Journal of Artificial Intelligence Research 73 (2022) 397–436 Submitted 08/2021; published 01/2022

Automatic Recognition of the General-Purpose Communicative Functions defined by the ISO 24617-2 Standard for Dialog Act Annotation

Eugénio Ribeiro EUGENIO.RIBEIRO@INESC-ID.PT
INESC-ID Lisboa, Portugal
Instituto Superior Técnico, Universidade de Lisboa, Portugal

Ricardo Ribeiro RICARDO.RIBEIRO@INESC-ID.PT
INESC-ID Lisboa, Portugal
Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

David Martins de Matos DAVID.MATOS@INESC-ID.PT
INESC-ID Lisboa, Portugal
Instituto Superior Técnico, Universidade de Lisboa, Portugal

Abstract

From the perspective of a dialog system, it is important to identify the intention behind the segments in a dialog, since it provides an important cue regarding the information that is present in the segments and how they should be interpreted. ISO 24617-2, the standard for dialog act annotation, defines a hierarchically organized set of general-purpose communicative functions which correspond to different intentions that are relevant in the context of a dialog. We explore the automatic recognition of these communicative functions in the DialogBank, which is a reference set of dialogs annotated according to this standard. To do so, we propose adaptations of existing approaches to flat dialog act recognition that allow them to deal with the hierarchical classification problem. More specifically, we propose the use of an end-to-end hierarchical network with cascading outputs and maximum a posteriori path estimation to predict the communicative function at each level of the hierarchy, preserve the dependencies between the functions in the path, and decide at which level to stop. Furthermore, since the amount of dialogs in the DialogBank is small, we rely on transfer learning processes to reduce overfitting and improve performance. The results of our experiments show that our approach outperforms both a flat one and hierarchical approaches based on multiple classifiers and that each of its components plays an important role towards the recognition of general-purpose communicative functions.

1. Introduction

From the perspective of a dialog system, it is important to identify the intention behind the segments in a dialog, since it provides an important cue regarding the information that is present in the segments and how they should be interpreted. According to Searle (1969), that intention behind the uttered words is revealed by the corresponding dialog acts, which he defines as 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. In an attempt to set the ground for more comparable research in the area, Bunt et al. (2012) defined the ISO 24617-2 standard for dialog act annotation. However, annotating dialogs according to this standard is an exhausting process, especially since the annotation of each segment does not consist of a single dialog act 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 acts and relations with other segments, among other aspects. Consequently, the amount of data annotated according to the standard is still small and the automatic recognition of the whole set of communicative functions it defines remains practically unexplored.

In this article, we explore the automatic recognition of communicative functions in the English dialogs available in the DialogBank (Bunt et al., 2016, 2019), which, to the best of our knowledge, is the only publicly available source of dialogs fully annotated according to the standard. We focus on general-purpose communicative functions, since they are predominant and, contrarily to the dialog act labels of widely explored corpora in dialog act recognition research, they pose a hierarchical classification problem, with paths that may not end on a leaf communicative function.

To approach the problem, we propose modifications to existing approaches to automatic dialog act recognition that allow them to deal with the hierarchical classification problem posed by the general-purpose communicative functions defined by the ISO 25617-2 standard. These modifications focus on the ability to predict communicative functions at the multiple levels of the hierarchy, identify when the available information is not enough to predict more specific functions, and preserve the dependencies between the functions in the path. Furthermore, given the small amount of annotated dialogs provided by the DialogBank, we rely on pre-trained dialog act recognition models by using them in transfer learning processes. This way, we can take advantage of their ability to capture generic intention information and focus on identifying that which is most relevant for recognizing the general-purpose communicative functions defined by the standard.

In the remainder of the article, we start by providing an overview on the ISO 24617-2 standard in Section 2 and on automatic dialog act recognition approaches in Section 3. Then, in Section 4, we discuss the problem posed by the hierarchy of general-purpose communicative functions and describe our approach to the prediction of those functions. Section 5 describes our experimental setup, including the datasets, evaluation methodology, and baselines for comparison. The results of our experiments are presented and discussed in Section 6. Finally, Section 7 summarizes the contributions of the article and provides 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, 2017) aims at setting the ground for more comparable research in the area. According to it, dialog act annotations should not be performed on turns or utterances, but rather on functional segments (Carroll & Tanenhaus, 1978). Furthermore, the annotation of each segment does not consist of a single label or set of labels, but rather of a complex structure containing information about the participants, relations with other functional segments, the semantic dimension of the dialog act, its communicative function, and optional qualifiers concerning certainty, conditionality, and sentiment.

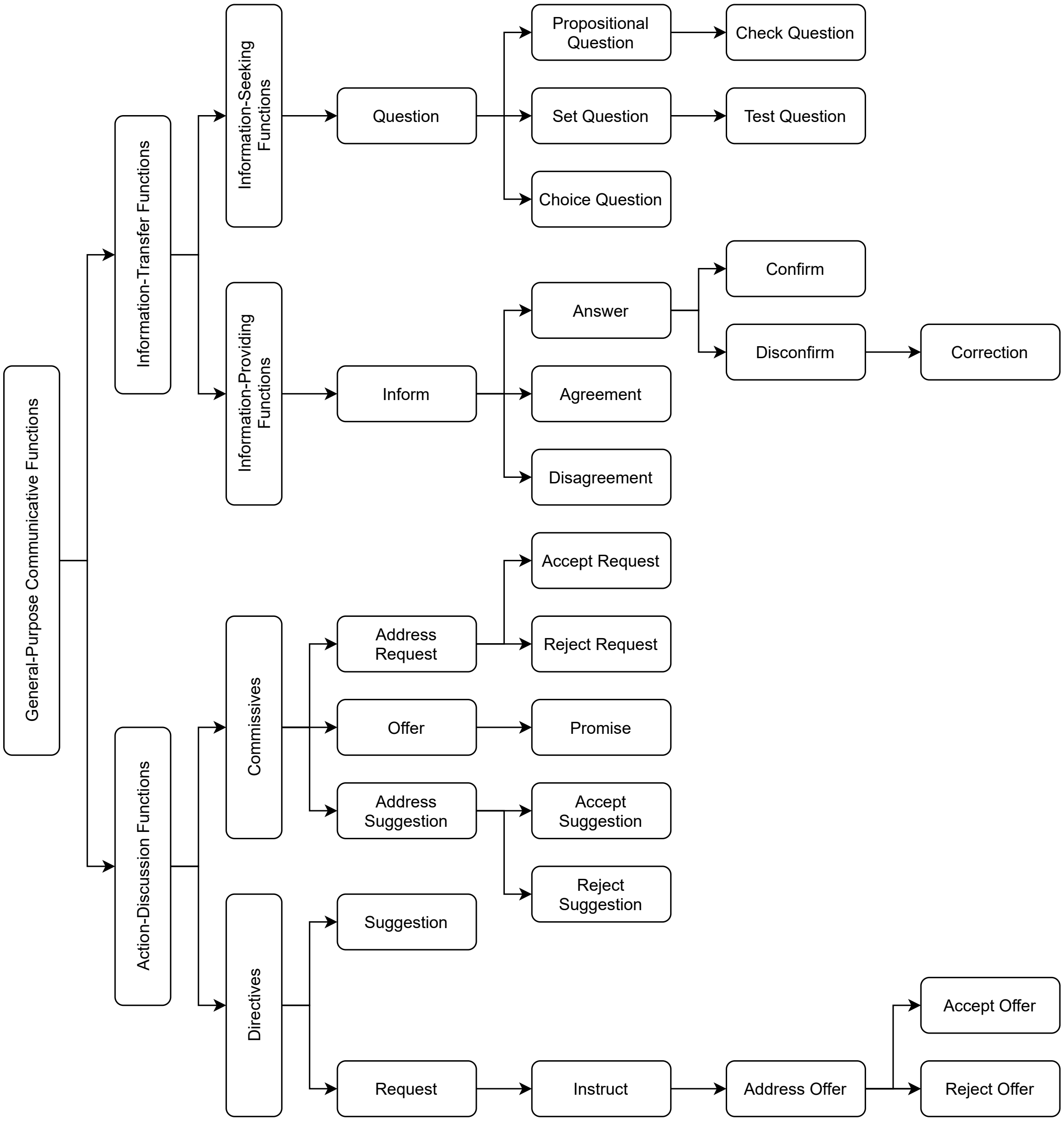

Figure 1: Hierarchy of general-purpose communicative functions defined by the ISO 24617-2 standard for dialog act annotation.

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 of the dialog act labels used to annotate dialogs before the introduction of the standard. They are divided into general-purpose and dimension-specific functions. The former can occur in any semantic dimension and are organized hierarchically as shown in Figure 1. The latter can only occur in the corresponding dimension and are distributed as shown in Table 1. Of the nine semantic dimensions, only the *Task* dimension does not have specific functions. This means that only general-purpose communicative functions occur in that dimension. Furthermore, with the exception of the functions specific to the *Social Obligations Management* dimension, which can be split into their initial and return counterparts, dimension-specific functions are all at the same level.

Annotating all of the aspects defined by the standard is an exhausting process and, consequently, the amount of available data is still reduced and, in many cases, not all of the aspects are considered (e.g. Petukhova et al., 2014; Bunt et al., 2016; Cerisara et al., 2018; Bunt et al., 2019; Anikina & Kruijff-Korbayová, 2019). To the best of our knowledge, the DialogBank (Bunt et al., 2016, 2019) is the only publicly available source of dialogs fully annotated according to the standard. It features (re)-annotated dialogs from several corpora with different characteristics, but, currently, there are only 15 dialogs in English and 9 in Dutch, which amount to less than 3,000 segments.

Considering this scarcity of publicly available annotated dialogs, we have mapped the dialog act labels of the LEGO corpus (Schmitt et al., 2012) into the communicative functions defined by the ISO 24617-2 standard and released them as LEGO-ISO (Ribeiro et al., 2020). The mapping was based solely on the original labels and the transcriptions of the turns. This means that the annotation is performed on turns rather than on functional segments and that it does not cover every semantic dimension, nor the full depth of the hierarchy of general-purpose communicative functions defined by the standard. Consequently, this dataset cannot be used as a gold standard. Still, it provides a significant amount of data that can be used to provide information regarding the generic intention behind the segments.

3. Automatic Dialog Act Recognition

Given a turn, utterance, or functional segment in a dialog, to which we will refer generically as segment in the remainder of the article, automatic dialog act recognition aims at identifying the intention behind that segment. This task has been widely explored over the years, using both classical machine learning and deep learning approaches. In both cases, the approaches differ mainly on how the representation of a segment is generated from the representations of its tokens and how they are able to weigh context information in the decision process. The article by Král and Cerisara (2010) provides a comprehensive overview on classical machine learning approaches on the task, except for the more recent Support Vector Machine (SVM)-based approaches (Gambäck et al., 2011; Ribeiro et al., 2015).

Regarding deep learning approaches, both Recurrent Neural Networks (RNNs) (e.g Lee & Dernoncourt, 2016; Ji et al., 2016; Khanpour et al., 2016; Tran et al., 2017b) and Convolutional Neural Networks (CNNs) (e.g. Kalchbrenner & Blunsom, 2013; Lee & Dernoncourt,

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 24617-2 standard for dialog act annotation.

2016; Liu et al., 2017) have been used to generate segment representations by combining the embedding representations of their words. While the first focus on capturing information from relevant sequences of tokens, the latter focus on the context surrounding each token and, thus, on capturing relevant patterns independently of where they occur in the segment. We have compared different RNN- and CNN-based representation approaches and achieved higher performance using a set of parallel CNNs with different window sizes, while also consuming less resources than when using RNNs (Ribeiro et al., 2019b). Still, most of the more recent studies on the task rely on bidirectional RNNs for segment representation (e.g. Kumar et al., 2018; Chen et al., 2018; Li et al., 2019; Raheja & Tetreault, 2019).

Regarding the representation of the segment's tokens, most approaches to dialog act recognition using deep learning have relied on pre-trained word embedding representations generated by Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). However, similarly to what happens on most NLP tasks, using contextualized word representations, especially those generated by BERT (Devlin et al., 2019), leads to higher performance (Ribeiro et al., 2019b). Additionally, there are studies which also rely on character-level information (e.g. Bothe et al., 2018; Chen et al., 2018; Li et al., 2019; Raheja & Tetreault, 2019). In our studies on that matter, we observed that there are complementary cues for intention at a sub-word level which cannot be captured when relying solely on word-level tokenization and that punctuation is also an important cue for certain dialog acts (Ribeiro et al., 2018a, 2019a). Finally, functional-level tokenization, mainly in the form of Part-of-Speech (POS) tags, has also been explored for the task (Chen et al., 2018; Ribeiro et al., 2019b). However, it seems to be less relevant than the remainder.

