

Mapping the Dialog Act Annotations of the LEGO Corpus into ISO 24617-2 Communicative Functions

Eugénio Ribeiro^{1,2}, Ricardo Ribeiro^{1,3}, David Martins de Matos^{1,2}

¹ INESC-ID Lisboa, Portugal

² Instituto Superior Técnico, Universidade de Lisboa, Portugal

³ Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

{eugenio.ribeiro, ricardo.ribeiro, david.matos}@inesc-id.pt

Abstract

ISO 24617-2, the ISO standard for dialog act annotation, sets the ground for more comparable research in the area. However, the amount of data annotated according to it is still reduced, which impairs the development of approaches for automatic recognition. In this paper, we describe a mapping of the original dialog act labels of the LEGO corpus, which have been neglected, into the communicative functions of the standard. Although this does not lead to a complete annotation according to the standard, the 347 dialogs provide a relevant amount of data that can be used in the development of automatic communicative function recognition approaches, which may lead to a wider adoption of the standard. Using the 17 English dialogs of the DialogBank as gold standard, our preliminary experiments have shown that including the mapped dialogs during the training phase leads to improved performance while recognizing communicative functions in the *Task* dimension.

Keywords: dialog acts, iso standard, communicative functions, LEGO corpus

1. Introduction

It is valuable for a dialog system to identify the intention behind its conversational partners' words since it provides an important cue concerning the information contained in a segment and how it should be interpreted. According to Searle (1969), that intention is revealed by dialog acts, which are the minimal units of linguistic communication. Consequently, automatic dialog act recognition is an important task in the context of Natural Language Processing (NLP), which has been widely explored over the years. However, dialog act annotation has typically been performed in the context of projects or the development of datasets, each introducing new tag sets or modifying the existing ones. This led to a wide scattering of data in terms of annotation, which hardens the comparison of results and conclusions obtained using different automatic dialog act recognition approaches. In an attempt to set the ground for more comparable research in the area, a standard for dialog act annotation was developed (Bunt et al., 2012; Bunt et al., 2017). However, annotating dialogs according to this standard is an exhaustive process, especially since the annotation does not consist of a single label, which in the standard nomenclature is called a communicative function, but rather of a complex structure which includes information regarding the semantic dimension of the dialog act and relations with other segments, among others. Consequently, currently, the only publicly available annotated dialogs are those provided in the DialogBank (Bunt et al., 2016; Bunt et al., 2019) (Bunt et al., 2016). These are not enough for training models using the data-hungry deep learning approaches that are the current state-of-the-art on automatic dialog act recognition (Ribeiro et al., 2019a; Ribeiro et al., 2019b).

In this paper, we describe a mapping of the dialog act annotations of the LEGO corpus (Schmitt et al., 2012) (Schmitt et al., 2012) into the communicative functions of the standard. This corpus, which features human-machine dialogs

in the bus information domain, has been widely explored in dialog related tasks (Ultes et al., 2013; Griol et al., 2016). However, its original dialog act annotations have been neglected. This is due to the high domain dependence of the labels, which are only relevant for that specific dataset. However, such specificity also simplifies the mapping of the labels into the higher level domain-independent communicative functions of the standard. Although this mapping does not lead to a complete annotation according to the standard and not all semantic dimensions are covered, it enables the use of the corpus to improve the performance on automatic recognition of the communicative functions of the standard. Furthermore, since we publicly distribute the mapped annotations (Ribeiro et al., 2019), the corpus can be used in future research in the area, promoting a more widespread adoption of the standard.

In the remainder of this document, we start by describing the standard for dialog act annotation in Section 2. and the LEGO corpus in Section 3.. Then, Section 4. describes the mapping process and Section 5. analyzes the distribution of communicative functions in the mapped corpus. Section 6. describes our preliminary experiments that confirm that the mapped corpus can be used to improve the performance on automatic communicative function recognition. Finally, in Section 7., we discuss the most important contributions described in the paper and provide pointers for future work.

2. ISO Standard for Dialog Act Annotation

ISO 24617-2, the ISO standard for dialog act annotation (Bunt et al., 2012; Bunt et al., 2017), states that, in order to isolate intentions, the annotation should not be performed on turns or utterances, but rather on functional segments (Carroll and Tanenhaus, 1978). Furthermore, according to the standard, the dialog act annotation of a segment does not consist of a single label, but rather of a complex structure containing information about the participants, relations with other functional segments, the semantic di-

mension of the dialog act, its communicative function, and optional qualifiers concerning certainty, conditionality, partiality, and sentiment.

The standard defines nine semantic dimensions – *Task*, *Auto-Feedback*, *Allo-Feedback*, *Turn Management*, *Time Management*, *Discourse Structuring*, *Own Communication Management*, *Partner Communication Management*, and *Social Obligations Management* – in which different communicative functions may occur. These communicative functions are the standard equivalent to the dialog act labels used to annotate dialogs before the introduction of the standard. They are divided into general-purpose functions, which can occur in any semantic dimension, and dimension-specific functions. The set of general-purpose functions is organized hierarchically as shown in Figure 1. On the other hand, dimension-specific functions, listed in Table 1, are all at the same level. However, the five communicative functions specific to the *Social Obligations Management* dimension are sometimes specialized into their initial and return counterparts. Furthermore, there are no functions specific to the *Task* dimension. That is, it features general-purpose functions only.

