technology*, **13**, 100060, 2023.

[13] K. P. Engelbrecht. *Estimating Spoken Dialog System Quality With User Models*. Springer Science & Business Media, 2012.

[14] M. Gačić, C. Breslin, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis and S. J. Young. On-line policy optimisation of Bayesian spoken dialogue systems via human interaction. In *Proc. of ICASSP'13*, pp. 8367–8371. Vancouver, Canada, 2013.

[15] S. Gandhe and D. Traum. An evaluation understudy for dialogue coherence models. In *Proc. of SIGdial'08*, pp. 172–181. Columbus, Ohio, USA, 2008.

[16] M. Gasic, N. Mrksic, L. M. Rojas-Barahona, P. H. Su, S. Ultes, D. Vandyke, T. H. Wen and S. J. Young. Dialogue manager domain adaptation using Gaussian process reinforcement learning. *Computer Speech & Language*, **45**, 552–569, 2017.

[17] L. Gkinko and A. Elbanna. The appropriation of conversational AI in the workplace: a taxonomy of AI chatbot users. *International Journal of Information Management*, **69**, 102568, 2023.

[18] A. C. Graesser and H. Li. Intelligent tutoring systems and conversational agents. In *Int.*

Encyclopedia of Education, R. J. Tierney, F. Rizvi and K. Ercikan, eds, 4th edn., pp. 637–647. Elsevier, Oxford, 2023.

[19] D. Griol, D. Pérez Fernández and Z. Callejas. Hispabot-Covid19: the official Spanish conversational system about Covid-19. In *Proc. of 5th Int. Conference IberSPEECH'21*. Valladolid, Spain, 2021.

[20] D. Griol, Z. Callejas, R. López-Cózar and G. Riccardi. A domain-independent statistical methodology for dialog management in spoken dialog systems. *Computer Speech & Language*, **28**, 743–768, 2014.

[21] K. S. Hone and R. Graham. Towards a tool for the subjective assessment of speech system interfaces (SASSI). *Natural Language Engineering*, **6**, 287–303, 2000.

[22] X. Huang, T. Ma, L. Jia, Y. Zhang, H. Rong and N. Alnabhan. An effective multimodal representation and fusion method for multimodal intent recognition. *Neurocomputing*, **548**, 126373, 2023.

[23] A. Iku-Silan, G.-J. Hwang and C.-H. Chen. Decision-guided chatbots and cognitive styles in interdisciplinary learning. *Computers & Education*, **201**, 104812, 2023.

[24] J. Ju, Q. Meng, F. Sun, L. Liu and S. Singh. Citizen preferences and government chatbot social characteristics: evidence from a discrete choice experiment. *Government Information Quarterly*, **40**, 101785, 2023.

[25] E. Konya-Baumbach, M. Biller and S. von Janda. Someone out there? A study on the social presence of anthropomorphized chatbots. *Computers in Human Behavior*, **139**, 107513, 2023.

[26] A. Kumar-Kushwaha, P. Kumar and A. Kumar-Kar. What impacts customer experience for B2B enterprises on using AI-enabled chatbots? Insights from big data analytics. *Industrial Marketing Management*, **98**, 207–221, 2021.

[27] J. Li, Z. Zhang and H. Zhao. Dialogue-adaptive language model pre-training from quality estimation. *Neurocomputing*, **516**, 27–35, 2023.

[28] M. Li, F. Guo, X. Wang, J. Chen and J. Ham. Effects of robot gaze and voice human-likeness on users' subjective perception, visual attention, and cerebral activity in voice conversations. *Computers in Human Behavior*, **141**, 107645, 2023.

[29] R. Li, Z. Jiang, L. Wang, X. Lu and M. Zhao. Directional attention weaving for text-grounded conversational question answering. *Neurocomputing*, **391**, 13–24, 2020.