In terms of context information, previous studies have mostly focused on information regarding the speakers and the dialog itself. Speaker information is typically used to capture the turn-taking history, which enables the identification of different intentions according to whether the previous segment was uttered by the same speaker (Liu et al., 2017; Ribeiro et al., 2019b; Zhao & Kawahara, 2019). Additionally, Wang et al. (2020) relied on the the whole dialog history by each speaker to identify speaker-dependent cues for specific intentions. Regarding the contents of the dialog, Li et al. (2019) have observed performance improvements when the topics covered by the dialog were provided as context information. Still, the most important source of context information for the identification of the intention behind a segment is the surrounding segments. Studies dedicated to that matter have shown that the influence of preceding segments decreases with the distance and that their dialog act classifications are more informative than their words, even when the classifications are obtained automatically (Ribeiro et al., 2015; Liu et al., 2017). Furthermore, we have shown that sequentiality information and long distance dependencies among the preceding segments can be captured by using an RNN to generate a summary of their classifications (Ribeiro et al., 2019b).

In fact, considering the dependencies between the multiple segments in a dialog, several studies attempted to predict the sequence of dialog acts in a complete dialog by approaching the task as a sequence labeling problem (e.g. Bothe et al., 2018; Kumar et al., 2018). These cases rely on a hierarchical approach in which the representations of the segments are provided to a conversation-level RNN that models the whole dialog. Furthermore, the performance can be improved by including attention mechanisms that identify the information in the surrounding segments that is most relevant for predicting the dialog acts (e.g. Tran et al., 2017b; Chen et al., 2018; Li et al., 2019; Raheja & Tetreault, 2019). Tran et al. (2017c) also observed that propagating uncertainty information concerning the previous predictions can lead to the prediction of better dialog act sequences. Finally, most of the studies that approach the task as a sequence labeling problem also rely on Conditional Random Fields (CRFs) (e.g. Kumar et al., 2018; Chen et al., 2018; Li et al., 2019; Raheja & Tetreault, 2019) or generative models (Tran et al., 2017a) as the final layer to predict the sequence of dialog acts. This way, the prediction of the dialog act for a segment is further conditioned to the previous predictions. Overall, this kind of approach achieves the highest performance on the task. However, the conversation-level RNN is typically bidirectional. This means that the prediction of the dialog act for a segment relies not only on the information from previous segments, but also from future ones, which are not available to a dialog system during the conversation.

In one of the most recent studies, Żelasko et al. (2021) explored the joint segmentation into functional segments and dialog act classification using Transformers, achieving higher performance than when using approaches based on bidirectional RNNs. Furthermore, they analyzed the importance of punctuation, context information, and label set size for the joint task. Overall, they concluded that, as observed in previous studies, all three are important for dialog act classification. Additionally, the first two, and especially punctuation, are also important for improving the performance on segmentation.

Additional studies on dialog act recognition explore alternative approaches or focus on specific applications. For instance, Ravi and Kozareva (2018) focused on developing dialog act recognition models that can be used in mobile applications. Wan et al. (2018) approached the task as a Question Answering (QA) problem and employed adversarial training. Ren and Xue (2020) trained siamese networks with a triplet loss to generate segment representations with larger distances between classes. Qin et al. (2020) addressed the joint detection of dialog acts and sentiment using co-attention mechanisms to capture mutually important information.

The automatic recognition of the communicative functions defined by the ISO 24617-2 standard has only been addressed in a reduced number of studies. Furthermore, these studies typically focused on small subsets of communicative functions and none of them explored the automatic recognition of the complete hierarchy of general-purpose communicative functions. Still, we summarize them below.

Anikina and Kruijff-Korbayová (2019) have explored the recognition of a compressed set of eight communicative functions on the TRADR corpus (Kruijff-Korbayová et al., 2015). This compressed set merges general-purpose functions and dimension-specific functions of the *Turn Management* and *Feedback* dimensions and does not consider the hierarchical nature of general-purpose functions. The authors compared the performance of several Deep Neural Network (DNN) architectures and uncontextualized embedding approaches and observed the highest performance when the representation of the segment was generated by passing GloVe embeddings through a Long Short-Term Memory Unit (LSTM). However, the use of parallel CNNs was not explored in their experiments and only large window sizes were considered. Furthermore, similarly to what was observed in previous studies on dialog act recognition, using context information regarding the dialog history led to improved performance. However, in this case, it was summarized as the average of the embedding representations of all the words in the dialog history.

Blache et al. (2020) have addressed an even smaller set of communicative functions consisting of the *Inform* and *Question* general-purpose communicative functions and three additional functions – *Opening*, *Closing*, and *Discourse* – which merge functions in the *Discourse Structuring*, *Social Obligations Management*, and *Feedback* dimensions. The authors explored the use of several classical machine learning approaches and achieved the best results using a simple logistic regression applied to Term Frequency – Inverse Document Frequency (TF-IDF) and morphosyntactic features and a set of domain-specific features based on occurrences of medical terms. The use of context information from the previous segment was also explored, but did not improve the performance in this case because the five communicative functions are typically easy to distinguish.

Cerisara et al. (2018) have explored the joint recognition of dialog acts and sentiment on the Mastodon dataset, which is annotated with a subset of the ISO 24617-2 communicative functions including general-purpose functions, dimension-specific functions of the *Feedback* and *Social Obligations Management* dimensions, and additional functions for specific cases. However, several of the communicative functions were merged together, leading to a reduced set of 15 functions. To approach the task they relied on a bidirectional LSTM to generate segment representations from uncontextualized word embeddings and on a conversation-level RNN to model the contextual dependencies between segments. The dialog act and sentiment are then predicted by independent branches of the network. Overall, the authors have observed that the joint modeling of the two tasks does not lead to improved performance in comparison to when using independent models. However, the joint modeling proved important for cross-task information transfer when the number of segments annotated for sentiment was reduced. Furthermore, a strong correlation was observed for some specific patterns. For instance, the sentiment tends to stay the same after an agreement.

Wang et al. (2021) have addressed the recognition of infrequent communicative functions using a hierarchical network with a two-pass attention mechanism. More specifically, they grouped the 15 communicative functions used to annotate the Mastodon dataset and 14 of the communicative functions used in the DialogBank into 4 coarse-grained communicative functions: *Inform*, *Question*, *Social*, and *Proposal*. The network was then jointly trained to predict the coarse- and fine-grained communicative functions of each segment. The attention mechanism is used to decorate the segment representation with information regarding the coarse-grained functions, so that it can be leveraged for the prediction of the fine-grained functions. This approach led to improved performance in comparison to when the coarse-grained functions were not considered, not only in terms of the recognition of infrequent communicative functions, but also globally. Furthermore, the performance increased when the representation of the segments was generated using the BERT model instead of a bidirectional LSTM applied to GloVe word embeddings. Finally, additional performance gains were obtained by pre-training the network to predict the coarse-grained communicative functions on a large domain-independent dataset.

In order to assess the utility the LEGO-ISO dialogs annotated using label mapping processes, we have performed preliminary experiments on the automatic recognition of general-purpose communicative functions in the DialogBank (Ribeiro et al., 2020). In those experiments, we flattened the hierarchy and addressed the task as a flat dialog act recognition problem using the same approach we used to predict the original dialog act labels of LEGO corpus (Ribeiro et al., 2019a). Then, we experimented with including the LEGO-

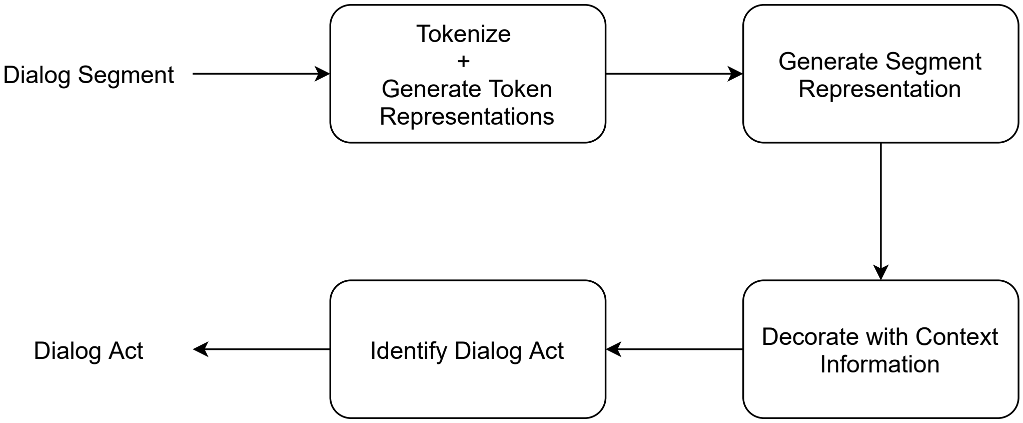

Figure 2: The generic dialog act recognition process.

ISO dialogs to provide additional information during the training phase while still tuning the hyperparameters to achieve the highest performance on the DialogBank dialogs. We observed performance improvements when the LEGO-ISO dialogs were included, which suggests that they can be used to generate more general segment representations in settings with a scarce amount of annotated data.

4. General-Purpose Communicative Function Recognition

The main difference between general-purpose communicative function recognition and traditional dialog act recognition is that the former poses a hierarchical classification problem, with paths that may not end on a leaf. However, both are intention recognition problems at their core. Thus, we approach the problem by adapting existing dialog act recognition approaches to deal with hierarchical problems. This way, we build on the ability of those approaches to capture generic information regarding intention. Furthermore, this allows us to explore the use of existing pre-trained models in transfer learning processes, in an attempt to minimize the impact of the scarcity of annotated data.

Although the studies on dialog act recognition covered in Section 3 explored different aspects that are relevant for the task, most approaches can be summarized as the generic four-step process shown in Figure 2. Given a segment in a dialog, the first step towards the identification of the main dialog act that it communicates is to split it into its constituent tokens and generate adequate representations for each of them. As discussed in Section 3, the tokenization is typically performed at the word-level. However, other tokenizations, such as at the character or functional levels, can be used to provide compositional or complementary information. In such cases, the different tokenizations of the segment can either be combined into a single one (e.g. the representation of a word combined with the representation of its characters) or considered to be independent branches in the subsequent steps of the process. The next step is to generate a representation of the segment by combining the representations of its tokens. Ideally, this representation should focus on capturing the characteristics of the segment that are relevant to identify the intention that it transmits. Then, the representation is decorated with context information regarding the dialog history and speaker information. This context information can be provided in the form of external features or by the recognition process itself in a recurrent manner. Finally, the information provided by the decorated segment representation is used to identify the dialog act communicated by the segment.

Since the segment representation decorated with context information focuses on providing information that allows the identification of the intention behind the segment, all the steps towards its generation are relevant for the recognition of both traditional dialog acts and general-purpose communicative functions. Consequently, the adaptation of existing dialog act recognition approaches to the recognition of general-purpose communicative functions refers to how that decorated segment representation can be specialized to allow the identification of the hierarchically structured functions.

A possible approach, which does not require modifications to the architecture of existing dialog act recognition approaches, is to simply flatten the hierarchy. This way, the recognition of general-purpose communicative functions can be approached as a flat single-label classification problem. However, this approach assumes that all communicative functions are at the same level. Thus, information regarding the relations between each function and its ancestors and descendants cannot be leveraged and each segment can only be used as an example of the terminal communicative function in its path.

Another approach to hierarchical classification problems is to train multiple classifiers, each specialized on a part of the hierarchy, and then use a combination of their predictions to obtain the final classification (Silla Jr. & Freitas, 2011). For instance, one can train one classifier per level of the hierarchy, one classifier per set of siblings, or even one classifier per node in the hierarchy and then apply a top-down prediction approach starting at the root of the hierarchy. The advantage of using an approach of this kind is that the relations between communicative functions can be leveraged when combining the predictions of the multiple classifiers and at least partially during training. However, the need for multiple classifiers increases complexity and the amount of resources required for training.

In an attempt to combine the advantages of these approaches while minimizing their disadvantages, we propose to use a single end-to-end classifier that jointly predicts the communicative functions at each level of the hierarchy while leveraging information regarding the dependencies between them. Both the flat and the multiple-classifier approaches will be used as baselines for performance comparison and will be described in further detail in Section 5.4. The details of our approach are described below.