Semantic Dimension	Communicative Functions
Auto-Feedback	Auto Positive Auto Negative
Allo-Feedback	Allo Positive Allo Negative Feedback Elicitation
Own Communication Management	Retraction Self Correction Self Error
Partner Communication Management	Correct Misspeaking Completion
Turn Management	Turn Accept Turn Assign Turn Grab Turn Keep Turn Release Turn Take
Time Management	Stalling Pausing
Discourse Structuring	Interaction Structuring Opening
Social Obligations Management	Greeting Self Introduction Apology Thanking Goodbye

Table 1: Dimension-specific communicative functions defined by the ISO standard for dialog act annotation.

Since automatic, and even manual, functional segmentation is a complex task on its own and the annotation process is non-trivial, the amount of data annotated according to the standard is reduced. To our knowledge, the only publicly available source of annotated dialogs is the DialogBank (Bunt et al., 2016; Bunt et al., 2019) (Bunt et al., 2016). It features (re-)annotated dialogs from four English corpora – MapTask (Anderson et al., 1991) (5 dialogs), Switchboard (Jurafsky et al., 1997) (4 dialogs), TRAINS (Allen and Schubert, 1991) (3 dialogs), and

DBOX (Petukhova et al., 2014) (5 dialogs) – and four Dutch corpora – DIAMOND (Geertzen et al., 2004) (3 dialogs), OVIS (Strik et al., 1997) (3 dialogs), MapTask (1 dialog), and Schiphol (Prust et al., 1985) (2 dialogs). There is a total of 17 annotated dialogs in English and 9 in Dutch. However, even though these dialogs are the closest we have to a gold standard, not all of them are segmented or annotated exactly as specified by the standard. Overall, this means that in order to disseminate the standard and perform solid experiments using its annotation methodology, it is important to put some effort into obtaining larger amounts of data annotated according to it.

3. LEGO Corpus

The LEGO corpus (Schmitt et al., 2012) (Schmitt et al., 2012) is an annotated subset of 347 calls from the Carnegie Mellon University (CMU)’s Let’s Go Bus Information System (Raux et al., 2006) recorded during 2006. It features 14,186 utterances – 9,083 system utterances and 5,103 user utterances. Since system utterances are generated through slot filling of fixed templates, they have no errors and contain casing and punctuation information. In contrast, the transcriptions of user utterances were obtained using an Automatic Speech Recognition (ASR) system and, thus, contain no casing nor punctuation information. Furthermore, the recognition was not always correct. However, a concrete value for the Word Error Rate (WER) is not revealed.

System Turns (9,083)		User Turns (5,103)	
Label	%	Label	%
Confirm Understood	10.04	Place Information	16.94
Confirm Departure	9.06	Unqualified / Unrecognized	15.09
Ask Another Query	8.01	Reject	14.38
Ask Bus	7.86	Line Information	8.48
Ask Confirm Time	5.47	Time Information	7.54
Ask Time	5.43	Confirm Departure	5.61
Ask Confirm Destination	5.04	Confirm Destination	4.74
Explain	4.82	Confirm Time	4.34
Ask Confirm Bus	4.68	Confirm	4.12
Ask Departure	4.51	Confirm Bus	3.45
Deliver Result	4.51	Request Next Bus	3.06
Filler	4.51	Reject Departure	2.60
Announce Querying	4.01	New Query	1.89
Offer Help	3.83	Reject Time	1.83
Greeting	3.82	Request Previous Bus	1.44
Ask Destination	3.80	Reject Bus	1.33
Inform No Schedule	3.78	Reject Destination	1.02
Ask Confirm With Keys	1.56	Request Help	1.00
Ask Confirm Neighborhood	1.41	Goodbye	0.59
Announce Restart	0.92	Request Schedule	0.35
Inform Shorter Answer	0.82	Polite	0.15
Inform Help	0.74	Inform	0.06
Goodbye	0.40		
Disambiguate Bus Stop	0.35		
Inform No Route	0.28		
Instruct Louder	0.11		
Confirm Restart	0.07		
Instruct More Quiet	0.01		

Table 2: Original label distribution in the LEGO corpus.

In terms of dialog acts, the LEGO corpus was originally annotated with on two distinct and domain-dependent tag sets for system and user turns. The set for system turns contains 28 labels, with the three most common being *Confirm Understood* (10%), *Confirm Departure* (9%), and *Ask Another Query* (8%). The set for user turns contains 22 labels, with the three most common being *Place Information* (17%),

