Improving Dialogue Act Classification by Considering Context: An investigation of dialogues acts using Bert Encoding (Devlin et al., 2019)

Mathieu Garrouty*
ENSAE
mathieu.garrouty@ensae.fr

Louise Carrotte*
ENSAE
louise.carrotte@ensae.fr

Abstract

The recognition of Dialog Acts has become a crucial area of research in recent years, particularly with the growth of chat assistants like ChatGPT. The key to generating a conversation that is as natural as possible is to understand the intent behind each message. In this study, we focus on classifying a dataset of multi-turn dialogs between two individuals, with each message labelled according to its Dialog Act. Our objective is to predict the Dialog Act classification. We compare a basic sequence-level model, where the neural network learns from all labelled sequences, with dialog-level models that take into account the context of a dialog. We employ Recurrent Neural Networks, both with and without self-attention mechanisms, and find that our prediction accuracy increases significantly within a comparable training time, highlighting the importance of context for better representation of dialogs in natural language processing. Code is available on Github [1].

1 Introduction

Conversational agents, such as chatbots and virtual assistants, have become increasingly prevalent in our daily lives. These agents rely on natural language processing (NLP) techniques to understand and respond to user requests and inquiries. One essential aspect of NLP is intent classification, which involves identifying the underlying purpose or goal behind a user's input. Accurately identifying the user's intent is crucial for designing effective conversational agents that can provide prompt and relevant responses (Colombo* et al., 2019; Jalalzai* et al., 2020), thereby improving the user's experience.

However, conversations are not solely about information exchange, as emotions also play a critical role in human communication. Recognizing and responding appropriately to the user's emotional state is essential for creating empathetic and engaging conversational experiences. Therefore, emotion recognition is another critical component that can help design better conversational agents. By detecting the user's emotional state, conversational agents can adjust their responses and tone to provide appropriate and supportive interactions.

In this context, intent classification and emotion recognition are closely intertwined, as the user's intent and emotional state can affect the content and tone of their input. Thus, improving the accuracy and effectiveness of these two NLP tasks can significantly enhance the conversational experience and overall performance of conversational agents.

In this work, we choose to focus on dialog act due to the huge availability of open source datasets (Godfrey et al., 1992; Li et al., 2017; Leech and Weisser, 2003; Busso et al., 2008; Passonneau and Sachar., 2014; Thompson et al., 1993; Shriberg et al., 2004).

1.1 Problem Formulation

Our aim is to build a Sequence Classifier for dialogs acts. In linguistics and in particular in natural language understanding, a **dialog act** is an utterance, in the context of a conversational dialog, that serves a function in the dialog. Types of dialog acts include a question, a statement, or a request for action (McTear et al., 2016). Each dialog involves two speakers, speaking turn by turn.

Dataset

To do so, we will use the **DailyDialog** Corpus (Li, 2017), which is a human-written multiturn dialogue dataset, reflecting our daily communication way and covering various topics about our daily life. The dataset is already splitted into train,

[1] https://github.com/mgarrouty/NLP_intent_classification

validation and test set.

Total Dialogues	13,118
Average Speaker Turns Per Dialogue	7.9
Average Tokens per Dialogue	114.7
Average Tokens Per Utterance	14.6

Table 1: Statistics of DailyDialog Dataset

Dialogs in our database are multi-turned, and always involve only two speakers. Each speaker can pronounce several sentences in a single sequence. We can represent a Dialog D as follow

$$D = (S_1^a, S_1^b, S_2^a, S_2^b, S_3^a..)$$

, where S_i^j represents a sequence. A sequence is represented as follow

$$S_i^j = (s_1^{i,j}, s_2^{i,j}..)$$

, where $s_k^{i,j}$ represents a sentence. Eventually, a sentence is made of utterances :

$$s_k^{i,j} = (u_1^{k,i,j}, u_2^{k,i,j}, u_3^{k,i,j}..)$$

Each sequence is manually labelled with its nature (inform (1), question (2), directive (3), commissive (4)), 0 being used as a dummy variable.

Dialog Act	Count	Percentage
Informative	39873	45.7
Question	24974	28.6
Directive	14242	16.3
Commissive	8081	9.4

Table 2: Dialog Act Repartition in Dailydialog

Sequence	Act
”Say, Jim, how about going for a few beers after dinner?”	3
”You know that is tempting but is really not good for our fitness.”	4
”What do you mean? It will help us to relax.”	2

Table 3: Extract of a dialog

2 Experiments Protocol

To build our Dialog Act Classifier, we use Neural Network Architecture, widely used in Natural Language Recognition. Especially, we are working with *PyTorch* library on Python. In this paper, we use the notation (X_i, Y_i) for data, where X_i can either represent a dialog (list of messages) or a message, and Y_i either a list of labels (Dialog Acts) or a label.

