The Importance of Context in Intent Classification: A Comparative Study of Encoder-Decoder Architectures

Clara Le Gallic-Ach*
ENSAE Paris
clara.legallic.ach@ensae.fr

Coni Soret*
ENSAE Paris
coni.soret@ensae.fr

Abstract

This study presents a benchmark of various encoder and decoder architectures for intent classification in dialogue systems, using the DailyDialog corpus. The role of context in classification accuracy is explored, with a particular focus on the importance of capturing dynamic structures of context in real-world applications. Our results demonstrate that including context significantly improves classification performance, and that the choice of decoder architecture is important both for their architecture and the level of context they use. We also demonstrated that analyzing accuracy according to the context - past, none, and full - provides valuable insights into the impact of context on real-world applications.[1]

1 Introduction

The identification of the intended purpose or goal behind a user's input, known as intent classification, is a crucial task in natural language processing (NLP). This is particularly significant in conversational interfaces such as virtual assistants [Garcia* et al., 2019], chatbots [Colombo* et al., 2019, Jalalzai* et al., 2020], and voice assistants [Dinkar* et al., 2020], where correctly identifying user intent can lead to more efficient interactions [Colombo, 2021]. The objective is to precisely determine the intent label that best describes a user's statement or message during a conversation. In the context of conversational interfaces, an utterance refers to the input provided by a user, and dialog act is a concept that is employed to describe the purpose behind an utterance. It refers to the predefined intent labels that are assigned to an utterance based on its intended purpose, such as making a request, providing information, expressing an opinion, or asking a question. Therefore, intent classification involves recognizing the communicative function of an utterance [Colombo et al., 2021a].

The primary focus of this paper is on intent classification, a crucial task in natural language processing (NLP). While there are numerous datasets available for intent classification [Shriberg et al., 2004, Poria et al., 2018, Thompson et al., 1993], our specific attention is on the DailyDialog dataset [Li et al., 2017]. Our objective is to leverage the unique features of DailyDialog, such as its diverse range of topics, dialogues, and sentiments, to develop a robust and reliable intent classification system that can improve the performance of conversational interfaces. Dialy Dialog is a publicly available dataset of human-human conversations in English that cover a wide range of topics. Our goal is to study the role of contextual dependencies in intent classification. We control the amount of context captured by each model in order to compare them. To do so, we use different encoders and classifiers that use no context, past context, or full dialog context.

In this study, we aim to shed light on the role of context in intent classification.

2 Dataset & Related Work

2.1 Available Datasets

Despite the availability of numerous datasets for intent classification, DailyDialog stands out due to its substantial size. It comprises of more than 13,000 dialogues, with each dialogue involving multiple exchanges between two individuals. The dataset has been meticulously labeled with four distinct intent labels: inform, question, directive, and commissive. The inform label is employed when a user provides information, while the question label is utilized for utterances that request information. The directive label is assigned to utterances that give orders or requests, and the com-

[1] https://github.com/coni26/Intents_classification

missive label is utilized for utterances that commit the speaker to future actions. Furthermore, this dataset is one of the largest, and it encompasses dialogues pertaining to everyday situations and events, thereby making it highly relevant and practical for real-world applications.

2.2 On the role of hierarchy for DA classification

The hierarchical structure of dialogues plays a pivotal role in capturing the contextual dependencies that exist between user utterances. As requests, responses, and other forms of speech are all interdependent in a dialogue, it is imperative to consider the structure of the conversation when modeling these dependencies. Prior research has concentrated on enhancing the treatment of contextual interdependencies in both encoder and decoder models. This has been achieved by using hierarchical or recurrent models that take into account the hierarchical nature of dialogues and the sequential dependencies of the various components within them. Indeed, one of the key challenges in intent classification is to effectively capture the full context and interdependencies present in a dialog. In their paper, [Chapuis et al., 2020] presented a hierarchical encoder that they compared with BERT on different dialogue classification tasks. Since the encoder was already capturing the context, the performance was very close for an MLP, a CRF or a GRU as decoder. In another study, [Colombo et al., 2020] sought to improve classification accuracy by incorporating more context into their model using the *seq2seq* approach with different attention mechanisms. This study improved classification accuracy compared to previous models. Then, these articles have aroused our interest in comparing the effectiveness of different intent classification models with varying levels of context, including no context, past context, and full context. Thus, we hope to provide some keys to the understanding of intent classification models for real-world applications. While capturing the full context and interdependencies of a dialogue is an important challenge in intent classification, it is also important to ensure that models are designed to capture what is realistic in real-world applications. For instance, in the case of real-time chatbots, only the direct past context may be relevant for predicting the intent of the user's current message. As such, our study takes into consideration the practical constraints of real-time dialogue, exploring the effectiveness of different classification models to identify the most accurate approach for real-world applications.