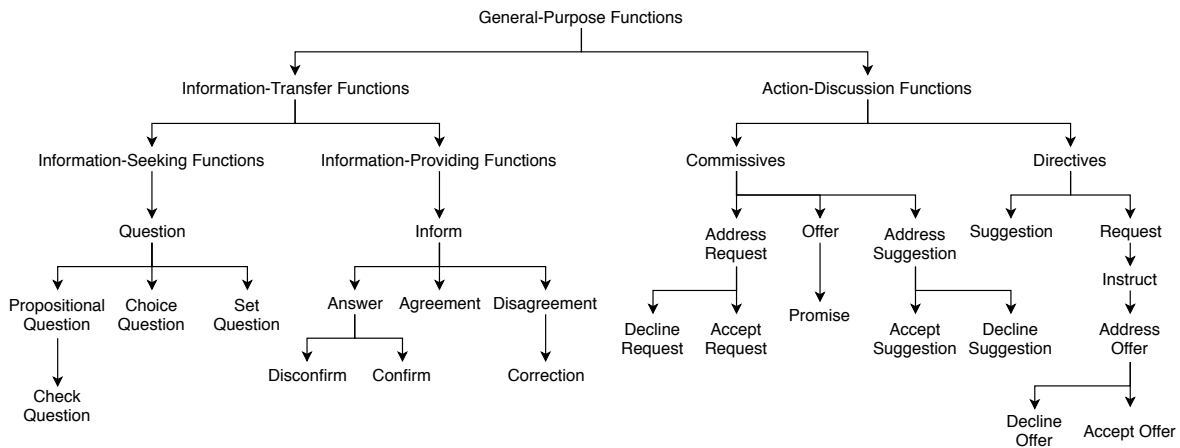


Figure 1: Hierarchy of general-purpose communicative functions defined by the ISO standard for dialog act annotation.

Unqualified / Unrecognized (15%), and *Reject* (14%). This reveals a high number of communication problems between the user and the system. The complete distribution of the labels in the corpus is shown in Table 2.

Although the LEGO corpus has been used to research many aspects related to dialog and interaction with Interactive Voice Response (IVR) systems (Ultes et al., 2013; Sidorov et al., 2014; Brester et al., 2015; Griol et al., 2016; Ribeiro et al., 2016), its dialog act annotations have been neglected. This is due to the high domain and task dependence of the labels, which are only relevant for this specific dataset. In fact, to our knowledge, their only relevant use was in the context of our study on dialog act recognition using character-level tokenization (Ribeiro et al., 2019a), in which they were used to show that sub-word information is relevant not only to identify generic dialog acts, but also domain-dependent dialog acts.

4. Label Mapping

As discussed in the previous section, the original dialog act annotations of the LEGO corpus have been neglected because they are too specific. However, that specificity also leads to an almost direct mapping of those labels into the higher level communicative functions of the ISO standard for dialog act annotation. This section describes that mapping, starting with the labels of system turns and then the labels of user turns.

Note that this mapping does not produce a complete dialog act annotation according to the standard for three main reasons. First, not all the communicative functions present in the turns are covered, since it is not possible to obtain information related to certain dimensions from the transcriptions and the original labels alone. Secondly, communicative functions are just part of the annotation. Thirdly, according to the standard, annotations are made at the functional segment level and not at the turn level. Still, the mapping of the original labels into the communicative functions of the standard is able to provide a significant amount of data for developing automatic communicative function recognition systems.

4.1. System Labels

Since the system turns of the LEGO corpus are generated through slot filling, each dialog act label is attributed to a set of templates. Thus, in order to map the labels into the communicative functions of the standard, we analyzed those templates and attributed them the corresponding functions. The strategies used for each label are presented below.

4.1.1. Announce Querying

The turns annotated with this label consist of techniques for pausing the dialog which fall under the *Time Management* dimension defined in the standard. Thus, we annotated them with the *Pausing* communicative function. Furthermore, in cases such as Example 2, the system makes a promise to look for some information. Those cases were also annotated with the *Promise* label in the *Task* dimension.

Examples:

1. “Just a second.”
2. “Hold on. I’ll look that up.”

4.1.2. Announce Restart

All turns annotated with this label contain the same utterance, which reveals the intention of the system to restart the current interaction. Thus, we annotated them with the *Interaction Structuring* communicative function in the *Discourse Structuring* dimension.

Example: “Okay, let’s start from the beginning.”

4.1.3. Ask Another Query

Turns annotated with this label state the different options available at the time, in an attempt to elicit one of the behaviors. These turns provide instructions to the user about the task and, thus, were annotated with the *Instruct* communicative function in the *Task* dimension. Furthermore, they are an attempt to structure the dialog and, thus, were also annotated with the *Interaction Structuring* communicative function in the *Discourse Structuring* dimension.

Examples:

1. “You can say, when is the next bus, when is the previous bus, start a new query, or goodbye.”

2. “To ask about a different trip, you can say, start a new query. If you are finished, you can say goodbye.”

4.1.4. Ask <Parameter>

In the turns annotated with this kind of label, the system prompts the user for the value of some parameter concerning the bus he or she wants to obtain information on. Thus, these turns correspond to *Set Questions* in the *Task* dimension.

Examples:

1. “What can I do for you?”
2. “What bus schedule information are you looking for?”
3. “Where are you leaving from?”
4. “What is your destination?”
5. “When do you want to travel?”

4.1.5. Ask Confirm <Parameter>

The turns annotated with this kind of label consist of two segments. The first states the information understood by the system, which is a case of auto-feedback, leading to annotation with the *Auto Positive* communicative function in the *Auto-Feedback* dimension. The second asks for confirmation by the user, which is a *Check Question* in the *Task* dimension. However, note that some of these turns seem to be wrongly annotated, as they correspond to instructions by the system on how to confirm or reject using the keys and, thus, should have the *Ask Confirm With Keys* label. These cases were converted using the strategy for that label, as described in Section 4.1.6.

Examples:

1. “The 54C. Did I get that right?”
2. “Leaving from Oakland. Is this correct?”
3. “Going to Fifth Avenue. Is this correct?”
4. “Leaving at 5 a.m.. Did I get that right?”