[30] Y. Li, S. Liang, B. Zhu, X. Liu, J. Li, D. Chen, J. Qin and D. Bressington. Feasibility and effectiveness of artificial intelligence-driven conversational agents in healthcare interventions: a systematic review of randomized controlled trials. *International Journal of Nursing Studies*, **143**, 104494, 2023.

[31] D. Litman and S. Pan. Designing and evaluating an adaptive spoken dialogue system. *User Modeling and User-Adapted Interaction*, **12**, 111–137, 2002.

[32] L. Martinengo, E. Lum and J. Car. Evaluation of chatbot-delivered interventions for self-management of depression: content analysis. *Journal of Affective Disorders*, **319**, 598–607, 2022.

[33] J. Martínez-Gil. A comprehensive review of stacking methods for semantic similarity measurement. *Machine Learning With Applications*, **10**, 100423, 2022.

[34] M. Matei, L. Alboaie and A. Iftene. Safety navigation using a conversational user interface for visually impaired people. *Procedia Computer Science*, **207**, 1164–1173, 2022.

[35] M. F. McTear. Conversational AI. In *Conversational Agents, and Chatbots*. Morgan and Claypool Publishers, Dialogue systems, 2020.

[36] M. F. McTear, Z. Callejas and D. Griol. *The Conversational Interface: Talking to Smart Devices*. Springer, 2016.

[37] B. Moser-Plautz and L. Schmidthuber. Digital government transformation as an organizational response to the covid-19 pandemic. *Government Information Quarterly*, **40**, 101815, 2023.
[38] MRC. *Chatbot Market Size, Share & Trends Analysis, by Type (Standalone, Web-Based, Messenger-Based/Third Party), by Application (Bots for Service, Bots for Social Media), Region and Forecast Period 2022–2030*. Technical Report. Market Research Center, 2022.
[39] H. Noh, S. Lee, K. Kim, K. Lee and G. G. Lee. Ranking dialog acts using discourse coherence indicator for language tutoring dialog systems. In *Proc. of IWSDS'11*, pp. 203–214. Granada, Spain, 2011.
[40] T. Paek and R. Pieraccini. Automating spoken dialogue management design using machine learning: an industry perspective. *Speech Communication*, **50**, 716–729, 2008.
[41] D. Pramod and P. Bafna. Conversational recommender systems techniques, tools, acceptance, and adoption: a state of the art review. *Expert Systems With Applications*, **203**, 117539, 2022.
[42] A. Purandare and D. J. Litman. Analyzing dialog coherence using transition patterns in lexical and semantic features. In *Proc. of 21st AIRS*, pp. 195–200. Coconut Grove, Florida, USA, 2008.
[43] A. Raux, D. Bohus, B. Langner, A. W. Black and M. Eskenazi. Doing research on a deployed spoken dialogue system: one year of Let's go! Experience. In *Proc. of ICSLP'06*, pp. 65–68. Pittsburgh, USA, 2006.
[44] H. D. Rebelo, L. A. F. de Oliveira, G. M. Almeida, C. A. M. Sotomayor, V. S. N. Magalhaes and G. L. Rochocz. Automatic update strategy for real-time discovery of hidden customer intents in chatbot systems. *Knowledge-Based Systems*, **243**, 108529, 2022.
[45] N. Reimers and I. Gurevych. Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv, 1908.10084, 2019.
[46] S. Rizou, A. Paflioti, A. Theofilatos, A. Vakali, G. Sarigiannidis and K. C. Chatzisavvas. Multi-lingual name entity recognition and intent classification employing deep learning architectures. *Simulation Modelling Practice and Theory*, **120**, 102620, 2022.
[47] O. Rubleva, E. Emelyanova, T. Mozharova, N. Polshakova and N. Gavrilovskaya. Development of a chatbot for a car service. In *Proc. of MIST: Aerospace-IV Conference*, vol. 2700, page 040016. Krasnoyarsk, Russian Federation, 2023.
[48] R. Sato, M. Yamada and H. Kashima. Re-evaluating word mover's distance. arXiv, 2105.14403, 2021.
[49] A. Schmitt and S. Ultes. Interaction quality: assessing the quality of ongoing spoken dialog interaction by experts—and how it relates to user satisfaction. *Speech Communication*, **74**, 12–36, 2015.
[50] A. Schmitt, S. Ultes and W. Minker. A parameterized and annotated corpus of the CMU Let's go bus information system. In *Proc. of LREC'12*, pp. 3369–3373. Istanbul, Turkey, 2012.
[51] J. R. Searle. Speech acts. In *An Essay on the Philosophy of Language*. Cambridge University Press, 1969.
[52] P. Sierra. BDI logic applied to a dialogical interpretation of human-machine cooperative dialogues. *Logic Journal of the IGPL*, **29**, 536–548, 2020.
[53] S. Ultes, A. Schmitt and W. Minker. On quality ratings for spoken dialogue systems–experts vs. users. In *Proc. of NAACL-HLT'13*, pp. 569–578. Association for Computational Linguistics, Atlanta, Georgia, USA, 2013.
[54] M. A. Walker, D. J. Litman, C. A. Kamm, A. A. Kamm and A. Abella. PARADISE: a framework for evaluating spoken dialogue agents. In *Proc. of ACL*, pp. 271–280. Madrid, Spain, 1997.