Our models are trained with these data, and are ranked according to their accuracy on the test set :

$$Acc = \frac{1}{|TestSet|} \sum_{X_i in TestSet} \mathbb{1}_{\hat{Y}_i = Y_i}$$

Where $\hat{Y}_i$ is the label predicted by our model. In order to compare message and dialog level accuracy, we always compute the accuracy at a message level. We are using accuracy to have results that are not relying on the loss used for training. As a consequence, we use both NLLLoss and CrossEntropyLoss in our models, depending on which provides the best accuracy.

Data Encoding

The first step of each NLP task consists in transforming language into vectors. They are several ways to do so, and we consider the **BERT** model (Horev, 2018). It is an open-source NLP pre-trained model powered by Google, transforming each word into a vector. The original model turns each token into a vector in $\mathbb{R}^{768}$. For computational efficience, we consider the *Bert tiny* library, allowing a dimension-reduced representation in $\mathbb{R}^{128}$.

Final Layer of our Network

As we are working on classification task, all our models have the same final layer, consisting of a Softmax Layer, to compute a probability score for each label. We then take the highest probability label.

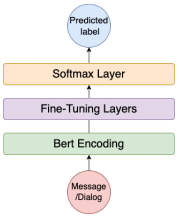

Figure 1: Generic Network Architecture

Loss Function

We choose the CrossEntropy Loss function, givent that it gives the best accuracy result. This function is widely used in classification problem. For an input X, with a softmax vector $\hat{Y} = (p_1, p_2, p_3, p_4)$, where (p_i) represents the computed probability of i being the right label for X, the value of the loss function is :

$$L(X, \hat{Y}) = \sum_{i=1}^{4} -y_i log(p_i)$$

where $Y = (y_i)$ is the vector of the real label of X ($y_i = 1$ only if label of X is i).

Dropout rate

In order to prevent our model from overfitting, we add a dropout layer (consisting in ignoring a share p of our data). This share, called the *dropout rate*, is chosen by trying several numbers and selecting the one giving the best accuracy **on the validation dataset**. Indeed, we can have a signficant difference between the accuracy on the training and the validation set.

2.1 Message-level classification

For our baseline, we flatten our dataset of dialog into a dataset of messages. We encode them using *Bert tiny*, and we add two linear layers before classification. Formally, for a message $S = (s_1, s_2, s_3, ...)$, where $s_i = (u_1^i, u_2^i, ...)$ denotes a sentence, and u denotes a token, our bert encoder returns a vector from $\mathbb{R}^{128}$ for each token. To obtain a vector for our sentence, we choose to average the vectors associated to each word of the sentence.

2.2 Dialog level classification

We now consider a dialog-level, inducing a hierarchical architecture (Colombo et al., 2020) . We now feed our models with matrices of $\mathbb{R}^{12*128}$, where each row represents a message. Each message is represented by the average of the embeddings of its words, and 12 is the max length of dialog considered. We add rows of zeros for shorter dialogs.

2.2.1 Linear Layer

After our encoding, we add two linear layers before classification.

2.2.2 BiLTSM Layer

After our encoding, we add a Bidirectionnal Recursive Layer, to let the model learn the links between the rows of the input matrice.

3 Results

We implement our different methods. For each one, we tried several hyperparameters (learning rate, number of epochs), in order to maximize our accuracy, within computation time constraint.

3.1 Baseline

For our baseline, we find the best accuracy for a learning rate of 10^{-4}, and the loss (best accuracy with CrossEntropy) is not significantly decreasing after 7-8 epochs (we ran over 10), as shown on the graph :

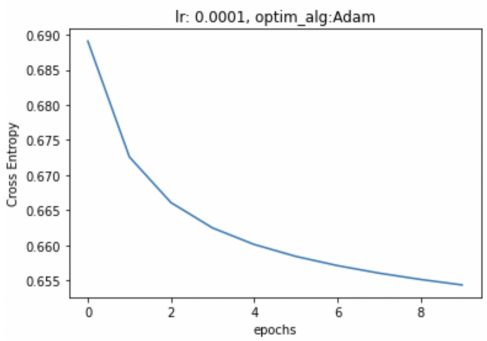

Figure 2: Loss on validation test for Baseline Model

	Precision	Recall	f1-score	Support
Inform	0.32	0.88	0.47	2948
Question	0.84	0.88	0.86	2175
Directive	0.59	0.48	0.53	1705
Commissive	0.49	0.08	0.14	867
Micro-averaged	0.45	0.70	0.55	7695
Macro-averaged	0.56	0.58	0.50	7695
Weighted-averaged	0.54	0.70	0.55	7695

Table 4: Classification report for Baseline Model

We end with an average accuracy of 0.452 with this model.