4.1.6. Ask Confirm With Keys

The turns labeled as *Ask Confirm With Keys* are similar to the previous in the sense that they state the understood information to provide feedback and request confirmation. However, instead of making a question, the system instructs the user on how to answer using the keypad. Thus, while in the *Auto-Feedback* dimension, these turns were also annotated with the *Auto Positive* communicative function, in the *Task* dimension, the *Instruct* communicative function is used instead of *Check Question*.

Example: “If you want the schedule of the 54C say yes or press one, otherwise say no or press three.”

4.1.7. Confirm Restart

Similarly to the turns annotated with the *Announce Restart* label (Section 4.1.2.), all turns annotated with the *Confirm Restart* label contain the same utterance. However, in this case, it is a check by the system on whether the user really wants to restart the interaction. Thus, we annotated these turns with the *Check Question* communicative function in

the *Task* dimension. Furthermore, since the system states that it understood a restart request, we also annotated these turns with the *Auto Positive* communicative function in the *Auto-Feedback* dimension.

Example: “Are you sure you want to start over?”

4.1.8. Confirm Understood / Filler

These turns signal the system’s understanding of the user’s intention through feedback utterances. Accordingly, we annotated them with the *Auto Positive* communicative function in the *Auto-Feedback* dimension.

Examples:

1. “Right.”
2. “Ok.”
3. “Alright.”

4.1.9. Deliver Result

In the turns annotated with this label the system provides the bus schedule information according to the parameters discussed along the dialog. Thus, we annotated them with the *Inform* communicative function in the *Task* dimension.

Example: “The next 61C leaves Eighth Avenue at Ann at 7:45 p.m. and arrives at Second Street at Grant at 7:59 p.m..”

4.1.10. Disambiguate Bus Stop

The turns annotated with this label occur when there is some confusion about the requested bus stops. Some cases, such as Example 1, correspond to questions that ask for a specific bus stop instead of a zone, while others, such as Example 2, request a different start or ending point. Thus, we annotated the first with the *Set Question* and the latter with the *Request* communicative functions in the *Task* dimension.

Examples:

1. “Which stop in Duquesne are you leaving from?”
2. “Downtown and Forbes are both the same stop. Please provide a different start or end point.”

4.1.11. Explain

The turns annotated with this label provide instructions and examples to the user according to what he or she can do at that time. Accordingly, we annotated them with the *Instruct* communicative function in the *Task* dimension.

Example: “For example, you can say, Forbes and Murray, Downtown, or McKeesport.”

4.1.12. Goodbye

These turns correspond to the system politely ending the dialog. Thus, they fall on the *Social Obligations Management* dimension of the standard and were annotated with the *Goodbye* communicative function.

Example: “Thank you for using the CMU Let’s Go Bus Information System. Goodbye.”

4.1.13. Greeting

All turns annotated with this label feature the same utterance, which opens the interaction and greets the user. Thus, we annotated them with the *Opening* communicative function in the *Discourse Structuring* dimension and with *Greeting* in the *Social Obligations Management* dimension.

Example: “Welcome to the CMU Let’s Go bus information system.”

4.1.14. Inform Help

Similarly to the turns annotated with the previous label, all turns annotated with the label *Inform Help* label contain the same utterance. In this case, that utterance informs the user of the buses that the system has information about. Thus, we annotated them with the *Inform* communicative function in the *Task* dimension.

Example: “I am an automated spoken dialogue system that can give you schedule information for bus routes in Pittsburgh’s East End. You can ask me about the following buses: 28X, 54C, 56U, 59U, 61A, 61B, 61C, 61D, 61F, 64A, 69A, and 501.”

4.1.15. Inform No Route / Inform No Schedule

The turns annotated with these labels state that there are no buses satisfying the indicated parameters and apologize for that. Thus, we annotated them with the *Inform* communicative function in the *Task* dimension and with the *Apology* communicative function in the *Social Obligations Management* dimension. However, many turns were wrongly annotated with the *Inform No Schedule* label and required manual and individual mapping.

Examples:

1. “I’m sorry, but there is no bus that goes between CMU and Squirrel Hill at that time.”
2. “I’m sorry but I do not have the schedule for the 500. The routes I currently cover are the following: 28X, 54C, 56U, 59U, 61A, 61B, 61C, 61D, 61F, 64A, 69A, and 501.”

4.1.16. Inform Shorter Answer

In these turns, the system asks the user to use shorter answers, both using polite requests, such as Example 1, and more assertive commands, such as Example 2. In the *Task* dimension, the polite requests were annotated with the *Request* communicative function, while the remaining were annotated with *Instruct*.

Examples:

1. “Please use shorter answers because I have trouble understanding long sentences.”
2. “I need you to give me a short answer.”

4.1.17. Instruct Louder / Instruct More Quiet

In turns annotated with these labels, the system asks the user to speak louder (Example 1) or quieter (Example 2) using a polite request. Thus, we annotated them with the *Request* communicative function in the *Partner Communication Management* dimension.

Examples:

1. “I’m having some trouble hearing you. If you’re still there, please try to talk a little bit louder or closer to the phone.”
2. “I can’t understand loud speech. Please speak more quietly.”

4.1.18. Offer Help

All turns annotated with this label contain the same utterance, which instructs the user on how to get help. Accordingly, we annotated them with the *Instruct* communicative function in the *Task* dimension.

Example: “To get help at any time, just say ‘Help’ or press zero.”