3 Method

Formally, we have a dataset of D conversations, $\mathcal{D} = (C_1, ..., C_D)$. Each conversation is composed of a variable number of utterances, $C_i = (u_1, ..., u_{|C_i|})$, which are themselves sequences of words, $u_{i,j} = (w_1, ..., w_{|u_{i,j}|})$. In parallel, for each conversation C_i, we have a Y_i label consisting of a sequence of labels of the same size as the conversation, $Y_i = (y_1, ..., y_{|C_i|})$. Our goal is to predict these labels using conversation.

3.1 Encoder

To model the utterances, we used different sentence transformers [Reimers and Gurevych, 2019], where in each case a mean-pooling of the word embeddings is performed to get the embedding of the utterance :

- *all-MiniLM-L6-v2*: MiniLM [Wang et al., 2020] (30M parameters) is a compact distilled model with RoBERTa-Large as teacher, in its version with 6 layers. Embedding size: 384.
- *all-mpnet-base-v2*: MPNet [Song et al., 2020] (110M parameters) has the same architecture as Bert and is based on a training method combining MLM (Bert) and PLM (XLNet). Embedding size: 768.
- *gtr-t5-large*: T5-Base [Raffel et al., 2019] (220M parameters) is a text-to-text transformer pre-trained on Colossal Clean Crawled Corpus. Embedding size: 768.
- *all-roberta-large-v1*: RoBERTa large [Liu et al., 2019] (354M parameters). Embedding size: 1024.

The cited models with the prefix *all* have been fine-tuned on a 1B sentence pairs dataset with a contrastive learning objective: given a sentence from the pair, the model should predict which out of a set of randomly sampled sentences, was actually paired with it. While *gtr-t5-large* has been fine-tuned for semantic search on MS-MARCO [Nguyen et al., 2016].

In our work, we use the encoders utterance by utterance, meaning there is no context of other utterances when embedding an utterance. It may not be the most powerful method, but it allows us to work with decoders that do not take context into account. It is one of the keys of our work. It allows us to have more knowledge of our models, both on the baseline efficiency of decoders and the level of context we choose for decoders. Indeed, we can play with the context in decoders without being biased by the encoders.

3.2 Decoder

The decoders will be separated into three categories depending on the context they take to make their prediction.

- No context: Only the embedding of the utterance will be used to make the prediction, the decoder uses nothing of the rest of the dialog. For this, we will use a simple MLP.
- Past context: Only information from previous utterances and the current utterance can be used for prediction. This case is the one that is closest to real case applications. For this, we will use different recurrent networks (RNN, GRU, LSTM) in their unidirectional version.
- Full context: All information from the dialog (both past and future) can be used for each label prediction. Recurrent models (RNN, GRU, LSTM) will be used in their bidirectional version for this.

The best decoder architectures are determined on the validation dataset.

3.3 Training details

We used the given train/validation/test split, with 11k/1k/1k dialogs in each dataset. We used Adam optimizer with a learning rate of 0.001, which is updated using a scheduler with a patience of 10 epochs and an exponential decrease with $\gamma = 0.9$. To have batches, given that the sequences are not all of the same lengths, we can do zero padding with a fixed maximum size. However, we choose to order the sequences in order to have batches of the same length. As a result, some batches were smaller (ie., containing fewer sequences), so we weighted the loss accordingly.

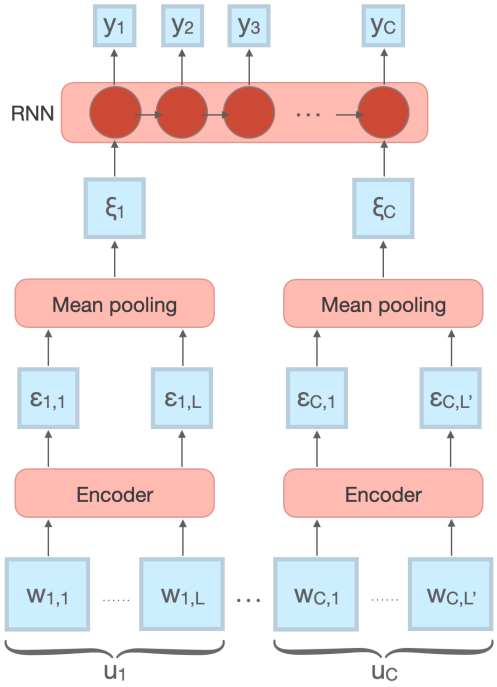

Figure 1: Diagram of our method with a unidirectional RNN as decoder

4 Results

The results are displayed in Table 1, with the best encoder and the best model for each type of context (details in Table 3).