The top performing dialog act recognition approaches are based on deep neural networks. In this context, when dealing with the multi-class single-label classification problems posed by most dialog act annotations, the output layer applies the *softmax* activation function to obtain a probability distribution of the classes. The dialog act of a given segment is then predicted by selecting the class with highest probability. As shown in Figure 3, in order to consider the hierarchical structure of the general-purpose communicative functions defined by the ISO 24617-2 standard, we propose to use an output layer per level of the hierarchy instead of using a single output layer. This way, each output layer focuses on distinguishing between communicative functions at the corresponding level without having to deal with the ambiguity caused by functions that are ancestors or descendants of each other.

Additionally, we introduce a specialization layer per level, which is a fully connected layer that, as the name suggests, specializes the decorated segment representation by cap-

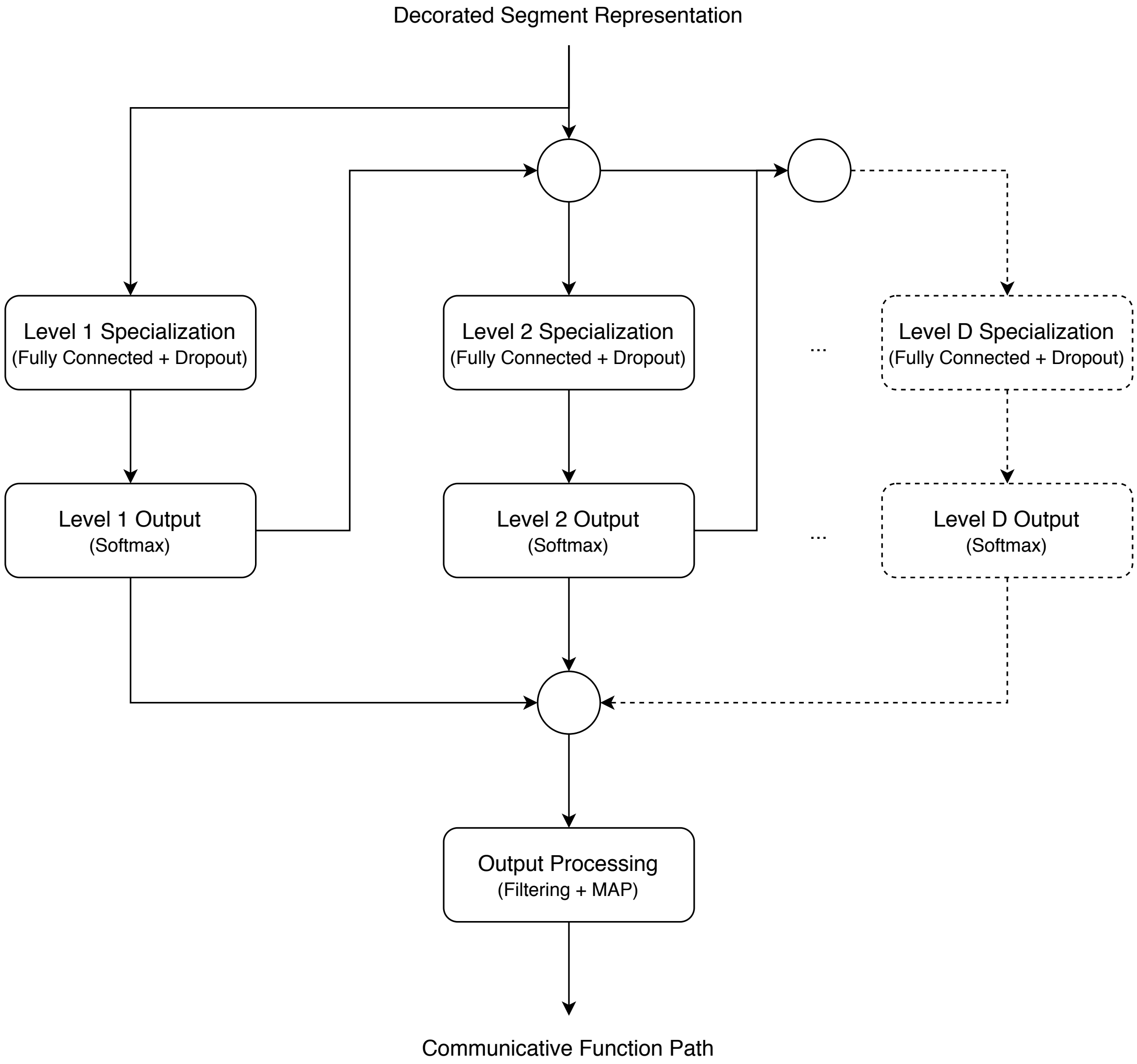

Figure 3: Our adaptation of a generic dialog act recognition approach to deal with the hierarchical problem posed by the ISO 24617-2 general-purpose communicative functions. The input is a segment representation decorated with context information. The circles represent concatenation operations.

turing the information that is most relevant for distinguishing the communicative functions at that level of the hierarchy. Furthermore, these layers are used to reduce the probability of overfitting by applying dropout (Srivastava et al., 2014) during the training phase. The use of specialization layers has already been proved important in our studies on automatic dialog act recognition (Ribeiro et al., 2018b, 2019b), including those on the DIHANA corpus (Benedí et al., 2006), which is annotated for dialog acts at three different levels.

The final modification to existing network architectures refers to the use of cascading outputs, that is, the probability distribution predicted by the network at a given level is appended to the decorated segment representation before it is passed to the specialization layer of the next level. This way, the network can capture information regarding the hierarchical dependencies between communicative functions at different levels.

The general-purpose communicative functions defined by the ISO 24617-2 standard follow a strict hierarchy, the classification of a segment does not necessarily end on a leaf, and the leaves are not all at the same level. Thus, the network must be able to predict paths with variable length. To approach this problem, we add an additional label, *None*, to each level of the hierarchy, to represent that there is no communicative function attributed to the segment at that level. This way, we are able to simulate paths with fixed length, while introducing minimal impact on the network during the training phase. The drawback is that the *None* label may become the predominant one in the deeper levels, biasing the output layers towards its prediction. These additional labels are also considered when providing context information to the network, in order to have fixed dimensionality.

During the inference phase, the parent-child relations between the general-purpose communicative functions must be considered in order to avoid predicting an invalid path. This means that, when selecting the label at a given level, only the children of the label selected for the level above it can be considered. As stated in the paragraph regarding hierarchical classification approaches that rely on multiple classifiers, this restriction can be enforced using a top-down prediction approach. That is, the prediction of the path starts by selecting the communicative function with highest probability at the top level and then applies a mask on the predictions of the level below it, in order to discard the communicative functions that are not children of the selected one. This process is then repeated for each level of the hierarchy, until a leaf communicative function or the *None* label is reached.

The main disadvantage of the top-down prediction approach is that its performance is highly impaired when misclassifications occur in the upper levels of the hierarchy. To attenuate the impact of such misclassifications, we explore a prediction approach based on Maximum a Posteriori (MAP) estimation in which the predicted communicative function for a given segment, s, is given by Equation 1:

$$\text{CommunicativeFunction}(s) = \underset{f \in F}{\text{argmax}} \prod_{d=1}^{D} P(L_d = \text{Path}(f)_d \mid s), \tag{1}$$

where F is the set of general-purpose communicative functions, D is the depth of the hierarchy, and $P(L_d = c \mid s)$ is given by the *softmax* output corresponding to label c of the output layer corresponding to the level at depth d. That is, instead of iteratively selecting a communicative function at each level, we compute the posterior probability of all valid paths in the hierarchy and select that with highest probability.

Since the classification of a segment does not necessarily end on a leaf and the leaves are not all at the same level, we also rely on the additional label *None* labels during the inference phase. For instance, given a segment s, the probability of selecting the path that ends in the *Answer* communicative function is given by Equation 2:

$$\begin{aligned} P(\mathit{Answer} \mid s) = {} & P(L_1 = \textit{Information-Transfer Functions} \mid s) \\ & \times P(L_2 = \textit{Information-Providing Functions} \mid s) \\ & \times P(L_3 = \mathit{Inform} \mid s) \\ & \times P(L_4 = \mathit{Answer} \mid s) \\ & \times P(L_5 = \mathit{None} \mid s) \\ & \times P(L_6 = \mathit{None} \mid s). \end{aligned} \tag{2}$$

5. Experimental Setup

This section describes our experimental setup, including the datasets (Section 5.1), the evaluation methodology (Section 5.2), the network architecture used in our experiments (Section 5.3), and the baselines used for comparison (Section 5.4). To conclude, in Section 5.5, we describe implementation details that allow future reproduction of our experiments.

5.1 Datasets

As discussed in Section 2, to the best of our knowledge, the DialogBank (Bunt et al., 2016, 2019) is the only publicly available source of dialogs annotated fully according to the ISO 24617-2 standard guidelines. Thus, in our experiments, we use it as gold standard for evaluating the performance of the different approaches. However, given the scarcity of annotated dialogs, we also explore the use of the LEGO-ISO dataset (Ribeiro et al., 2020), which features dialogs that were partially annotated with the communicative functions defined by the ISO 24617-2 standard through label mapping processes. Finally, we also rely on the Switchboard Dialog Act Corpus (Jurafsky et al., 1997) to train models for transfer learning purposes. Although its dialogs are not annotated according to the ISO 24617-2 standard, it is the largest and most explored corpus in domain-independent dialog act recognition studies. Thus, its dialogs can provide important information for the generation of segment representations that capture generic intention information. These datasets are described in further detail below.

5.1.1 The DialogBank

The DialogBank (Bunt et al., 2016, 2019) aims at collecting and providing dialogs annotated fully according to the ISO 24617-2 standard guidelines. At the time of this study, it features (re-)annotated dialogs from four English corpora and four Dutch corpora. There is a total of 15 annotated dialogs in English and 9 in Dutch. To avoid the issues regarding multilinguality, we focus on the English dialogs in this study. Three of those dialogs are originally from MapTask (Anderson et al., 1991), four are from Switchboard (Godfrey et al., 1992), three are from TRAINS (Allen & Schubert, 1991), and five are from DBOX (Petukhova et al., 2014). Excerpts extracted from each corpus are shown in Figures 4-7. In total, the dialogs contain 2,360 annotated segments, out of which 1,118 have general-purpose communicative functions in the *Task* dimension.

SPEAKER 1: And do you have a graveyard?
SPEAKER 2: No.
SPEAKER 1: Do you have a carved wooden pole in this area?
SPEAKER 2: Right away down the very bottom.
SPEAKER 1: Right, okay.
SPEAKER 1: Now, go... Go right from the diamond mine.
SPEAKER 2: Okay.

Figure 4: An excerpt of the transcription of a MapTask dialog. For readability, we added punctuation and capitalization.

SPEAKER 1: Jimmy, so, how do you get most of your news?
SPEAKER 2: Well, I kind of, uh, I watch the, uh, national news every day, for one.
SPEAKER 2: I also read one or two papers a day.
SPEAKER 2: And I'm a... I'm pretty much a news junkie.
SPEAKER 2: And I tune in to CNN a lot.
SPEAKER 1: Oh, wow.
SPEAKER 1: So, when you say the morning news, or evening news or national news is when?
SPEAKER 2: Uh, every evening at six thirty, I believe, I watch the national news.

Figure 5: An excerpt of the transcription of a Switchboard dialog. For readability, we removed disfluency annotations and replaced them with appropriate punctuation when necessary.

The general-purpose communicative functions are distributed in the DialogBank according to Table 2. We can see that, overall, the distribution is highly unbalanced, with the most common, *Inform*, covering 42% of the segments that have a general-purpose communicative function, while 10 of the functions that occur in the DialogBank occur in less than 10 segments. The predominance of the *Inform* communicative function becomes even more apparent if we consider the paths in the hierarchy. *Answer*, *Agreement*, *Confirm*, *Disconfirm*, and *Disagreement* are descendants of *Inform*. This means that of the segments that have a general-purpose communicative function, 62% have the *Inform* function.

Another important aspect revealed in Table 2 is that the distribution of general-purpose communicative functions is also highly unbalanced across the dialogs of the different corpora

Function	**MapTask**	**Switchboard**	**TRAINS**	**DBOX**	**Total**
Inform	56	338	44	37	475
Instruct	143	0	1	11	155
Answer	35	30	16	31	112
Propositional Question	26	11	2	25	64
Set Question	7	12	13	28	60
Accept Request	52	0	1	1	54
Agreement	8	42	2	1	53
Check Question	28	9	7	6	50
Confirm	11	9	6	14	40
Suggest	3	3	2	5	13
Disconfirm	1	1	0	10	12
Request	2	2	1	4	9
Choice Question	3	0	1	4	8
Correction	2	0	0	1	3
Address Request	3	0	0	0	3
Offer	0	0	0	2	2
Reject Offer	0	0	0	1	1
Disagreement	1	0	0	0	1
Accept Offer	0	0	0	1	1
Accept Suggest	0	0	0	1	1
Promise	0	0	0	1	1
General-Purpose CFs	381	457	96	184	1,118
None	281	555	140	266	1,242
Total	662	1,012	236	450	2,360

Table 2: Distribution of the general-purpose communicative functions defined by the ISO 24617-2 standard in the DialogBank.