4.2. User Labels

Contrarily to system turns, the user turns are open and contain recognition errors. Thus, the mapping of their labels into the communicative functions of the standard is not as straightforward. Still, since the turns are typically short and the labels are highly domain-dependent, there are still mapping strategies that can be applied. For each label, they are presented below. Note that although the user turns do not contain casing nor punctuation information, we complemented them with that information in the examples.

4.2.1. Confirm <Parameter>

The turns annotated with this kind of label consist of user confirmations that the system understood correctly. Thus, we annotated them with the *Confirm* communicative function in the *Task* dimension. We also annotated them with the *Allo Positive* communicative function in the *Allo-Feedback* dimension, since they serve as feedback for the system.

Examples:

1. “Correct.”
2. “Yes.”

4.2.2. Goodbye

This label is the user counterpart of the same label for system turns (Section 4.1.12.). Consequently we also mapped it into the *Goodbye* communicative function in the *Social Obligations Management* dimension.

Example: “Goodbye.”

4.2.3. Inform

Only three turns are annotated with this label and one of them should be annotated with the *Time Information* label (Section 4.2.5.) instead. The remaining provide information unrelated to the system’s question and were annotated with the *Inform* communicative function in the *Task* dimension.

4.2.4. Line Information

The turns annotated with this label provide information about the bus that the user wants to obtain information about. However, this information may come in the form of a question, such as Example 1, or a statement, such as Example 2. Thus, in the *Task* dimension, these turns were annotated with either the *Set Question* or *Inform* communicative functions, respectively.

Examples:

1. “When is the next 28X from Downtown to the Airport?”
2. “The 61A.”

4.2.5. Place / Time Information

The turns annotated with these labels answer a system prompt for place (Example 1) or time (Examples 2 and 3) information. Since these are answers to specific questions, we annotated them with the *Answer* communicative function in the *Task* dimension.

Examples:

1. “Downtown.”
2. “Eleven o’clock.”
3. “Now.”

4.2.6. New Query

In these turns, the user instructs the system to start a new query, guiding the dialog in that direction. Thus, we annotated them with the *Instruct* communicative function in the *Task* dimension and the *Interaction Structuring* communicative function in the *Discourse Structuring* dimension.

Example: “Start a new query.”

4.2.7. Polite

In the turns annotated with this label, the user thanks the system for some reason. Thus, we annotated them with the *Thanking* communicative function in the *Social Obligations Management* dimension.

Example: “Thank you.”

4.2.8. Reject <Parameter>

The turns annotated with this kind of label consist of user rejections or corrections of the system’s understanding. Thus, we annotated them with the *Disconfirm* label in the *Task* dimension. Furthermore, we also annotated them with the *Allo Negative* communicative function in the *Allo-Feedback* dimension, since they serve as feedback for the system.

Examples:

1. “No.”
2. “No, I need the next bus.”

4.2.9. Request Help

In the turns annotated with this label, the user asks for help using the keyword “Help” or by pressing the corresponding numeric key. Thus, we annotated them with the *Request* communicative function in the *Task* dimension.

Example: “Help.”

4.2.10. Request Next Bus / Request Previous Bus

These turns consist of the user asking for information about the next or previous bus. This request may come in the form of a question, such as Example 1, or a statement, such as Example 2. Furthermore, when a statement is used, it may be in response to a time request by the system. Thus, in the *Task* dimension, we used three different communicative functions to annotate these turns. Respectively, *Set Question*, *Inform*, and *Answer*.

Examples:

1. “When is the next bus?”
2. “The previous bus.”

4.2.11. Request Schedule

These turns consist of the user stating that he or she wants schedule information. Thus, we annotated them with the *Inform* communicative function in the *Task* dimension.

Example: “Holiday schedule.”

4.2.12. Unqualified / Unrecognized

The turns annotated with these labels typically correspond to problems in the dialog. For instance, cases when the ASR system failed to recognize most of the sentence and only output a small part of it, cases of self-talk or third-party talk, and cases when the user says utterances unrelated to the task or that disrupt the dialog flow. Many of these turns are gibberish and do not correspond to any communicative function of the standard. However, around two thirds of them actually reveal an intention and, thus, should be annotated regardless of whether they make sense according to the flow of the dialog. Those turns were annotated manually with different communicative functions, with *Answer* and *Inform* in the *Task* dimension being the most common.

5. Corpus Analysis

By applying the mapping described in the previous section, the LEGO corpus gets the communicative function distribution shown in Table 3. As expected in a corpus featuring task-oriented dialogs, communicative functions in the *Task* dimension are predominant, with 78.15% of system turns and 92.87% of user turns having functions in that dimension. The lower percentage in system turns is due to the necessity of structuring the dialog to follow the system’s predefined interaction templates. The most frequent communicative functions in system turns are *Check Question* (24.84%), *Set Question* (21.88%), and *Instruct* (19.95%). In user turns, those are replaced by the *Answer* (28.70%), *Confirm* (22.81%), and *Disconfirm* (21.69%) functions. This is coherent with the nature of the dialogs, which typically consist of the system questioning the user to obtain the required information. All turns considered, 83.33% have communicative functions in the *Task* dimension, with the most frequent functions being the same as in system turns, but with lower impact – 15.92%, 15.50%, and 13.53%, respectively. In terms of the feedback dimensions, 44.50% of the user turns have communicative functions in the *Allo-Feedback* dimension and no functions in the *Auto-Feedback* dimension. In system turns, the values are reversed, with 41.99% having communicative functions in the *Auto-Feedback* dimension and only 1.77% in the *Allo-Feedback* dimension. Once again, these values are coherent with the nature of the dialogs, since the system typically uses feedback functions to check whether it understood what the user said, while the user confirms or disconfirms that.