3.2 Linear Layer

We are now working at a dialog-level here, slightly modifying our data preprocessing. We have the same hyperparameters as for our baseline

The average accuracy on our validation test is 0.807.

3.3 BiLTSM Model

For this model, we replace the former linear layers by a Bidirectionnal LTSM. These layers allow

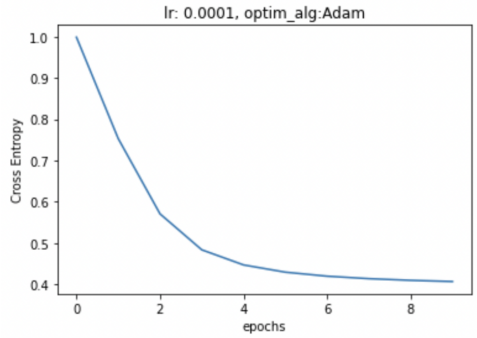

Figure 3: Loss on validation test for Linear Model

	Precision	Recall	f1-score	Support
Inform	0.66	0.88	0.76	2948
Question	0.82	0.91	0.86	2175
Directive	0.58	0.43	0.50	1705
Commissive	0.46	0.06	0.10	867
Micro-averaged	0.70	0.70	0.70	7695
Macro-averaged	0.63	0.57	0.55	7695
Weighted-averaged	0.67	0.70	0.66	7695

Table 5: Classification report for linear Model

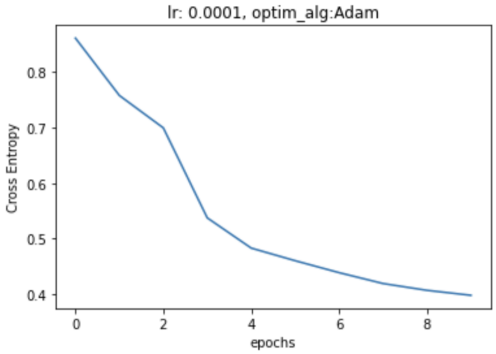

Figure 4: Loss on validation test for BiLTSM Model

	Precision	Recall	f1-score	Support
Inform	0.67	0.89	0.76	2948
Question	0.81	0.92	0.86	2175
Directive	0.57	0.41	0.48	1705
Commissive	0.51	0.03	0.06	867
Micro-averaged	0.70	0.70	0.70	7695
Macro-averaged	0.64	0.56	0.54	7695
Weighted-averaged	0.67	0.70	0.65	7695

Table 6: Classification report for BiLTSM Model

to backpropagate data in the Network, but they require more computational power. The average accuracy on our validation test is 0.804.

4 Discussion/Conclusion

4.1 Overall results on accuracy

Despite the quality of Bert encoding, the accuracy of our baseline model was less than 0.5. We obtained a significant increase when we took our data to a dialog-level (30 percents). What can we learn from this result ?

Our results demonstrate that Dialog Act Recognition is not solely determined by individual messages, as context plays a critical role in enhancing classification accuracy. This additional contextual information can be learned by a neural network.

However, we did not observe a significant improvement in accuracy when using a BiLTSM model at the dialog level, as messages are already vectorized at the message level. Linking messages and utterances would require greater computational resources.

Our classification report highlights that the majority of the difference in accuracy is attributed to precision in classifying "inform" messages, which is the most frequently occurring label.

In addition, our model fine-tuning process revealed the importance of incorporating a dropout layer **dropout layer**, as it resulted in a 5% increase in accuracy. This suggests that our models may have too many parameters, particularly in the case of the linear models, where two linear layers were added for fine-tuning Bert.

4.2 Limitation of current loss

One possible approach to address the issue of unbalanced data in our models would be to apply weighting schemes to adjust the importance of different classes during the training process. This could potentially improve the accuracy and fairness of our models, especially when dealing with imbalanced datasets.

In addition, an interesting research direction would be to explore the joint classification of both dialog act and emotion recognition in the same model. This would enable the model to capture the cross-information and potential relationships between these two tasks, which could lead to more accurate and nuanced predictions. However, implementing such a model would require careful consideration of the appropriate architecture and training approach to optimize performance.