USER: I need to design a plan for shipping a boxcar of oranges to Bath by eight a.m. today and the time now is midnight.
SYSTEM: Okay.
USER: The orange warehouse where I need the oranges from is in Corning.
SYSTEM: Right.
USER: So, I need... Is it possible for one of the engines... Would it be faster for an engine to come from Elmira or Avon?
SYSTEM: Uh, Elmira is a lot closer.
USER: What time would engine two and three leave Elmira?
SYSTEM: Um, well, they're not scheduled yet, but we can send them at any time we want.
USER: Okay.

Figure 6: An excerpt of the transcription of a TRAINS dialog. The system was simulated using the Wizard-of-Oz method. For readability, we removed disfluency annotations and added punctuation and capitalization.

USER: Uhm... Are you from America?
SYSTEM: No, I am not from America.
USER: Uh... What were you doing when you were alive?
SYSTEM: Ehm... Let me think for a minute... Uhm, I was... I was a princess when I was alive.
USER: And where are you from?
SYSTEM: I am from England.
USER: Okay, uhm... Uh... Where did you die?
SYSTEM: Just a second... Uhm... I died in Paris.
USER: Mhm... Uhn... Uhm... You are, uh... Princess Diana.

Figure 7: An excerpt of the transcription of a DBOX dialog. The system was simulated using the Wizard-of-Oz method. For readability, we removed disfluency annotations and added punctuation and capitalization.

that are included in the DialogBank, even in terms of the most common communicative functions. For instance, not considering paths, 92% of the segments with the *Instruct* and 96% with the *Accept Request* communicative functions belong to MapTask dialogs. On the other hand, 71% of the segments with the *Inform* communicative function belong to Switchboard dialogs. This reveals the heterogeneity of the DialogBank which is representative of the different kinds of dialog that occur in human-human and human-machine interaction.

Overall, although it includes dialogs from multiple corpora, the amount of data provided by the DialogBank is not enough for drawing solid conclusions from the results of DNN-based approaches trained solely on it, especially considering the hierarchical nature of the general-purpose communicative functions that we intend to recognize automatically. However, since these dialogs are the closest we have to a gold standard annotation, the evaluation of our approaches on general-purpose communicative function recognition must be based on the performance on the DialogBank.

5.1.2 LEGO-ISO

LEGO-ISO (Ribeiro et al., 2020) consists of 347 dialogs from the Let's Go Bus Information System (Raux et al., 2006), containing 14,186 utterances annotated with the communicative functions defined by the ISO 24617-2 standard. Each dialog features the system and a human user. Since the 9,803 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 the 5,103 user utterances were obtained using an Automatic Speech Recognition (ASR) system and, consequently, are subject to recognition errors and contain no casing nor punctuation information. However, a concrete value for the transcription Word Error Rate (WER) was not revealed. An excerpt of a dialog is shown in Figure 8.

SYSTEM: Welcome to the CMU Let's Go bus information system.
SYSTEM: To get help at any time, just say Help or press zero.
SYSTEM: What can I do for you?
USER: 54C.
SYSTEM: The 54C. Did I get that right?
USER: Yes.
SYSTEM: Ok.
SYSTEM: Where do you wanna leave from?
USER: South Hills Junction.

Figure 8: An excerpt of the transcription of a LEGO-ISO dialog. For readability, we added punctuation and capitalization.

The annotation with the standard's communicative functions was obtained through the mapping of the original dialog act annotations of the LEGO corpus (Schmitt et al., 2012). The mapping was based solely on the original labels and the transcriptions of the turns. This means that the annotation is performed on turns rather than on functional segments and that it does not cover every semantic dimension. Table 3 shows the distribution of general-purpose communicative functions in the corpus after the mapping. We can see that, although the number of segments is larger, the set of communicative functions covered by the corpus is only a subset of that covered by the DialogBank. This is partially due to the

Function	**System Segments**	**User Segments**	**Total**
Check Question	2,256	1	2,257
Set Question	1,987	210	2,197
Instruct	1,812	106	1,918
Answer	0	1,462	1,462
Inform	656	600	1,256
Confirm	0	1,162	1,162
Disconfirm	0	1,105	1,105
Promise	277	0	277
Request	54	85	139
Suggest	40	0	40
General-Purpose CFs	7,082	4,731	11,813
None	2,001	372	2,373
Total	9,083	5,103	14,186

Table 3: Distribution of the general-purpose communicative functions defined by the ISO 24617-2 standard in the LEGO-ISO corpus.

label mapping process, which did not consider the specificities of certain segments, but, most importantly, it is due to the characteristics of the dialogs, which are highly focused on the task. Thus, system segments typically have a communicative function that is a descendant of *Question*, so that it can obtain all the information required to fulfill the task. On the other hand, user segments typically have a communicative function that is a descendant of *Inform*, since they aim at providing that information to the system. Furthermore, the most common communicative function is *Check Question* because the system tries to confirm that it understood every piece of information provided by the user correctly. Also given the focus on the task, only 17% of the segments do not have a general-purpose communicative function, which contrasts with the 53% of the DialogBank.

Since its communicative-function annotations were obtained through label mapping processes, this dataset cannot be used as a gold standard. Still, it is 20 times larger than the DialogBank in number of English dialogs and 6 times larger in number of annotated segments. Thus, it provides a significant amount of data that, according to the results of our preliminary studies (Ribeiro et al., 2020), can be used during the training phase to improve the performance on general-purpose communicative function recognition. However, given its size in comparison to the DialogBank, classifiers may overfit to its characteristics.

5.1.3 Switchboard Dialog Act Corpus

The Switchboard Dialog Act Corpus (Jurafsky et al., 1997) is an annotated subset of the Switchboard (Godfrey et al., 1992) corpus. It is the largest and most explored corpus annotated with dialog act information, consisting of 1,155 manually transcribed conversations, containing 223,606 segments. The conversations are between pairs of humans and cover multiple domains. An excerpt of a dialog is shown in Figure 5. The corpus is annotated for dialog acts using the domain-independent SWBD-DAMSL tag set, which features over 200 unique labels. However, most studies use a reduced set of 42 to 44 labels to obtain a higher inter-annotator agreement and higher example frequencies per class. Table 4 shows this set of labels and its distribution in the corpus. The total number of segments labeled with a dialog act is lower than the previously referred 223,606 since some segments are considered continuations of the previous one by the same speaker and aggregated to it.

Label	**Segments**	**Label**	**Segments**
Statement-Non-Opinion	72,824	Collaborative Completion	699
Acknowledgement	37,096	Repeat-Phrase	660
Statement-Opinion	25,197	Open-Question	632
Agreement	10,820	Rhetorical-Question	557
Abandoned	10,569	Hold	540
Appreciation	4,663	Reject	338
Yes-No-Question	4,624	Negative Non-No Answer	292
Non-Verbal	3,548	Non-understanding	288
Yes Answer	2,934	Other Answer	279
Conventional Closing	2,486	Conventional Opening	220
Uninterpretable	2,158	Or-Clause	207
Wh-Question	1,911	Dispreferred Answers	205
No Answer	1,340	3rd-Party-Talk	115
Response Acknowledgement	1,277	Offers / Options	109
Hedge	1,182	Self-talk	102
Declarative Yes-No-Question	1,174	Downplayer	100
Other	1,074	Maybe	98
Backchannel-Question	1,019	Tag-Question	93
Quotation	934	Declarative Wh-Question	80
Summarization	919	Apology	76
Affirmative Non-Yes Answer	836	Thanking	67
Action Directive	719		
Total			195,061

Table 4: Dialog act distribution in the Switchboard Dialog Act Corpus.

Although the corpus is not annotated according to the ISO 24617-2 standard, its dialog act annotations and the communicative functions defined by the standard reveal similar intentions. For instance, regarding general-purpose communicative functions, labels such as *Yes-No-Question*, *Yes Answer*, and *No Answer*, can be directly mapped into the *Propositional Question*, *Confirm*, and *Disconfirm* communicative functions, respectively. Mappings of this kind are possible for several other labels and not only to general-purpose communicative functions, but also to dimension-specific functions. Fang et al. (2012) provide further insight into the possible mapping between the dialog act labels used to annotate the Switchboard Dialog Act Corpus and the communicative functions defined by the ISO 24617-2 standard. However, the corpus with mapped annotations was not released.

Given the size of the corpus and the similarity of the intentions revealed by its dialog act label set, we use it in our experiments to train a flat dialog act recognition model, so that its weights can be used in transfer learning processes for the generation of segment representations. This way, the probability of overfitting the representations to the characteristics of the training dialogs is reduced, which may improve the generalization ability of the classifiers and improve the overall performance.

5.2 Evaluation Methodology

In this study, we focus on the recognition of ISO 24617-2 general-purpose communicative functions in the DialogBank. Below, we describe the evaluation scenarios, the evaluation approach, and the metrics.

5.2.1 Scenarios

Although general-purpose communicative functions may occur in any of the semantic dimensions defined by the ISO 24617-2 standard, we focus on the *Task* dimension, since the number of occurrences of general-purpose communicative functions in the remaining dimensions is not representative in the DialogBank. In this context, we defined two evaluation scenarios. While the first considers every segment in the DialogBank, the other only considers the segments that have communicative functions in the *Task* dimension. In the first scenario, the segments which do not have communicative functions in that dimension are given the *None* label. By considering these two scenarios, we are not only able to assess the overall ability to identify segments with general-purpose communicative functions in a dialog, but also to focus on the multiple levels of the hierarchy and the ability to recognize those functions in more detail, highlighting the capabilities of our hierarchical approach.

5.2.2 Cross-Validation

Given the small number of dialogs in the DialogBank, it is not feasible to split it into partitions for training, development, and testing. Thus, we evaluate performance using two cross-validation approaches. The first is leave-one-dialog-out cross-validation, that is, the predictions for the segments in each dialog are made by classifiers trained on all the remaining dialogs. We use this as our main evaluation approach because it maximizes the amount of gold standard data available for training.

The second evaluation approach, leave-one-corpus-out cross-validation, takes advantage of the fact the DialogBank features dialogs from multiple corpora. In this case, the predictions for the segments in each dialog do not rely on training information from other dialogs in the same corpus. Thus, to an extent, this approach can be used to assess cross-corpora generalization capabilities.

To keep the evaluation as fair as possible, contrarily to what we did when assessing the ability of the LEGO-ISO dialogs to provide relevant information for training a communicative function recognizer (Ribeiro et al., 2020), we do not perform any fine tuning to maximize the performance on the left out dialog(s) in each fold. Instead, in each fold, we train an ensemble of classifiers on the corresponding training dialogs. Each of the classifiers in the ensemble is fine-tuned to maximize the performance on one of those training dialogs, while being trained on the remainder. This way, we remove the impact of selecting a single dialog for fine-tuning. The predicted classification of each segment in the left out dialog(s) is then given by a weighted majority vote of the classifiers in the ensemble. The weights are given by the estimated probability for the predicted path and ties are broken randomly.

In the experiments that use the LEGO-ISO dialogs, their segments are included in the training set of every fold and are never considered for testing.

5.2.3 Metrics

The most common metric used to evaluate dialog act recognition approaches is accuracy. Its counterpart in the context of hierarchical classification problems is the exact match ratio (MR), which is defined by Equation 3:

$$\mathrm{MR} = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i), \tag{3}$$

where n is the number of evaluation examples, Y_i is the set of labels in the gold standard path of example i, Z_i is the set of labels in the path predicted by the classifier for the same example, and I is the indicator function. We use the exact match ratio (MR) nomenclature to avoid confusion with the accuracy metric in the context of hierarchical problems, which considers each label in the paths independently.

In addition to the exact match ratio (MR), we also report results in terms of the hierarchical versions of precision (hP), recall (hR), and F-measure (hF) proposed by Kiritchenko et al. (2005), defined by Equations 4, 5, and 6, respectively:

$$\mathrm{hP} = \frac{\sum_{i=1}^{n} |Y_i \cap Z_i|}{\sum_{i=1}^{n} |Z_i|}, \tag{4}$$

$$\mathrm{hR} = \frac{\sum_{i=1}^{n} |Y_i \cap Z_i|}{\sum_{i=1}^{n} |Y_i|}, \tag{5}$$

$$\mathrm{hF} = \frac{2 * \mathrm{hP} * \mathrm{hR}}{\mathrm{hP} + \mathrm{hR}}. \tag{6}$$

These hierarchical metrics are relevant, since they consider partial path matches and, thus, 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.