As for the remaining semantic dimensions, it is important to refer that it is difficult to find communicative functions in those dimensions by simply converting the original labels of the LEGO corpus. Thus, the identified functions are just a

Dimension	Function	System (%)	User (%)
Task	Answer	0.00	28.70
	Check Question	24.84	0.02
	Confirm	0.00	22.81
	Disconfirm	0.00	21.69
	Inform	7.22	11.78
	Instruct	19.95	2.08
	Promise	3.05	0.00
	Request	0.59	1.67
	Set Question	21.88	4.12
	Suggest	0.44	0.00
	Total	78.97	92.87
Allo-Feedback	Allo Negative	0.00	21.69
	Allo Positive	0.00	22.81
	Total	0.00	44.50
Auto-Feedback	Auto Negative	0.00	0.37
	Auto Positive	41.99	1.39
	Total	41.99	1.77
Discourse Structuring	Interaction Structuring	9.38	2.02
	Opening	3.82	0.00
	Total	13.20	2.02
Social Obligations Management	Apology	1.79	0.06
	Goodbye	0.40	0.61
	Greeting	3.82	0.92
	Thanking	0.00	0.20
	Total	6.01	1.79
Time Management	Pausing	4.01	0.00
Partner Comm. Management	Request	0.18	0.00
	Total	0.18	0.00

Table 3: Distribution of ISO standard communicative functions in the LEGO corpus after the mapping.

small part of all those that exist in the corpus and that could be found through a manual and exhaustive processing of each turn. Still, it is interesting to note that at least 13.20% of the system turns contain communicative functions in the *Discourse Structuring* dimension. This reveals the rigid nature of the system’s utterances and its intention to structure the dialog according to a specific path.

6. Communicative Function Recognition

Since the mapping of the original dialog act labels of the LEGO corpus into the communicative functions of the standard does not lead to a complete dialog act annotation according to the standard, the mapped corpus cannot be used as a gold standard for future dialog act research. However, it is 13 times the size of DialogBank in number of dialogs, when considering all the dialogs. If only English dialogs are considered, it is 20 times the size of DialogBank. Thus, the mapped dialogs are relevant for the development of systems able to automatically recognize the communicative functions of the standard. In this section, we describe a set of preliminary experiments that confirm that ability.

6.1. Classification Approach

In our experiments, we use the same classification approach we used to achieve the best results on the automatic recognition of the original labels for user turns in the LEGO corpus (Ribeiro et al., 2019a). As revealed by the architecture shown in Figure 2, it is a deep learning approach that generates two complementary segment representations, one based on Word2Vec (Mikolov et al., 2013) representations of its words and another on its characters. In both cases, the

representations are generated by a set of three parallel Convolutional Neural Networks (CNNs) that capture functional patterns of different sizes – one, two, and three for words and three, five, and seven for characters. Additionally, context information is provided in the form of a flag stating whether the speaker changed in relation to the previous segment and the one-hot representations of the classifications of the three preceding segments. The two segment representations and the context information are then concatenated and passed through a dense layer that generates a reduced representation of the segment and applies dropout during the training phase to reduce the probability of overfitting. Finally, the reduced representation is passed through a dense layer with the softmax activation to identify the communicative function with highest probability.

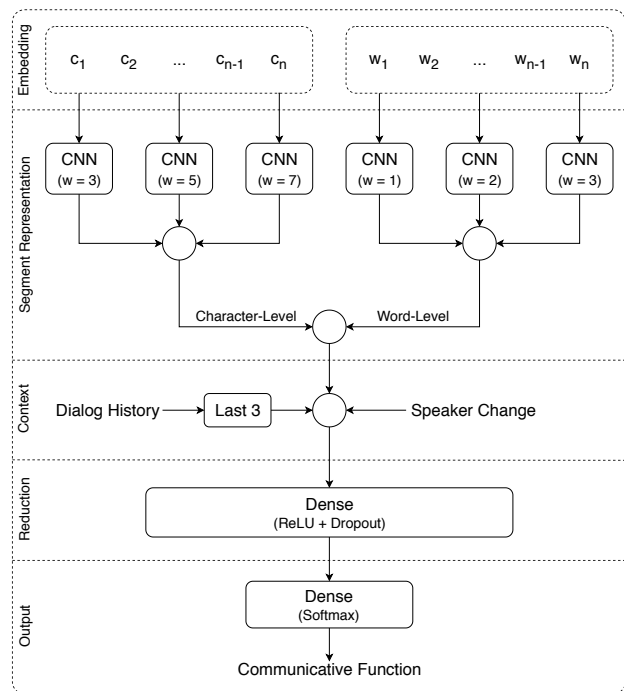


Figure 2: The architecture of the communicative function recognition approach.