4.3 Extension

To expand upon our findings, further research could involve testing our models on additional datasets to determine their generalizability. Additionally, exploring the integration of self-attention mechanisms in our last model (Bert + BiLTSM) could be a promising avenue for improving the

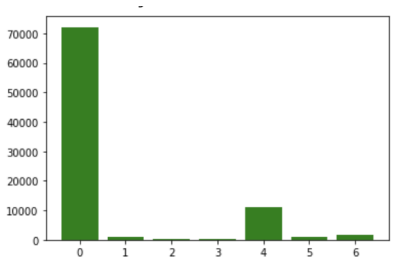

Figure 5: Distribution of emotion label in dailydialog

classification accuracy, especially for capturing the relationships between different words in a sentence. However, this approach would require greater computational resources to train and test effectively.

Another potential research direction would be to adapt our models for code-switched dialogues (Chapuis et al., 2021), where multiple languages are used within a single conversation. This would require an additional layer of classification to identify the language used in each utterance accurately.

Furthermore, pre-training our encoders on large datasets could also improve their performance on specific tasks. Additionally, our models could be adapted for other classification tasks beyond intent classification, such as classifying the emotional state of the user (Dinkar* et al., 2020; Witon* et al., 2018) or finding its opinion (Garcia* et al., 2019). However, we note that this dataset's data distribution is significantly unbalanced, and addressing fairness concerns (Colombo, 2021; Colombo et al., 2021, 2022; Pichler et al., 2022) in classification remains an important and pressing research direction for improving the accuracy and fairness of NLP models.

References

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ICASSP'92, page 517–520, USA. IEEE Computer Society.

Henry Thompson, Anne Anderson, Ellen Bard, Gwyneth Doherty-Sneddon, Alison Newlands, and Cathy Sotillo. 1993. The hcrc map task corpus: natural dialogue for speech recognition.

Geoffrey Leech and Martin Weisser. 2003. Generic speech act annotation for task-oriented dialogues.

Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The ICSI meeting recorder dialog act (MRDA) corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004*, pages 97–100, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower Provost, Samuel Kim, Jeannette Chang, Sungbok Lee, and Shrikanth Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42:335–359.

R. Passonneau and E. Sachar. 2014. Loqui human-human dialogue corpus (transcriptions and annotations).

Michael McTear, Zoraida Callejas, and David Griol. 2016. The conversational interface: Talking to smart devices.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset.

Yanran Li. 2017. A manually labelled multi-turn dialogue dataset.

Rani Horev. 2018. Bert model.

Wojciech Witon*, Pierre Colombo*, Ashutosh Modi, and Mubbasir Kapadia. 2018. Disney at iest 2018: Predicting emotions using an ensemble. In *Wassa @EMNP2018*.

Pierre Colombo*, Wojciech Witon*, Ashutosh Modi, James Kennedy, and Mubbasir Kapadia. 2019. Affect-driven dialog generation. *NAACL 2019*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Alexandre Garcia*, Pierre Colombo*, Slim Essid, Florence d'Alché Buc, and Chloé Clavel. 2019. From the token to the review: A hierarchical multimodal approach to opinion mining. *EMNLP 2019*.

Pierre Colombo, Emile Chapuis, Matteo Manica, Emmanuel Vignon, Giovanna Varni, and Chloé Clavel. 2020. Guiding attention in sequence-to-sequence models for dialogue act prediction.

Hamid Jalalzai*, Pierre Colombo*, Chloé Clavel, Éric Gaussier, Giovanna Varni, Emmanuel Vignon, and Anne Sabourin. 2020. Heavy-tailed representations, text polarity classification & data augmentation. *NeurIPS 2020*.

Tanvi Dinkar*, Pierre Colombo*, Matthieu Labeau, and Chloé Clavel. 2020. The importance of fillers for text representations of speech transcripts. *EMNLP 2020*.

Emile Chapuis, Pierre Colombo, Matthieu Labeau, and Chloé Clavel. 2021. Code-switched inspired losses for generic spoken dialog representations.

Pierre Colombo. 2021. *Learning to represent and generate text using information measures*. Ph.D. thesis, (PhD thesis) Institut polytechnique de Paris.

Pierre Colombo, Chloe Clavel, and Pablo Piantanida. 2021. A novel estimator of mutual information for learning to disentangle textual representations. *() ACL 2021*.

Georg Pichler, Pierre Jean A Colombo, Malik Boudiaf, Günther Koliander, and Pablo Piantanida. 2022. A differential entropy estimator for training neural networks. In *() ICML 2022*.

Pierre Colombo, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida. 2022. Learning disentangled textual representations via statistical measures of similarity. *() ACL 2022*.