In some of our experiments, we use additional metrics to assess the performance in terms of each communicative function in the hierarchy. More specifically, we rely on the traditional metrics of precision and recall.

The use of the ensemble of classifiers for prediction already attenuates the impact of the non-determinism involved in the training of neural networks. Still, we perform three runs of each experiment and report the average (mean) and standard deviation of the results. To improve readability, the values of every metric are reported in percentage form.

5.3 Network Architecture

As discussed in Section 4, our approach to deal with the hierarchical problem posed by the general-purpose communicative functions defined by the ISO 24617-2 standard can be applied on top of any approach to generate segment representations. In our experiments, we relied on the same approach used in our study that explored the multiple aspects that contribute to dialog act recognition (Ribeiro et al., 2019b). This way, we know that the segment representation approach is able to capture information regarding intention and we can use the models trained during that study in transfer learning processes.

Figure 9 shows the complete architecture of the network used in our experiments. Two representations of the segment are generated in parallel, one based on its characters and another on contextualized embedding representations of its words, generated by BERT (Devlin et al., 2019). In both cases, the representation of the segment is generated by concatenating the outputs of three parallel CNNs with different window sizes followed by a max-pooling operation. At the character level, we use windows of size three, five, and seven, in order to focus on affixes, lemmas, and inter-word relations. At the word level, we use windows of size one, two, and three, in order to focus on independent words and short word patterns. The two representations are then concatenated and decorated with context information.

Regarding context information, considering that the number of dialogs in the DialogBank is small, we do not rely on a summary of the whole dialog history as in the original approach, because it is prone to overfitting. Instead, we use a flattened sequence of classifications and turn-taking information of the three preceding segments, which have been proved the most important in previous studies (Ribeiro et al., 2015; Liu et al., 2017). The classification of each preceding segment is represented as a concatenation of the one-hot representations of the communicative functions at each level of the hierarchy. Turn-taking information is provided as flags stating whether the speaker changed.

The specialization and output layers are as described in Section 4. That is, there are dedicated specialization and output layers per each level in the hierarchy, the specialization layer of each level also considers the output at the upper levels, and each output layer considers an additional class representing the lack of communicative function at that level.

In order to take advantage of the intention information captured by a dialog act recognition model trained on a large corpus, we apply a transfer learning process. More specifically, we preset the weights of the segment representation layers of our hierarchical model using the corresponding weights of the flat dialog act recognition model. Consequently, only the specialization and output layers are trained on the dialogs annotated according to the ISO 24617-2 standard. This way, the segment representations provide generic information regarding intention that is then specialized for the distinction among general-purpose communicative functions at each level.

5.4 Baselines

As discussed in Section 4, we use the flat and multiple classifier approaches to hierarchical classification as baselines for comparison with our approach. In every case, the classifiers are based on the architecture described in Section 5.3. As shown in Figure 10, the difference is that there is only one specialization layer and one output layer instead of one pair per level of the hierarchy. The activation function and the number of communicative functions considered by the output layer varies according to the target of each classifier. The prediction approach also varies among the flat and multiple classifier approaches. These are described in further detail below.

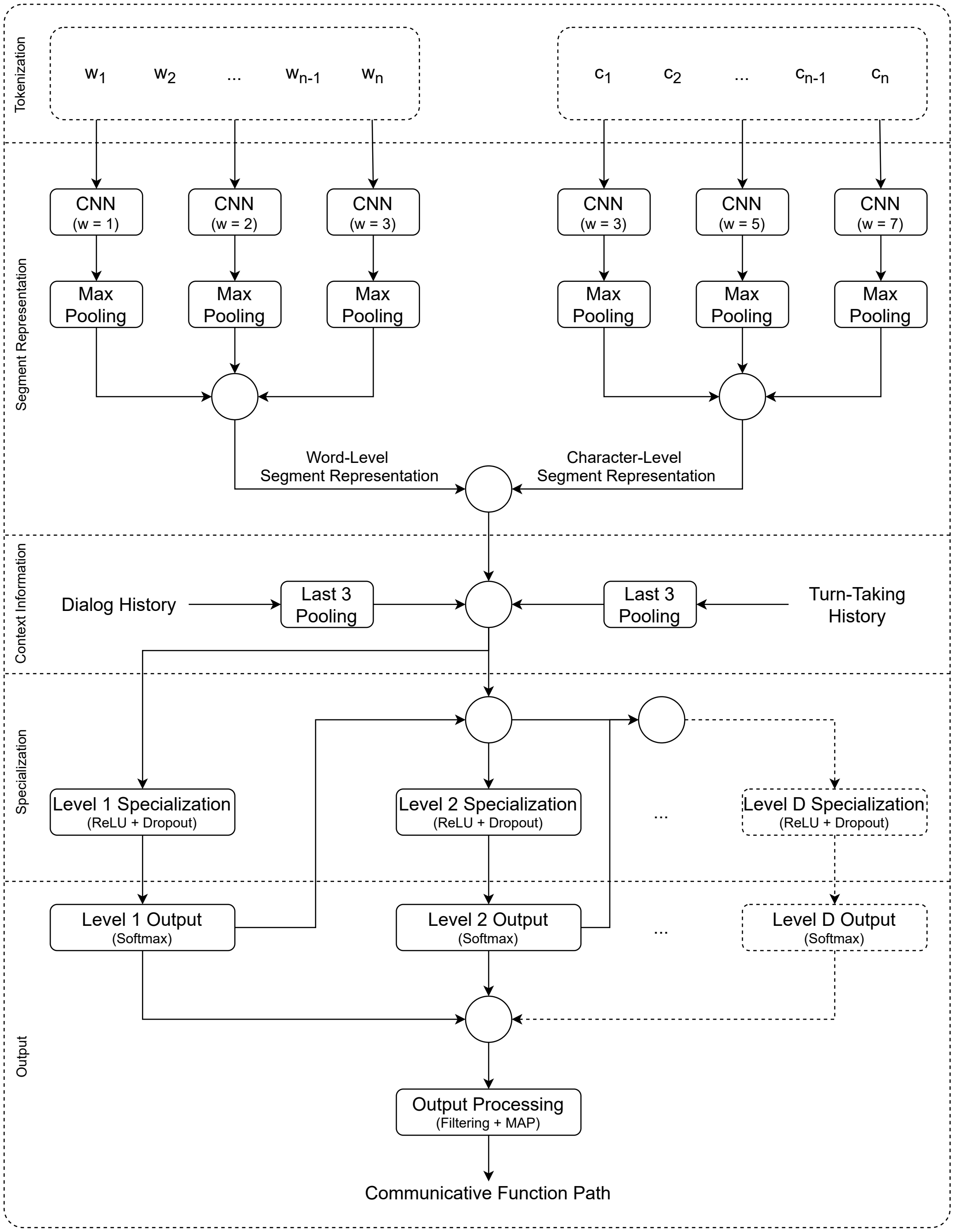

Figure 9: The full architecture of the automatic communicative function recognition approach used in our experiments. w_n and c_n refer to the embedding representation of the n-th word and character, respectively. The representations of words are generated by BERT (Devlin et al., 2019) and, thus, are contextualized. w in the CNNs refers to the width of the context window. D refers to the depth of the hierarchy. The circles represent concatenation operations.

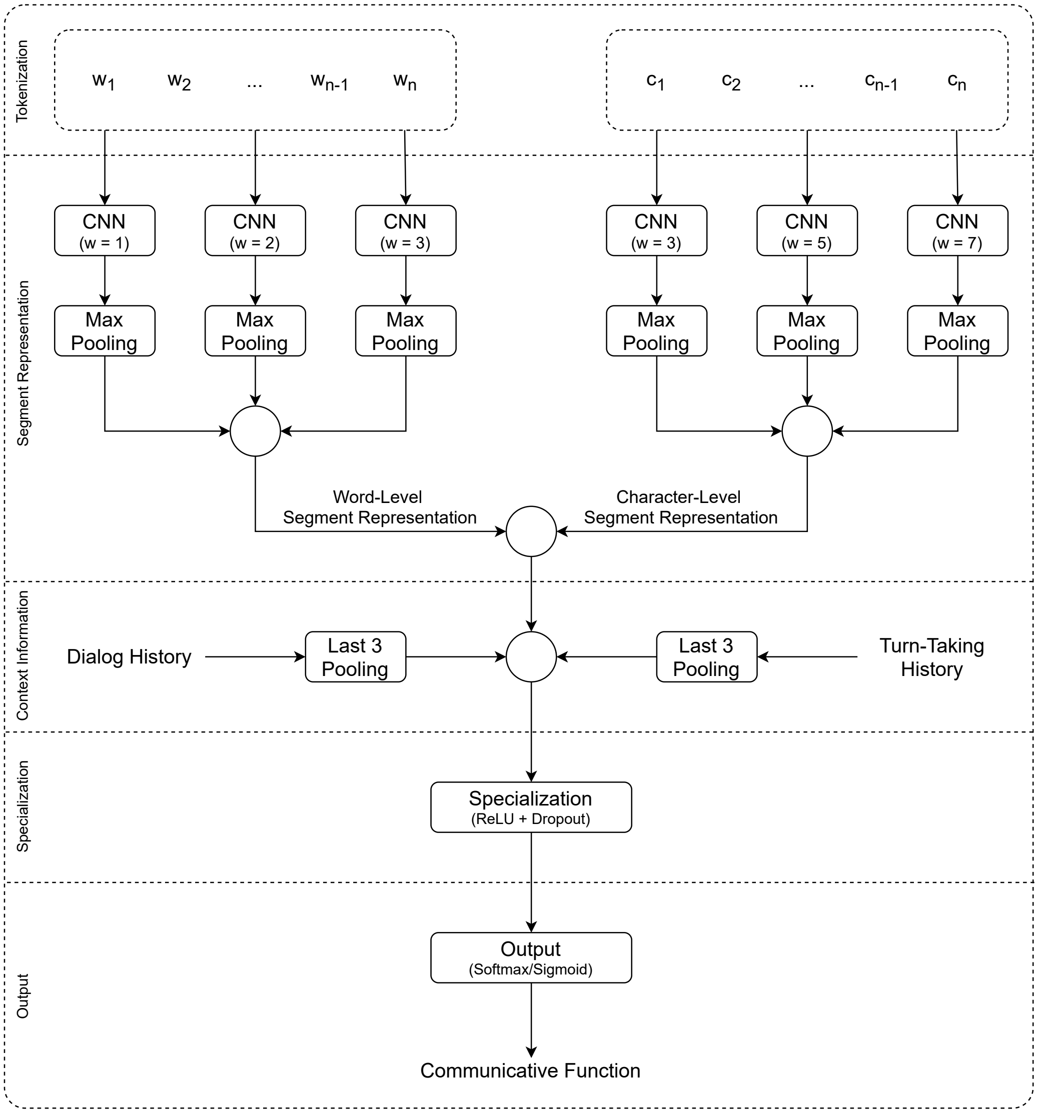

Figure 10: The architecture of the flat communicative function recognition approach used in our baselines. It differs from the hierarchical architecture shown in Figure 9 in the specialization and output layers, as there is only of each instead of one pair per level of the hierarchy.

The flat approach discards the hierarchical relations between communicative functions and considers that they are all at the same level. In this case, the output layer applies the *softmax* function to compute the probability distribution among the terminal communicative functions (the last in the path) that occur in the DialogBank. The predicted communicative function is that with highest probability. In a sense, this approach simplifies the problem, as it does not consider every possible communicative function. However, it is not able to leverage information regarding the hierarchical dependencies and each segment can only be used as an example of its terminal communicative function.

We use three multiple classifier approaches as baselines: one-vs-all classifiers, one classifier per level, and one classifier per set of siblings. In the first, each classifier focuses on identifying segments with a specific communicative function. Thus, each classifier is binary and the *sigmoid* function is used in the output layer to obtain the probability that a segment has the corresponding function. In the other two, each classifier focuses on distinguishing among a set of communicative functions. Thus, the *softmax* function is used to compute the probability distribution among those functions. In the three cases, since the classifiers do not distinguish among communicative functions at different levels, each segment can be used as an example of every function in its path. To combine the predictions of the multiple classifiers and obtain the predicted communicative function path for a segment, we apply a top-down approach. That is, starting at the root of the hierarchy, we select the communicative function with highest probability and then the process is repeated among its descendants, until a leaf communicative function or the stop case is reached. When using one-vs-all classifiers, the selection is performed by comparing the probabilities predicted by the classifiers corresponding to a set of sibling communicative functions. The prediction stops when none of them has a probability above 50%. When using one classifier per level, the predictions are masked according to the prediction in the level above, in order to avoid predicting invalid paths. When using either one classifier per level or per set of siblings, the prediction stops when the *None* label is that with highest probability. In order to attenuate the bias towards the prediction of shallower communicative functions, when training the classifiers for these two approaches, only the segments that have a communicative function in the level above but not in the current one are used as examples of the *None* label.