6.2. Evaluation Methodology

To confirm that the mapped LEGO corpus can be used to improve the performance on automatic communicative function recognition, we performed preliminary experiments using the English dialogs in the DialogBank as gold standard. Since most semantic dimensions do not have a representative number of occurrences in the dialog, we focused on the *Task* dimension, which poses the more complex problem. We defined two scenarios regarding the automatic recognition of communicative functions in this dimension. The first focuses on the recognition of the different general-purpose functions in the segments that have communicative functions in the *Task* dimension. Thus, the remaining segments are discarded. The second scenario also considers the identification of segments which have communicative functions in the *Task* dimension. Thus, all segments are considered and a new label, *None*, is given to those which do not have communicative functions in that dimension.

Scenario	Data	MR		hP		hR		hF	
		μ	σ	μ	σ	μ	σ	μ	σ
Task	DialogBank	51.40	3.06	76.56	1.99	68.56	2.11	72.34	2.04
	DialogBank + LEGO	70.30	0.37	86.37	0.46	84.45	0.47	85.40	0.44
Task + None	DialogBank	73.53	0.28	79.51	0.16	62.36	0.73	69.90	0.48
	DialogBank + LEGO	74.87	0.20	78.05	0.47	69.78	0.48	73.69	0.38

Table 4: Results achieved while predicting *Task* dimension communicative functions in the DialogBank.

Since there are only 17 English dialogs in the DialogBank, we evaluated the performance using a leave-one-dialog-out cross-validation approach. That is, the communicative functions of each dialog in the DialogBank were predicted by a classifier trained on the remaining dialogs. To assess the performance improvement enabled by considering the mapped LEGO corpus, the same procedure was followed, but with the LEGO dialogs added to the training data of every classifier. In every case, the training phase stopped after ten epochs without improvement.

Since the general-purpose communicative functions pose a hierarchical classification problem, we report results in terms of exact match ratio (MR), as well as the hierarchical versions of precision (hP), recall (hR), and F-measure (hF) proposed by Kiritchenko et al. (2005). The latter are relevant because they capture the difference between predicting a label that shares part of its path with the correct label and one that follows a completely different path. Since there is some non-determinism involved in the training of the classifiers, we report the mean (μ) and standard deviation (σ) of the metrics achieved over ten runs, in percentage form.

6.3. Results

Table 4 shows the results of our experiments. Starting with the scenario that focuses on segments with communicative functions in the *Task* dimension, we can see that, without considering the LEGO dialogs, the average performance is of just 51.40% in terms of exact match ratio. However, by looking at the hierarchical metrics, we can see that most predictions follow at least part of the correct path and that most errors occur because a more abstract communicative function is predicted.

By considering the LEGO dialogs during the training of the classifiers, the average performance improves by 19 percentage points in terms of exact match ratio and 13 in terms of hierarchical F-measure. Furthermore, the difference between precision and recall is reduced to two percentage points, from the original eight. This reveals the importance of the mapped dialogs, which provide additional information that enables the prediction of more specific communicative functions without deviating from the correct path.

In the second scenario, which also considers the identification of segments that have communicative functions in the *Task* dimension, the difference between the results in terms of exact match ratio and hierarchical F-Measure is less pronounced. This occurs because the segments without communicative functions in the *Task* dimension have a label that is in the top level of the hierarchy. Furthermore, these segments are easier to predict, leading to an average exact

match ratio of 73.53%, even without considering the LEGO dialogs. However, in this case, the average hierarchical F-measure is of just 69.90%, which, in combination with the 62.36% recall, suggests that the need for identifying which segments have communicative functions in the *Task* dimension impairs the ability to identify the actual communicative functions, especially those which are more specific.

In this scenario, by considering the LEGO dialogs during the training phase, the improvement in terms of average exact match ratio is of just one percentage point. This is explainable by the high predominance of the *Task* dimension in the mapped dialogs, which impairs the ability to identify segments which do not have communicative functions in that dimension. However, the average hierarchical F-measure increases by four percentage points, which shows that the mapped dialogs are still useful. This improvement in terms of F-measure is due to the improvement of seven percentage points in terms of recall, which confirms that the additional information provided by the mapped dialogs improves the ability to recognize more specific communicative functions.

7. Conclusions

We have described a mapping of the original dialog act labels of the LEGO corpus into the communicative functions of the ISO standard for dialog act annotation. Although communicative functions alone do not make a complete dialog act annotation according to the standard, by performing this mapping, the 347 dialogs can now be used to improve the ability to recognize communicative functions. This is particularly relevant given the limited amount of existing data annotated according to the standard. We have confirmed this importance in our preliminary experiments, which have shown that considering the mapped dialogs during the training phase leads to improved performance while recognizing communicative functions in the *Task* dimension in the English dialogs of the DialogBank.

As future work, it is important to perform a more thorough study on automatic communicative function recognition, considering all semantic dimensions and including an error analysis. Also, the use of contextualized word representations and a summary of the dialog history should be explored (Ribeiro et al., 2019b). Finally, the performance on the recognition of general-purpose functions is expected to improve by using a hierarchical classifier that leverages the dependencies between the functions.

8. Acknowledgements

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT), under project UIDB/50021/2020.