[55] X. Wang, X. Lin and B. Shao. How does artificial intelligence create business agility? Evidence from chatbots. *International Journal of Information Management*, **66**, 102535, 2022.
[56] Y. Wilks, R. Catizone, S. Worgan and M. Turunen. Some background on dialogue management and conversational speech for dialogue systems. *Computer Speech and Language*, **25**, 128–139, 2011.
[57] J. D. Williams, I. Arizmendi and A. Conkie. Demonstration of AT&T Let's go: a production-grade statistical spoken dialog system. In *Proc. of SLT'10*, pp. 157–158. Berkeley, California, USA, 2010.
[58] C. Wu, G. Luo, C. Guo, Y. Ren, A. Zheng and C. Yang. An attention-based multi-task model for named entity recognition and intent analysis of Chinese online medical questions. *Journal of Biomedical Informatics*, **108**, 103511, 2020.
[59] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov and Q. V. Le. XLNet: generalized autoregressive pretraining for language understanding. arXiv, 1906.08237, 2019.
[60] J. Yoo and Y. Cho. ICSA: intelligent chatbot security assistant using text-CNN and multi-phase real-time defense against SNS phishing attacks. *Expert Systems With Applications*, **207**, 117893, 2022.
[61] S. Young. *The Statistical Approach to the Design of Spoken Dialogue Systems*. Technical Report. Cambridge University Engineering Department, 2002.
[62] S. Young. *Hey Cyba. The Inner Workings of a Virtual Personal Assistant*. Cambridge University Press, 2021.
[63] X. Zhang, F. Cai, X. Hu, J. Zheng and H. Chen. A contrastive learning-based task adaptation model for few-shot intent recognition. *Information Processing & Management*, **59**, 102863, 2022.
[64] X. Zhang, M. Jiang, H. Chen, J. Zheng and Z. Pan. Incorporating geometry knowledge into an incremental learning structure for few-shot intent recognition. *Knowledge-Based Systems*, **251**, 109296, 2022.
[65] X. Zhang, B. Peng, J. Gao and H. Meng. Toward self-learning end-to-end task-oriented dialog systems. In *Proc. of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 516–530. Edinburgh, UK, 2022.
[66] C. Zhao, S. Gella, S. Kim, D. Jin, D. Hazarika, A. Papangelis, B. Hedayatnia, M. Namazifar, Y. Liu and D. Hakkani-Tur. "What do others think?": Task-oriented conversational modeling with subjective knowledge. arXiv, 2305.12091, 2023.

Received 1 April 2023