Context	Encoder	Model	Accuracy
None	RoBERTa	MLP	0.779
Past	gtr-t5	LSTM	0.810
Full	RoBERTa	biLSTM	0.819

Table 1: Best accuracy achieved on each type of context

At the level of encoders, we observe that all-MiniLM-L6-v2 is below the others, it is also the one with the least parameters and the smallest embedding size. Besides, gtr-t5-large and all-roberta-large-v1, which are the biggest encoders in terms of number of parameters, have quite close results which are the best in our benchmark.
As expected, performance increases exactly with the amount of context. Even though we get good performance with only the past context, the difference is greater when we look at no context.
Finally, we notice that LSTM is better than GRU which is better than RNN, both in uni and bidirectional.

With figure 2, one could argue that the best performance comes only from the increase in the

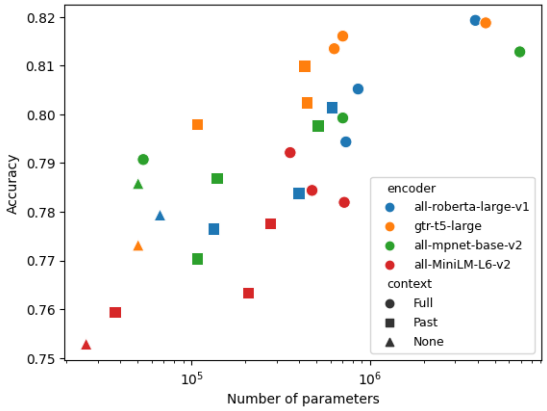

Figure 2: Accuracy depending on the number of parameters, according to the type of context and the encoder

number of parameters and not from the context or the choice of the model. We therefore built models with a constant number of parameters (about 500k) to compare their performance.

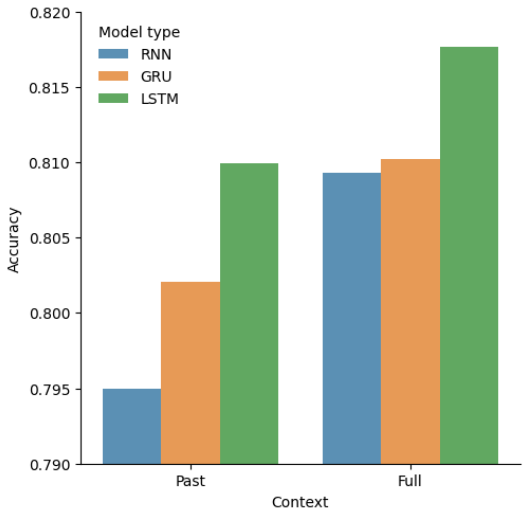

Figure 3: Accuracy depending on the decoder, for a fixed maximum number of parameters, about 500k (decoder: gtr-t5-large)

We notice on figure 3, that for a fixed number of parameters (other numbers of parameters in Appendix), we have the same hierarchy between model and context as presented before. We also notice that LSTM layers outperform RNN and GRU. Even in the unidirectional version, LSTM has very close results to biRNN and even biGRU. We can therefore conclude that performance is not only due to the number of parameters: more efficient models with more context are able to capture more information.

Finally, to observe the relevance of past context, we can also look at the gain in accuracy that we have between the MLP and the recurrent models on the first and last utterance, presented in Table 2.

Model	First utterance	Last utterance
MLP	0.861	0.685
RNN	-0.018	+0.048
GRU	-0.012	+0.058
LSTM	-0.002	+0.070

Table 2: Gain in accuracy on the first and last utterance compared to the MLP baseline, averaged over the 4 encoders.

The accuracy is not comparable between the different utterance positions, because some are simpler than others. For example, on the first utterance, we never have a *commissive* label, so it is obvious to have a higher accuracy. By comparing the performances on the first and the last utterance, we understand that the recurrent models do not have better performances on the first utterance, they are even a little weaker, which can be due to the initialization of the hidden state that skew the results of the classifier. On the other hand, on the last utterance, we have much better performances with the recurrent models. It follows the expected hierarchy RNN, GRU, LSTM. Therefore, as expected, the context allows to greatly improve the performance as the dialog goes on. It is confirmed by the results on the last utterance.