5.5 Implementation Details

To implement our classifiers, we used Keras (Chollet et al., 2015) with the TensorFlow (Abadi et al., 2015) backend. To update the weights during the training phase, we used the Adam optimizer (Kingma & Ba, 2015) with the default parameterization and mini-batches with size 512. To decide when to stop training, we used early stopping with 10 epochs of patience. That is, the training phase of each classifier stopped after ten epochs without improvement on the validation set composed of the corresponding fine-tuning dialog(s).

To obtain contextualized word representations, we used the output of the last layer of the large uncased BERT model (Devlin et al., 2019). When using character-level tokenization, embedding representations of the characters were trained together with the network to capture relations between them. To generate the segment representations, we used 100 filters in each CNN and aggregated the results using the max-pooling operation. Finally, the specialization layers were implemented as Rectified Linear Units (ReLUs) (Nair & Hinton, 2010) with 200 neurons and 50% dropout probability (Srivastava et al., 2014).

For transfer learning purposes, we relied on the model that achieved the top performance on the Switchboard Dialog Act Corpus in our study on dialog act recognition (Ribeiro et al., 2019b). Thus, the weights of the CNNs and the character-level embedding layer were set to those of that model and fixed during the training phase.

6. Results and Discussion

In this section, we present and discuss the results of our experiments. We start by comparing the results of our approach with those of the multiple baselines defined in Section 5.4. Then, we analyze the performance of our approach in terms of each general-purpose communicative function. After that, we rely on the leave-one-corpus-out cross-validation approach to analyze the generalization ability of the classifiers. Finally, we discuss the importance of the multiple components of the architecture and of transfer learning processes while looking into the results of our ablation studies. In most of the tables referred to in this section, we report the average (mean) and standard deviation of the results among the multiple runs. However, for simplicity, we will focus on the average values in this discussion. Although we started by including the LEGO-ISO dialogs to increase the amount of training data, we noticed that they were not beneficial for the model when paired with the information provided by the transfer learning process from the Switchboard Dialog Act Corpus. Thus, the results reported in this section refer to the experiments that do not include those dialogs in the training phase. We provide further insight into this issue in Section 6.4.

6.1 Overall Comparison

In order to compare our approach with the flat and multiple-classifier baselines, we will look into the results achieved using leave-one-dialog-out cross-validation. We start by looking into the results achieved in the scenario in which all segments in the DialogBank are considered. That is, the task is not only to predict the general-purpose communicative function of a segment, but also to identify which segments have communicative functions in the *Task* dimension. Then, we focus on the segments that have communicative functions in the *Task* dimension, in order to analyze the performance of our hierarchical classification approach in a scenario that is more appropriate for its characteristics.

6.1.1 All Segments

Table 5 shows the results of our experiments in the scenario that considers all the segments in the DialogBank when evaluating using leave-one-dialog-out cross-validation. We can see that the approaches that rely on multiple classifiers to obtain the final predictions lead to worse performance than those that rely on a single classifier, either flat or hierarchical, both in terms of exact match ratio and hierarchical F-measure. This suggests that, although each individual classifier is able to focus on a specific part of the hierarchy, as a whole, they are not able to capture or consider information that is important for the overall problem.

Comparing the three multiple classifier approaches, we can see that using one-vs-all classifiers leads to the worst performance in terms of exact match ratio, 63.87%. However,

Approach	**MR**	**hP**	**hR**	**hF**
Flat	**74.17±.28**	79.47±.34	66.25±.23	72.26±.28
One-vs-All Classifiers	63.87±.20	**82.38±.16**	59.98±.03	69.42±.08
One Classifier per Level	69.77±.23	73.89±.19	66.58±.25	70.04±.22
One Classifier per Sibling Set	69.81±.27	72.89±.47	66.84±.37	69.74±.42
End-to-End Hierarchical	72.15±.21	77.62±.23	70.99±.13	**74.16±.17**
Two-Step Hierarchical	73.30±.04	76.87±.08	**71.32±.08**	73.99±.08

Table 5: Results achieved while predicting general-purpose communicative functions in the DialogBank, using the leave-one-dialog-out cross-validation approach. The two-step approach uses a binary classifier to decide whether a segment has a general-purpose communicative function before applying the hierarchical approach.

in terms of hierarchical F-measure, the three approaches have similar performance. Still, the behavior observed when using one-vs-all classifiers differs from the other two, with a higher bias towards the prediction of shallower or no communicative functions. This is revealed by the 22.40 percentage point gap between the performance in terms of hierarchical precision and recall. This can be explained by the fact that most of the one-vs-all classifiers are trained on a highly unbalanced set of examples and that the imbalance increases with depth. Changing the probability threshold for predicting an example as positive could attenuate the impact of the imbalance. However, studying the impact of that change is outside the scope of this work.

Using one classifier per level of the hierarchy or one classifier per set of siblings leads to similar performance overall. The most noticeable difference is one percentage point in terms of hierarchical precision. As discussed in Section 5.4, the classifiers used in these two approaches are only trained on the segments that have the communicative functions that the classifier focuses on distinguishing. Thus, the training sets are more balanced than those of the one-vs-all classifiers. This is reflected in the narrower gap between the performance in terms of hierarchical precision and recall. However, the gap is still of 7.31 percentage points when using one classifier per level and of 6.05 when using one classifier per set of siblings, which suggests that there is still a bias towards the prediction of shallower communicative functions. However, this is still due to the reduction in number of labeled examples with depth, which makes it hard to train appropriate classifiers.

Comparing the flat approach with the end-to-end hierarchical one, we can see that the former achieves the highest performance in terms of exact match ratio, 74.17%, while the latter is the top performer in terms of hierarchical F-measure, with 74.16%. Looking into the remaining metrics, we can see that the higher F-measure of the hierarchical approach is due to a higher recall (4.74 percentage points). This means that it is able to identify communicative functions that are less prominent and/or deeper in the hierarchy. However, that comes at a cost of 2.05 percentage points in terms of precision, which also explains the lower performance in terms of exact match ratio. Still, the additional misclassifications are typically due to the prediction of longer paths and not completely distinct ones, which suggests that the end-to-end hierarchical approach is actually able to capture information regarding the hierarchical dependencies between communicative functions.

Although the end-to-end hierarchical approach is that with highest recall, that value is still 6.63 percentage points below the precision of the same approach, which still reveals the bias towards the prediction of shallower communicative functions. As previously discussed, this can be explained by the small size of the DialogBank, which does not provide a representative coverage of the communicative functions that are deeper in the hierarchy. Thus, we expect that the bias can disappear or at least be attenuated if the classifier is trained on a sufficiently large amount of annotated dialogs. Still, in an attempt to attenuate the bias without relying on additional annotated data, we explored the use of a two-step approach which uses a binary classifier to identify the segments which have a communicative function in the *Task* dimension before applying the hierarchical approach to those segments.

Still in Table 5, we can see that the two-step approach is actually able reduce the gap between recall and precision. However, the 0.75 percentage point drop in terms of precision is higher than the 0.33 percentage point improvement in terms of recall. Still, it outperforms the end-to-end approach in terms of exact match ratio by 1.15 percentage points. The average performance of the binary classifier in terms of exact match ratio is of 85.52 percentage points. This value is in line with that achieved by the end-to-end approach on the top level. Thus, the differences between the two approaches can be explained by two factors. On the one hand, the hierarchical part of the two-step approach is trained solely on the segments that have communicative functions in the *Task* dimension. Thus, it is less biased towards the prediction of the *None* label in every level, which improves the performance in terms of exact match ratio on those segments. On the other hand, an incorrect decision of the binary classifier cannot be corrected by the predictions on the lower levels using MAP prediction. Thus, these misclassifications at the top level have a more prominent impact on the performance of the approach in terms of hierarchical F-measure.

6.1.2 Task Dimension

Table 6 shows the results of our experiments in the scenario that focuses on the recognition of general-purpose communicative functions in segments that have a communicative function in the *Task* dimension. In this scenario, the segments that do not have a communicative function in that dimension are not used to train the classifiers, which allows them to focus on the hierarchical problem. Looking at the table, we can see that, in comparison to the previous scenario, there is a wider gap between the results in terms of exact match ratio and hierarchical F-measure. In this case, the narrower gap is of 15.77 percentage points, while the wider one when considering all segments was of 5.55 percentage points. This difference can be explained by the predominance of segments without communicative function in the *Task* dimension, which cover more than half of the segments. On the one hand, the classifiers are able to identify most of those segments. Thus, the exact match ratio is higher when they are considered. On the other hand, misclassifying a segment as not having communicative functions in the *Task* dimension means that all the functions in the correct path are missed, which leads to a lower hierarchical F-measure. Additionally, our experiments using the two-step approach have already shown that, since the segments without communicative functions in the *Task* dimension have the *None* label in every level

Approach	**MR**	**hP**	**hR**	**hF**
Flat	68.00±.08	85.67±.10	81.97±.12	83.77±.10
One-vs-All Classifiers	45.65±.08	**88.64±.51**	68.83±.23	77.49±.30
One Classifier per Level	56.61±.91	80.09±.66	74.43±.61	77.16±.63
One Classifier per Sibling Set	55.99±.63	77.66±.55	74.35±.62	75.97±.59
End-to-End Hierarchical	**68.22±.94**	86.34±.88	**83.02±.66**	**84.65±.77**

Table 6: Results achieved while predicting general-purpose communicative functions in the DialogBank segments that have a communicative function in the *Task* dimension, using the leave-one-dialog-out cross-validation approach.

of the hierarchy, the classifiers trained on them become more biased towards the prediction of shallower communicative functions, leading to lower recalls.

Similarly to when the segments that do not have communicative function in the *Task* dimension are considered, the approaches that rely on multiple classifiers to obtain the final predictions lead to worse performance than those that rely on a single classifier, either flat or hierarchical. In fact, the difference in performance is higher in this scenario, with the flat classifier outperforming the best multiple classifier approach by 11.39 percentage points in terms of exact match ratio and 6.61 in terms of hierarchical F-measure. This is due to the fact that the classifiers used to distinguish among the communicative functions that are siblings or at the same level were already trained solely on the segments that had one of those communicative functions. On the other hand, the flat and end-to-end hierarchical classifiers were trained on all segments. Thus, the large amount of segments without communicative functions in the *Task* dimension impaired their performance. By training the classifiers solely on the segments that have a communicative function in the *Task* dimension, the bias towards the prediction of shallower functions is attenuated, leading to improved performance.

Comparing the flat approach with the end-to-end hierarchical one, we can see that, in this scenario, the latter is the top performer in terms of every metric, which confirms the appropriateness of the hierarchical approach. Still, the difference is only higher than one percentage point in terms of hierarchical recall. However, one must consider that while the hierarchical approach covers the whole hierarchy of general-purpose communicative functions, the flat approach is trained specifically for distinguishing among the terminal communicative functions existent in the DialogBank. Thus, in a sense, the flat approach tackles a simpler problem. Considering this, the performance of the hierarchical approach, and especially its ability to identify communicative functions that are deeper in the hierarchy, suggests that it is able to capture and leverage information regarding the hierarchical dependencies as it was designed to.

Comparing the performance of the end-to-end hierarchical approach in terms of precision and recall, we can see that, in this case, the gap is reduced to 3.32 percentage points. This value is half of that observed when all the segments were considered, which confirms that given a more balanced and representative set of examples, the end-to-end hierarchical approach can capture information regarding the whole hierarchy.

6.2 Communicative Function Analysis

In order to analyze the performance of our end-to-end hierarchical approach in further detail, in Table 7, we present precision and recall results per communicative function. The number of segments differs from the reported when describing the DialogBank in Section 5.1 because, in that case, only terminal communicative functions were considered. However, in this case, we consider complete paths, as each segment is used as an example of every communicative function in its path.

Looking at the table, we can confirm that, as discussed in the previous section, the performance on the recognition of each communicative function is higher when the classifiers are trained solely on the segments that have communicative functions in the *Task* dimension. Considering that most dialogs include segments with communicative functions in the different semantic dimensions, the process of identifying which segments have a general-purpose communicative function cannot be discarded. However, including that distinction in the hierarchical classifier clearly harms its performance, due to the imbalance caused by the high number of segments that do not have a general-purpose communicative function. Thus, using a two-step approach as described in Section 6.1.1 may be the best option. In fact, instead of a binary classifier, we can use a multilabel classifier that identifies the semantic dimensions in which each segment has communicative functions. However, further experiments with additional data are required to confirm this hypothesis.