9. Bibliographical References

- Allen, J. F. and Schubert, L. K. (1991). The TRAINS Project. Technical Report Technical Report 382 and TRAINS Technical Note 91-1, Computer Science Dept., University of Rochester.
- Anderson, A. H., Bader, M., Bard, E. G., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., et al. (1991). The HCRC Map Task Corpus. *Language and Speech*, 34(4):351–366.
- Brester, C., Semenkina, E., Sidorov, M., and Semenkina, O. (2015). Multicriteria Neural Network Design in the Speech-Based Emotion Recognition Problem. In ICINCO, volume 1, pages 621–628.
- Bunt, H., Alexandersson, J., Choe, J.-W., Fang, A. C., Hasida, K., Petukhova, V., Popescu-Belis, A., and Traum, D. (2012). Iso 24617-2: A semantically-based standard for dialogue annotation. In LREC, pages 430–437.
- Bunt, H., Petukhova, V., Malchanau, A., Wijnhoven, K., and Fang, A. (2016). The dialogbank. In LREC, pages 3151–3158.
- Bunt, H., Petukhova, V., Traum, D., and Alexandersson, J. (2017). Dialogue Act Annotation with the ISO 24617-2 Standard. In *Multimodal Interaction with W3C Standards*. Springer, pp. 109–135.
- Bunt, H., Petukhova, V., Malchanau, A., Fang, A., and Wijnhoven, K. (2019). The DialogBank: Dialogues with Interoperable Annotations. *Language Resources and Evaluation*, 53(2):213–249.
- Carroll, J. M. and Tanenhaus, M. K. (1978). Functional Clauses and Sentence Segmentation. *Journal of Speech, Language, and Hearing Research*, 21(4):793–808.
- Geertzen, J., Girard, Y., Morante, R., Van der Sluis, I., Van Dam, H., Suijkerbuijk, B., Van der Werf, R., and Bunt, H. (2004). The DIAMOND Project. In CATALOG.
- Griol, D., Iglesias, J. A., Ledezma, A., and Sanchis, A. (2016). A Two-Stage Combining Classifier Model for the Development of Adaptive Dialog Systems. *International Journal of Neural Systems*, 26(1):1650002.
- Jurafsky, D., Shriberg, E., and Biasca, D. (1997). Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coder Manual. Technical Report Draft 13, Institute of Cognitive Science, University of Colorado.
- Kiritchenko, S., Matwin, S., and Famili, A. F. (2005). Functional Annotation of Genes using Hierarchical Text Categorization. In BioLINK SIG.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In NIPS, pages 3111–3119.
- Petukhova, V., Gropp, M., Klakow, D., Schmidt, A., Eigner, G., Topf, M., Srb, S., Motlicek, P., Potard, B., Dines, J., Deroo, O., Egeler, R., Meinz, U., and Liersch, S. (2014). The DBOX Corpus Collection of Spoken Human-Human and Human-Machine Dialogues. In LREC, pages 252–258.
- Prust, H., Minnen, G., and Beun, R.-J. (1985). Transcriptie Dialogoexperiment Juni/Juli 1984. Technical Report IPO Rapport 481, Eindhoven University of Technology.
- Raux, A., Bohus, D., Langner, B., Black, A. W., and Eskenazi, M. (2006). Doing Research on a Deployed Spoken Dialogue System: One Year of Let’s Go! Experience. In INTERSPEECH, pages 65–68.
- Ribeiro, E., Batista, F., Trancoso, I., Lopes, J., Ribeiro, R., and Martins de Matos, D. (2016). Assessing User Expertise in Spoken Dialog System Interactions. In INTERSPEECH, pages 245–254.
- Ribeiro, E., Ribeiro, R., and Martins de Matos, D. (2019a). A Multilingual and Multidomain Study on Dialog Act Recognition using Character-Level Tokenization. *Information*, 10(3):94.
- Ribeiro, E., Ribeiro, R., and Martins de Matos, D. (2019b). Deep Dialog Act Recognition using Multiple Token, Segment, and Context Information Representations. *Journal of Artificial Intelligence Research*, 66:861–899.
- Schmitt, A., Ultes, S., and Minker, W. (2012). A Parameterized and Annotated Spoken Dialog Corpus of the CMU Let’s Go Bus Information System. In LREC, pages 3369–3373.
- Searle, J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.
- Sidorov, M., Brester, C., Semenkina, E., and Minker, W. (2014). Speaker State Recognition with Neural Network-Based Classification and Self-Adaptive Heuristic Feature Selection. In ICINCO, volume 1, pages 699–703.
- Strik, H., Russel, A., van den Heuvel, H., Cucchiari, C., and Boves, L. (1997). A Spoken Dialog System for the Dutch Public Transport Information Service. *International Journal of Speech Technology*, 2(2):119–129.
- Ultes, S., Schmitt, A., and Minker, W. (2013). On Quality Ratings for Spoken Dialogue Systems – Experts vs. Users. In NAACL-HLT, pages 569–578.

10. Language Resource References

- Harry Bunt and Kars Wijnhoven and Volha Petukhova and Andrei Malchanau and Alex Chengyu Fang. (2016). The DialogBank. Tilburg University, <https://dialogbank.uvt.nl/>.
- Eugénio Ribeiro and Ricardo Ribeiro and David Martins de Matos. (2019). LEGO-ISO. L²F - Spoken Language Systems Laboratory, INESC-ID Lisboa, <https://gitlab.hlt.inesc-id.pt/eugenio/lego-iso/>.
- Alexander Schmitt and Stefan Ultes and Wolfgang Minker. (2012). LEGO Spoken Dialogue Corpus. University of Ulm, <https://www.ultes.eu/ressources/lego-spoken-dialogue-corpus/>.