5 Conclusion

This study aimed to evaluate the influence of context in intent classification by comparing various encoders and decoders. Our findings indicate that context is a crucial factor that affects the accuracy of classification, and the selection of decoder architecture, such as GRU, RNN, or LSTM, can have a significant impact on performance. Additionally, we recognize the significance of capturing the dynamic changes in context, such as past vs. full, position, and other factors relevant to real-world applications. Moving forward, we believe that a crucial research direction would be to address fairness [Colombo et al., 2022, Pichler et al., 2022, Colombo et al., 2021b] in intent classification, as biases in data and models can have significant implications in decision-making processes. Therefore, it is crucial to consider fairness concerns when developing such models.

References

E. Chapuis, P. Colombo, M. Manica, M. Labeau, and C. Clavel. Hierarchical pre-training for sequence labelling in spoken dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2636–2648, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.239. URL https://aclanthology.org/2020.findings-emnlp.239.

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

P. Colombo*, W. Witon*, A. Modi, J. Kennedy, and M. Kapadia. Affect-driven dialog generation. *NAACL 2019*, 2019.

P. Colombo, E. Chapuis, M. Manica, E. Vignon, G. Varni, and C. Clavel. Guiding attention in sequence-to-sequence models for dialogue act prediction. 2020. doi: 10.48550/ARXIV.2002.08801. URL https://arxiv.org/abs/2002.08801.

P. Colombo, E. Chapuis, M. Labeau, and C. Clavel. Code-switched inspired losses for spoken dialog representations. In *EMNLP 2021*, 2021a.

P. Colombo, C. Clavel, and P. Piantanida. A novel estimator of mutual information for learning to disentangle textual representations. *ACL 2021*, 2021b.

P. Colombo, G. Staerman, N. Noiry, and P. Piantanida. Learning disentangled textual representations via statistical measures of similarity. *ACL 2022*, 2022.

T. Dinkar*, P. Colombo*, M. Labeau, and C. Clavel. The importance of fillers for text representations of speech transcripts. *EMNLP 2020*, 2020.

A. Garcia*, P. Colombo*, S. Essid, F. d'Alché Buc, and C. Clavel. From the token to the review: A hierarchical multimodal approach to opinion mining. *EMNLP 2019*, 2019.

H. Jalalzai*, P. Colombo*, C. Clavel, É. Gaussier, G. Varni, E. Vignon, and A. Sabourin. Heavy-tailed representations, text polarity classification & data augmentation. *NeurIPS 2020*, 2020.

Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan, Nov. 2017. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I17-1099.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. 2019. URL https://arxiv.org/abs/1907.11692.

T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. Ms marco: A human generated machine reading comprehension dataset. 2016. URL https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.

G. Pichler, P. J. A. Colombo, M. Boudiaf, G. Koliander, and P. Piantanida. A differential entropy estimator for training neural networks. In *ICML 2022*, 2022.

S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea. Meld: A multimodal multi-party dataset for emotion recognition in conversations, 2018.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, Y. Z. Michael Matena, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. 2019. URL https://arxiv.org/abs/1910.10683.

N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey. 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, Apr. 30 - May 1 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-2319.

K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu. Mpnet: Masked and permuted pre-training for language understanding. 2020. URL https://arxiv.org/abs/2004.09297.

H. Thompson, A. Anderson, E. Bard, G. Doherty-Sneddon, A. Newlands, and C. Sotillo. The hcrc map task corpus: natural dialogue for speech recognition. 01 1993. doi: 10.3115/1075671.1075677.

W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pretrained transformers. 2020. doi: 10.48550/arXiv.2002.10957. URL https://arxiv.org/abs/2002.10957.

Appendix

A Details on the accuracy of each model with different fixed numbers of parameters

We present every classifier with different amounts of parameters and compare their accuracy. We notice an almost linear increase for GRU and LSTM in uni and bidirectional versions. However, the results of the RNN are more unstable. Then, as we develop in our article, the most important differences are between past and full context and between the models, GRU and LSTM achieving better results.