The requirement for additional annotated data is brought to attention once again by the inability to identify 15 communicative functions due to reduced or inexistent coverage. Additionally, in spite of a few exceptions, the metrics, especially recall, tend to decrease as the number of segments with the corresponding communicative function decreases.

Regarding particular communicative functions, we can see that the classifiers fail to identify check questions. These are used to elicit a confirmation of a piece of information that the sender is confident on. In most cases, they are expressed in declarative form, which leads to misclassifications with the more predominant *Inform* communicative function. This issue also impairs the performance in terms of the more generic propositional questions, which are in the same path in the hierarchy. Still, even without considering check questions, there are problems in the recognition of propositional questions. These are mostly due to confusions with requests. This happens because, in many cases, requests are performed in question form, which causes confusion even among human annotators.

Another interesting set of results is that of commissives. When the classifiers were trained on all segments in the DialogBank, the performance was of just 33.33% precision and 3.28% recall. On the other hand, when the classifiers were trained solely on the segments with communicative functions in the *Task* dimension, the performance improved to 95.55% precision and 70.49% recall. This is due to the fact that most of the commissives in the DialogBank are segments with the *Accept Request* communicative function. The majority of these segments are short acceptances (e.g. “Yes.” or “Ok.”), which, without context information, are indistinguishable from the widely common segments with positive communicative functions in feedback dimensions. Thus, they are predicted as having no

		All Segments		**Task Dimension**	
Communicative Function	**Segments**	**Precision**	**Recall**	**Precision**	**Recall**
None	1242	85.25	88.08	-	-
General-Purpose	1118	86.25	83.06	100.00	100.00
Information-Transfer	878	78.66	82.46	90.53	96.13
Information-Providing	696	71.16	80.22	81.96	94.20
Inform	696	71.16	80.22	81.96	94.20
Action-Discussion	240	69.02	40.56	81.68	63.19
Information-Seeking	182	67.91	47.25	69.27	50.37
Question	182	67.91	47.25	69.27	50.37
Directives	179	70.57	52.70	76.53	60.15
Request	166	69.82	56.22	74.17	62.85
Answer	164	70.08	50.00	75.22	74.59
Instruct	157	71.12	56.69	73.41	63.91
Propositional Question	114	58.53	27.19	52.97	33.92
Commissives	61	33.33	3.28	95.55	70.49
Set Question	60	78.72	51.67	84.41	66.11
Address Request	57	33.33	3.51	95.55	75.44
Accept Request	54	33.33	3.70	88.89	74.07
Agreement	53	25.00	1.89	76.78	47.80
Check Question	50	0.00	0.00	0.00	0.00
Confirm	40	51.82	27.50	53.51	44.17
Suggestion	13	0.00	0.00	0.00	0.00
Disconfirm	12	0.00	0.00	0.00	0.00
Choice Question	8	0.00	0.00	0.00	0.00
Disagreement	4	0.00	0.00	0.00	0.00
Offer	3	0.00	0.00	0.00	0.00
Correction	3	0.00	0.00	0.00	0.00
Address Offer	2	0.00	0.00	0.00	0.00
Address Suggestion	1	0.00	0.00	0.00	0.00
Promise	1	0.00	0.00	0.00	0.00
Accept Suggestion	1	0.00	0.00	0.00	0.00
Accept Offer	1	0.00	0.00	0.00	0.00
Reject Offer	1	0.00	0.00	0.00	0.00
Reject Suggestion	0	-	-	-	-
Reject Request	0	-	-	-	-
Test Question	0	-	-	-	-

Table 7: Results of the end-to-end hierarchical approach per communicative function, when evaluating using leave-one-dialog-out cross-validation. The number of segments containing each communicative function in based on complete paths and not only on terminal functions. For readability, we present the average values among the multiple runs, but not the standard deviation.

Scenario	**Approach**	**MR**	**hP**	**hR**	**hF**
All Segments	Flat	-8.00	-9.28	-11.06	-10.47
	One-vs-All	-6.45	-7.24	-6.47	-6.91
	Level	-4.56	-3.70	-4.08	-3.92
	Sibling Set	-4.51	-3.80	-4.73	-4.32
	End-to-End	-6.73	-7.54	-9.11	-8.44
	Two-Step	-7.75	-7.57	-8.76	-8.23
Task Dimension	Flat	-19.85	-12.82	-13.85	-13.37
	One-vs-All	-8.14	-9.51	-4.38	-7.06
	Level	-10.55	-5.24	-5.56	-5.42
	Sibling Set	-10.37	-5.03	-6.03	-5.56
	End-to-End	-20.43	-12.21	-13.50	-12.89

Table 8: Average performance changes when using leave-one-corpus-out cross-validation in comparison to when using leave-one-dialog-out cross-validation.

communicative functions in the *Task* dimension when those segments are used for training. Still, even when those segments are not considered, some of the request acceptances are misclassified as *Confirm*, because the segments with that communicative function are also similar and their distinction is highly dependent on the context.

6.3 Generalization Ability

To assess the generalization ability of the classifiers, we rely on the leave-one-corpus-out cross-validation approach. This approach simulates the process of training on one corpus and testing on another used in cross-corpora generalization assessment. However, it reduces the number of dialogs used for training in each fold. Given the small size of the DialogBank, this is enough for impairing the performance of the classifiers. Thus, although we believe this is the most appropriate way to assess cross-corpora generalization ability with the annotated data that is available, the results can only been seen as indicators.

Table 8 shows the difference in performance in comparison to when using the leave-one-dialog-out cross-validation approach. We can see that, as expected, the performance of every approach decreased in terms of every metric. However, the approaches that rely on multiple classifiers seem to be less affected. This can be justified by two factors. On the one hand, these approaches were already those with worst performance and, especially, those with less ability to recognize the least covered communicative functions. On the other hand, except for the one-vs-all classifiers (which are those with worst performance), the classifiers used in these approaches are trained only on a subset of the segments. Thus, they are less influenced by additional imbalances in the whole training set.

Regarding the flat and end-to-end hierarchical approaches, we can see that the decrease in performance is higher when only the segments that have a communicative function in the *Task* dimension are considered. This means that the decrease is not related to the identification of the segments that have communicative functions in that dimension, but rather

Approach	**MR**	**hP**	**hR**	**hF**
Full	**68.22±.94**	**86.34±.88**	**83.02±.66**	**84.65±.77**
- Cascading Outputs	66.73±.00	86.05±.13	82.30±.25	84.14±.19
- Specialization Layers	65.71±.99	85.18±.51	81.42±.84	83.25±.68
- MAP Prediction	66.82±.19	85.98±.12	82.54±.08	84.23±.10
+ LEGO-ISO	64.25±.22	84.59±.07	82.23±.08	83.40±.02
- Transfer Learning	59.48±.19	83.73±.09	80.35±.10	82.00±.09

Table 9: Results of the ablation studies. The first block shows the performance of the full approach for comparison. The second block shows the results achieved when one component of the architecture is removed. The last block shows the results regarding the inclusion of information from external data.

to the inability to recognize some of the general-purpose communicative functions. Still, when considering all segments, the performance decrease of the flat approach is higher than that of the end-to-end hierarchical approach. This is another indicator of the greater ability of the hierarchical approach to identify communicative functions that are less prominent and/or deeper in the hierarchy, even in the presence of a higher imbalance.

Overall, we believe that the decrease in performance when evaluating using leave-one-corpus-out cross-validation is not due to an innate inability of our approach to generalize, but rather to the less representative set of segments used to train the classifiers in each fold, which makes it impossible to learn how to recognize certain communicative functions. This happens not only because a lower number of segments is used to train each classifier, but also because, as discussed in Section 5.1, the small size of the DialogBank combined with the different characteristics of the corpora it features makes some communicative functions mostly or only occur in one of the corpora. A prominent example of this is the fact that 85% of the segments with a communicative function in the *Action-Discussion* branch of the hierarchy belong to MapTask dialogs. This highly impacts the ability to recognize 17 different communicative functions in this scenario.

6.4 Ablation Studies

In order to assess the importance of the multiple components of our end-to-end hierarchical approach to the automatic recognition of general-purpose communicative functions, we performed a set of ablation studies, in which one of the components was removed and the performance was compared with that of the full approach. These studies can be split into two categories: those regarding the architecture of the approach and those regarding data and transfer learning aspects. The experiments were performed in the scenario that targets the recognition of general-purpose communicative functions in segments that have a communicative function in the *Task* dimension and were evaluated using leave-one-dialog-out cross-validation. Table 9 shows the results of these experiments, with the first block showing the performance of the full approach for comparison.

The second block of Table 9 shows the results achieved when one of the components of the architecture is removed. We can see that the performance is negatively impacted in every case. By removing the connections between the output at each level and those below it, the average performance decreased by 1.49 percentage points in terms of exact match ratio and 0.51 percentage points in terms of hierarchical F-measure. Furthermore, while the performance drops in terms of both precision and recall, the drop is higher in terms of the latter. This suggests that the cascading outputs help attenuating the bias towards the prediction of shallower communicative functions.

By removing the specialization layers, the impact on performance is higher than that of removing the cascading outputs. More specifically, the average performance drops 2.51, 1.16, 1.60, and 1.40 percentage points in terms of exact match ratio and hierarchical precision, recall, and F-measure, respectively. This means that the specialization layers are able to capture the information present in the representation of the segment decorated with context information that is most relevant to distinguish among the communicative functions at the corresponding level of the hierarchy.

Finally, when not using the MAP prediction approach, the performance decreased 1.40 percentage points in terms of exact match ratio and 0.42 in terms of F-measure. In this case, the prediction approach was replaced by the top-down one used in the multiple-classifier approaches. That is, the prediction starts at the root of the hierarchy and, at each level, the selected communicative function is the highest-probability child of the one selected in the previous level. The decrease in performance confirms that the MAP prediction approach can attenuate the impact of misclassifications in the top levels by relying on the distributional outputs obtained using *softmax* to obtain a joint prediction for all levels of the hierarchy. Additionally, the higher decrease in performance in terms of exact match ratio suggests that the MAP prediction approach also helps with the prediction of paths of the correct length.

The last block in Table 9 shows the results regarding the inclusion of external data. We can see that, contrarily to what we observed in our preliminary study using the LEGO-ISO dialogs (Ribeiro et al., 2020), in this case, their inclusion during the training phase impaired the performance by 3.97 percentage points in terms of exact match ratio and 1.25 in terms of hierarchical F-measure. This can be explained by several factors. First, on that study, we did not rely on BERT word embeddings nor on transfer learning processes to obtain segment representations. Thus, the LEGO-ISO dialogs were important for generating more generic segment representations and avoiding overfitting. On the other hand, when those dialogs only contribute for the training of the specialization and output layers as in this study, they have a negative impact because their characteristics differ from those of the dialogs in the DialogBank, especially in terms of the user utterances. Second, considering that the hierarchical approach has one pair of specialization and output layers per level, it has more trainable parameters than the flat one. Consequently, it is more prone to overfitting. That is relevant in this context since the LEGO-ISO dialogs are in larger number, they are highly repetitive, and are mostly covered by a small set of general-purpose communicative functions. Last, we were not focusing on the generalization ability of the classifiers in our preliminary study. Thus, the cross-validation process fine-tuned the classifiers to achieve the highest performance on the test dialog of the corresponding fold. This fine-tuning process made the classifiers less prone to overfit to the specificities of the LEGO-ISO dialogs.

Finally, the results achieved without pre-training the segment representation layers for dialog act recognition on the Switchboard Dialog Act Corpus reveal that using transfer learning improves the performance in terms of every metric, especially exact match ratio, which improves by 8.74 percentage points. This suggests that, given the reduced amount of data, training the layers that generate the segment representations solely on the dialogs annotated according to the ISO 24617-2 standard leads to overfitting. On the other hand, since the Switchboard Dialog Act Corpus is sufficiently large, a model trained on its dialogs generates representations that capture information regarding generic intention that is not specific to a single set of labels. The specificities of different sets are then captured by the specialization and output layers, leading to the generation of models that have higher generalization potential. In this context, we also performed experiments evaluated using leave-one-corpus-out cross-validation. These revealed an even higher drop in performance when not relying on transfer learning, especially in terms of hierarchical F-measure, as the classifiers became overfit to the characteristics of the training corpora.