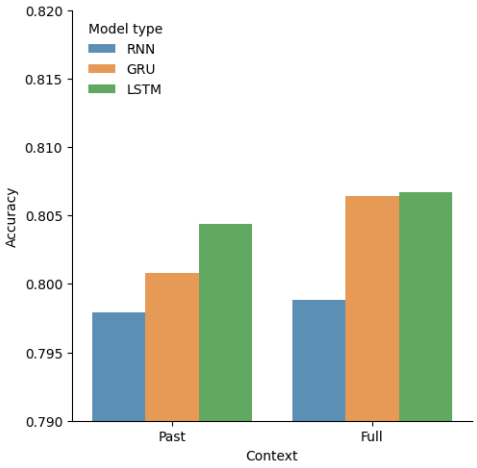

Figure 4: Accuracy depending on the decoder, for a fixed maximum number of parameters, about 100k (decoder: gtr-t5-large)

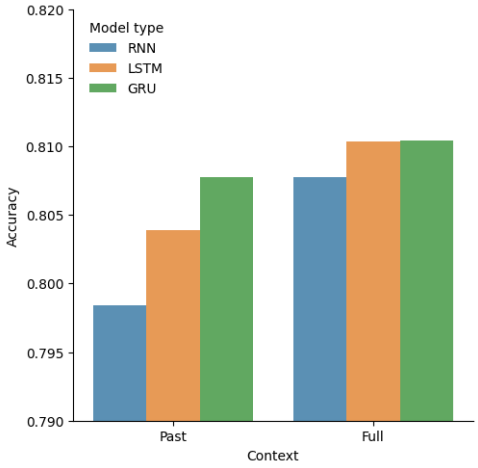

Figure 6: Accuracy depending on the decoder, for a fixed maximum number of parameters, about 300k (decoder: gtr-t5-large)

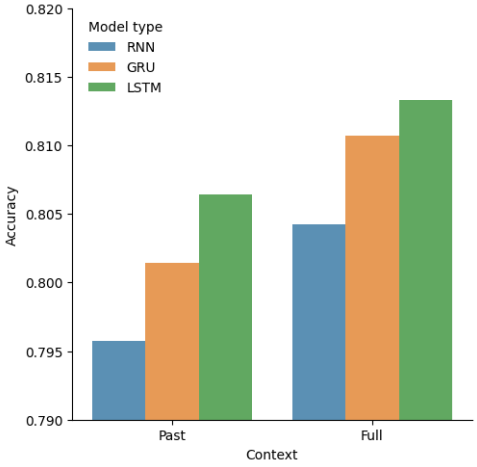

Figure 5: Accuracy depending on the decoder, for a fixed maximum number of parameters, about 200k (decoder: gtr-t5-large)

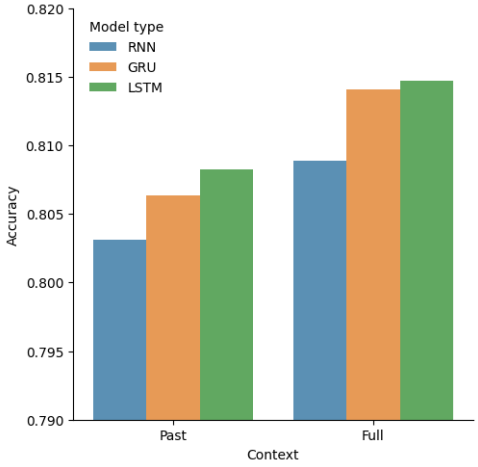

Figure 7: Accuracy depending on the decoder, for a fixed maximum number of parameters, about 400k (decoder: gtr-t5-large)

B Details of accuracy for every model with different encoder, classifier, and parameters

Here are the details of our models, with every encoder-decoder architecture. We also specify the number of parameters, direction, and the accuracy each model reaches. We notice that the decoder gtr-t5-large provides the best results, with results higher than 0.8 for every classifier, except for the MLP.

Encoder	Direction	Model	Accuracy	Nb. parameters
MiniLM	None	MLP	0.700	26k
	Uni	RNN	0.755	69k
		GRU	0.767	207k
		LSTM	0.778	276k
	Bi	RNN	0.782	718k
		GRU	0.784	473k
		LSTM	0.792	357k
mpnet	None	MLP	0.786	50k
	Uni	RNN	0.770	107k
		GRU	0.787	138k
		LSTM	0.798	510k
	Bi	RNN	0.791	53k
		GRU	0.799	704k
		LSTM	0.813	6,9M
gtr-t5	None	MLP	0.773	50k
	Uni	RNN	0.806	507k
		GRU	0.801	322k
		LSTM	0.810	429k
	Bi	RNN	0.813	630k
		GRU	0.816	704k
		LSTM	0.819	3,5M
RoBERTa	None	MLP	0.779	66k
	Uni	RNN	0.776	133k
		GRU	0.785	399k
		LSTM	0.801	612k
	Bi	RNN	0.794	733k
		GRU	0.805	858k
		LSTM	0.819	3,9M

Table 3: Accuracy for each model and each encoder