7. Conclusions

In this article, we have explored the automatic recognition of the general-purpose communicative functions defined by the ISO 24617-2 standard for dialog act annotation. To do so, we proposed modifications to existing approaches to flat dialog act recognition that allow them to deal with the hierarchical classification problem posed by these communicative functions. Experiments on the DialogBank, which is a reference set of dialogs annotated according to the standard, have shown that our end-to-end hierarchical approach outperforms a flat approach similar to those used on most dialog act recognition tasks, as well as hierarchical approaches based on multiple classifiers, both in terms of exact match ratio and hierarchical F-measure.

Addressing the modifications more specifically, instead of a single output layer, our approach includes one output layer per level of the hierarchy. This allows it to focus on distinguishing among communicative functions that are at the same level without having to deal with the ambiguities caused by communicative functions that are ancestors or descendants of each other. Furthermore, it also includes one specialization layer per level, which captures the information provided by the generic segment representation decorated with context information that is most relevant for the corresponding level.

Since the segments in a dialog may have a communicative function that is not a leaf of the hierarchy, we included an additional label in each level of the hierarchy, which refers to the absence of a label at that level and those under it, allowing the prediction of paths with variable length. Furthermore, in order to avoid predicting invalid communicative function paths, our approach relies on a prediction approach based on MAP estimation, which considers the parent-child relations between the communicative functions and improves robustness to misclassifications in individual levels.

Finally, since the DialogBank only features a small set of dialogs, we relied on transfer learning processes to generate more generic segment representations. More specifically, the layers that generate the representations were pre-trained on the Switchboard Dialog Act Corpus, which is the largest corpus annotated for dialog acts. The generic representations are then fine-tuned for the distinction among communicative functions by the specialization layers. This way, the classifier is less prone to overfit to the characteristics of specific DialogBank dialogs, leading to improved performance.

To the best of our knowledge, this was the first study to focus on the automatic recognition of the complete hierarchy of general-purpose communicative functions defined by the ISO 24617-2 standard. Still, it focused on devising an approach that is appropriate to deal with the hierarchical classification problem posed by the communicative functions. Thus, as future work, it would be interesting to compare the different segment and context information representation approaches used in dialog act recognition studies to identify the most appropriate one for this task. Furthermore, in addition to the general-purpose communicative functions, the ISO 24617-2 standard also defines dimension-specific communicative functions and a complete dialog act annotation includes additional information. Thus, the automatic recognition of all the relevant aspects should also be addressed as future work. However, that requires a representative amount of annotated data, which the DialogBank does not possess. Consequently, additional efforts should be made to increase the number of publicly available dialogs fully annotated according to the standard.

Acknowledgements

Eugénio Ribeiro is supported by a PhD scholarship granted by Fundação para a Ciência e a Tecnologia (FCT), with reference SFRH/BD/148142/2019. Additionally, this work was supported by Portuguese national funds through FCT, with reference UIDB/50021/2020.

References

Abadi, M., et al. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/.

Allen, J. F., & Schubert, L. K. (1991). The TRAINS Project. Tech. rep. Technical Report 382 and TRAINS Technical Note 91-1, Computer Science Department, 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.

Anikina, T., & Kruijff-Korbayová, I. (2019). Dialogue Act Classification in Team Communication for Robot Assisted Disaster Response. In *SIGDIAL*, pp. 399–410.

Benedí, J.-M., Lleida, E., Varona, A., Castro, M.-J., Galiano, I., Justo, R., de Letona, I. L., & Miguel, A. (2006). Design and Acquisition of a Telephone Spontaneous Speech Dialogue Corpus in Spanish: DIHANA. In *LREC*, pp. 1636–1639.

Blache, P., Abderrahmane, M., Rauzy, S., Ochs, M., & Oufaida, H. (2020). Two-Level Classification for Dialogue Act Recognition in Task-Oriented Dialogues. In *COLING*, pp. 4915–4925.

Bothe, C., Weber, C., Magg, S., & Wermter, S. (2018). A Context-based Approach for Dialogue Act Recognition using Simple Recurrent Neural Networks. In *LREC*, pp. 1952–1957.

Bunt, H., Alexandersson, J., Choe, J.-W., Fang, A. C., Hasida, K., Petukhova, V., Popescu-Belis, A., & Traum, D. (2012). ISO 24617-2: A Semantically-Based Standard for Dialogue Annotation. In *LREC*, pp. 430–437.

Bunt, H., Petukhova, V., Malchanau, A., Fang, A., & Wijnhoven, K. (2019). The DialogBank: Dialogues with Interoperable Annotations. *Language Resources and Evaluation*, *53*(2), 213–249.

Bunt, H., Petukhova, V., Malchanau, A., Wijnhoven, K., & Fang, A. (2016). The DialogBank. In *LREC*, pp. 3151–3158.

Bunt, H., Petukhova, V., Traum, D., & Alexandersson, J. (2017). Dialogue Act Annotation with the ISO 24617-2 Standard. In *Multimodal Interaction with W3C Standards*, pp. 109–135. Springer.

Carroll, J. M., & Tanenhaus, M. K. (1978). Functional Clauses and Sentence Segmentation. *Journal of Speech, Language, and Hearing Research*, *21*(4), 793–808.

Cerisara, C., Jafaritazehjani, S., Oluokun, A., & Le, H. T. (2018). Multi-task Dialog Act and Sentiment Recognition on Mastodon. In *COLING*, pp. 745–754.

Chen, Z., Yang, R., Zhao, Z., Cai, D., & He, X. (2018). Dialogue Act Recognition via CRF-Attentive Structured Network. In *SIGIR*, pp. 225–234.

Chollet, F., et al. (2015). Keras: The Python Deep Learning Library. https://keras.io/.

Devlin, J., Chang, M.-W., Kenton, L., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, Vol. 1, pp. 4171–4186.

Fang, A., Cao, J., Bunt, H., & Liu, X. (2012). The Annotation of the Switchboard Corpus with the New ISO Standard for Dialogue Act Analysis. In *Workshop on Interoperable Semantic Annotation*, pp. 13–18.

Gambäck, B., Olsson, F., & Täckström, O. (2011). Active Learning for Dialogue Act Classification. In *INTERSPEECH*, pp. 1329–1332.

Godfrey, J. J., Holliman, E. C., & McDaniel, J. (1992). SWITCHBOARD: Telephone Speech Corpus for Research and Development. In *ICASSP*, Vol. 1, pp. 517–520.

Ji, Y., Haffari, G., & Eisenstein, J. (2016). A Latent Variable Recurrent Neural Network for Discourse Relation Language Models. In *NAACL-HLT*, Vol. 1, pp. 332–342.

Jurafsky, D., Shriberg, E., & Biasca, D. (1997). Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coder Manual. Tech. rep. Draft 13, Institute of Cognitive Science, University of Colorado.

Kalchbrenner, N., & Blunsom, P. (2013). Recurrent Convolutional Neural Networks for Discourse Compositionality. In *Workshop on Continuous Vector Space Models and their Compositionality*, pp. 119–126.

Khanpour, H., Guntakandla, N., & Nielsen, R. (2016). Dialogue Act Classification in Domain-Independent Conversations Using a Deep Recurrent Neural Network. In *COLING*, pp. 2012–2021.

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *ICLR*.

Kiritchenko, S., Matwin, S., & Famili, F. (2005). Functional Annotation of Genes using Hierarchical Text Categorization. In *BioLINK SIG*.

Král, P., & Cerisara, C. (2010). Dialogue Act Recognition Approaches. *Computing and Informatics*, *29*(2), 227–250.

Kruijff-Korbayová, I., Colas, F., Gianni, M., Pirri, F., de Greeff, J., Hindriks, K., Neerincx, M., Ögren, P., Svoboda, T., & Worst, R. (2015). TRADR Project: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response. *KI - Künstliche Intelligenz*, *29*(2), 193–201.

Kumar, H., Agarwal, A., Dasgupta, R., & Joshi, S. (2018). Dialogue Act Sequence Labeling Using Hierarchical Encoder With CRF. In *AAAI*, pp. 3440–3447.

Lee, J. Y., & Dernoncourt, F. (2016). Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. In *NAACL-HLT*, Vol. 2, pp. 515–520.

Li, R., Lin, C., Collinson, M., Li, X., & Chen, G. (2019). A Dual-Attention Hierarchical Recurrent Neural Network for Dialogue Act Classification. In *CoNLL*, pp. 383–392.

Liu, Y., Han, K., Tan, Z., & Lei, Y. (2017). Using Context Information for Dialog Act Classification in DNN Framework. In *EMNLP*, pp. 2160–2168.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, pp. 3111–3119.

Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*, pp. 807–814.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *EMNLP*, pp. 1532–1543.

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., & Liersch, S. (2014). The DBOX Corpus Collection of Spoken Human-Human and Human-Machine Dialogues. In *LREC*, pp. 252–258.

Qin, L., Che, W., Li, Y., Ni, M., & Liu, T. (2020). DCR-Net: A Deep Co-Interactive Relation Network for Joint Dialog Act Recognition and Sentiment Classification. In *AAAI*, pp. 8665–8672.

Raheja, V., & Tetreault, J. (2019). Dialogue Act Classification with Context-Aware Self-Attention. In *NAACL*, pp. 3727–3733.

Raux, A., Bohus, D., Langner, B., Black, A. W., & Eskenazi, M. (2006). Doing Research on a Deployed Spoken Dialogue System: One Year of Let's Go! Experience. In *INTERSPEECH*, pp. 65–68.

Ravi, S., & Kozareva, Z. (2018). Self-Governing Neural Networks for On-Device Short Text Classification. In *EMNLP*, pp. 804–810.

Ren, F., & Xue, S. (2020). Intention Detection Based on Siamese Neural Network With Triplet Loss. *IEEE Access*, *8*, 82242–82254.

Ribeiro, E., Ribeiro, R., & Martins de Matos, D. (2015). The Influence of Context on Dialogue Act Recognition. *Computing Research Repository*, *arXiv:1506.00839*.

Ribeiro, E., Ribeiro, R., & Martins de Matos, D. (2018a). A Study on Dialog Act Recognition using Character-Level Tokenization. In *AIMSA*, pp. 93–103.

Ribeiro, E., Ribeiro, R., & Martins de Matos, D. (2018b). End-to-End Multi-Level Dialog Act Recognition. In *IberSPEECH*, pp. 301–305.

Ribeiro, E., Ribeiro, R., & 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., & 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.

Ribeiro, E., Ribeiro, R., & Martins de Matos, D. (2020). Mapping the Dialog Act Annotations of the LEGO Corpus into ISO 24617-2 Communicative Functions. In *LREC*, pp. 531–539.

Schmitt, A., Ultes, S., & Minker, W. (2012). A Parameterized and Annotated Spoken Dialog Corpus of the CMU Let's Go Bus Information System. In *LREC*, pp. 3369–3373.

Searle, J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

Silla Jr., C. N., & Freitas, A. A. (2011). A Survey of Hierarchical Classification Across Different Application Domains. *Data Mining and Knowledge Discovery*, *22*(1–2), 31–72.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Tran, Q. H., Zukerman, I., & Haffari, G. (2017a). A Generative Attentional Neural Network Model for Dialogue Act Classification. In *ACL*, Vol. 2, pp. 524–529.

Tran, Q. H., Zukerman, I., & Haffari, G. (2017b). A Hierarchical Neural Model for Learning Sequences of Dialogue Acts. In *EACL*, Vol. 1, pp. 428–437.

Tran, Q. H., Zukerman, I., & Haffari, G. (2017c). Preserving Distributional Information in Dialogue Act Classification. In *EMNLP*, pp. 2151–2156.

Wan, Y., Yan, W., Gao, J., Zhao, Z., Wu, J., & Yu, P. S. (2018). Improved Dynamic Memory Network for Dialogue Act Classification with Adversarial Training. In *IEEE International Conference on Big Data*, pp. 841–850.

Wang, D., Li, Z., Sheng, D., Zheng, H.-T., & Shen, Y. (2021). Balance the Labels: Hierarchical Label Structured Network for Dialogue Act Recognition. In *IJCNN*, pp. 1–8.

Wang, D., Li, Z., Zheng, H., & Shen, Y. (2020). Integrating User History into Heterogeneous Graph for Dialogue Act Recognition. In *COLING*, pp. 4211–4221.

Żelasko, P., Pappagari, R., & Dehak, N. (2021). What Helps Transformers Recognize Conversational Structure? Importance of Context, Punctuation, and Labels in Dialog Act Recognition. *Transactions of the Association for Computational Linguistics*, *9*, 1179–1195.

Zhao, T., & Kawahara, T. (2019). Effective Incorporation of Speaker Information in Utterance Encoding in Dialog. *Computing Research Repository*, *arXiv:1907.05599*.