Impact of Encoder Architecture and Input Features on Dialogue Act Classification: A Comparative Study of RNN Encoders

Salah-eddine El Mouslih*
ENSAE
salah-eddine.elmouslih@ensae.fr

Imad Boudroua*
ENSAE
imad.boudroua@ensae.fr

Abstract

This paper examines the impact of encoder architecture and input features on dialogue act classification, an important task in dialogue systems. We conduct several experiments comparing the performance of different recurrent neural network encoders. These include a GRU encoder initialized with BERT weights that considers only the previous utterance as context, and the same model with speaker-level embeddings. We also compare two variants of a bi-directional LSTM encoder for dialogue act classification: one that takes multi-utterance conversations of BERT pooled outputs with and without speaker-level embeddings, and another that averages the LSTM layer outputs. Our findings indicate that incorporating a bi-directional LSTM encoder with BERT's pooled representation improves classification performance significantly.

1 Introduction

Accurately identifying the dialogue act in a conversation is crucial for building effective and reliable conversational agents such as chatbots, virtual assistants, and voice assistants like Siri, Alexa, and Google Home. These agents are designed to interact with users using natural language, and dialogue act classification is essential to understand the intention behind the user's input and generate an appropriate response (Colombo et al., 2021b; Jalalzai* et al., 2020; Colombo* et al., 2019). It has been widely studied and applied in various NLP applications, including customer service, personal assistants, education, and healthcare. The ability to accurately classify dialogue acts not only enhances the user experience but also improves the efficiency and effectiveness of these systems.

In recent years, extensive research has been conducted to develop efficient models for DA labeling. These models can be classified into two categories: single-sentence and contextual models. Single-sentence models predict the corresponding DA label of a single utterance, while contextual models require historical or contextual information, such as previous dialogue utterances, previously predicted DA labels, or a change in speaker, to predict the DA label of each utterance. Incorporating contextual information has been shown to improve performance compared to single-sentence models (Lee and Dernoncourt, 2016; Liu and Lane, 2017; Chandrakant Bothe and Wermter, 2018; Colombo et al., 2021a; Chapuis* et al., 2020). Moreover, various approaches falling within the same category have utilized speaker information, resulting in a significant enhancement of performance. (Shang et al., 2020; Bothe et al., 2018; He et al., 2021; Colombo et al., 2020)

In this paper, we aim to investigate the impact of encoder architecture and input features on dialogue act classification. We compare two RNN encoders, a GRU (Cho et al., 2014) encoder initialized with BERT (Devlin et al., 2019) weights that considers only the previous utterance as a context, and the same model incorporating speaker-level embeddings. Furthermore, we evaluate two variations of a bi-directional LSTM (Graves et al., 2005) encoder that leverage BERT outputs: one that takes the pooled representation and another that takes an average of the last LSTM layer. We also examine the impact of speaker-level embeddings on the performance of the bi-directional LSTM encoder for the first variation.

The source code for this research has been made publicly available on github: [1].

[1] https://github.com/ib-proj/intent-classification/

2 Problem Framing

The Problem can be formally defined as follows. At the highest level, we have a set $\mathcal{D}$ of N conversations, each consisting of a sequence of utterances $C_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,|C_i|})$ and a corresponding set of dialogue act labels Y_i.

At a lower level, each utterance $u_{i,j}$ is associated with a unique label $Y_{i,j}$.

We also consider the corresponding sequence of speaker turns $S_{i,j} \in 0, 1$ for the j-th utterance in the i-th conversation.

Our initial goal is to consider only the previous utterance as context and incorporate the speakers turn in a later stage. Next, we intend to combine the utterances at the conversation level before including the respective speakers in the Bi-LSTM variants.

3 Models

3.1 GRU Encoder

This model utilizes transfer learning through the BERT model to obtain a representation for each utterance in a conversation, which has been shown to be effective in downstream tasks such as Dialogue Act Classification (Noble and Maraev, 2021). In order to further enhance the contextual awareness of each utterance, we compute a representation for the previous utterance within the same conversation using BERT. This approach is inspired by (Chandrakant Bothe and Wermter, 2018), who demonstrated that considering the most recent preceding utterances can significantly improve classification accuracy for short utterances.

To capture long-term dependencies between the utterance and its context, we merge the two representations and input them into a GRU model. In order to optimize the model, we perform context computation during the pre-processing phase instead of the forward pass for each utterance.

3.1.1 Speaker Modeling

To incorporate speaker information, the model utilizes the previous speaker turn s_{t-1} and the current speaker turn s_t to derive the corresponding speaker embeddings e_{t-1} and e_t.

Specifically, the context vector c_{t-1} is first encoded with BERT and then combined with the previous speaker embedding e_{t-1} to obtain the final output. Similarly, the current utterance representation is passed through BERT and then combined with the current speaker embedding e_t.

Finally, the resulting representations are passed through the GRU layer in a similar manner to the first model.

3.2 Baseline LSTM Encoder

We employed an LSTM model on utterance-level representations provided by a BERT model. This straightforward approach was implemented to investigate whether capturing the interdependencies among utterances is sufficient for accurate classification.

3.3 Bi-LSTM Encoder

In this part, we propose an extension to the previous encoders that takes into account the entire conversation as the input context.

For this, we use a combination of BERT to model the temporal dependencies in a sequence of text inputs as well as a bi-directional LSTM (bi-LSTM) as they are the most widely used architecture (Kumar et al., 2017).

3.3.1 Context Modeling

We use two different techniques to encode multi-utterance conversations.

The first technique (i) involves leveraging the powerful representations obtained from BERT. Following the approach of (He et al., 2021), we extract the embeddings of the [CLS] token from each input sequence u_t for each conversation in the batch. This token represents the entire utterance in BERT. We then pass the resulting embeddings through a bi-directional LSTM layer to capture inter-utterance dependencies. We refer to this model as $BI - LSTM_{cls}$.The second technique (ii) involves using the resulting embeddings of the entire utterances and passing them to a bi-directional LSTM layer. However, in this case, the LSTM output is averaged across the sequence length to obtain a single feature vector for each utterance, denoted as $\mathbf{v} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{h}_i$. We refer to this model as $BI - LSTM_{avg}$.

Finally, we pass the pooled output from either $BI - LSTM_{cls}$ or $BI - LSTM_{avg}$ through a linear layer to obtain logits for dialogue act classification.

3.3.2 Speaker turn Modeling

in the second stage, we introduce a layer of speaker embeddings to the input context in the

$BI - LSTM_{cls}$ model. This modification aims to investigate whether incorporating speaker turns can enhance the model's ability to capture temporal dependencies and speaker turns in the conversation.

Specifically, the model incorporates speaker embeddings to take into account the speaker turns when predicting the dialogue act, where $S_{i,j} \in 0, 1$ is the speaker for the j-th utterance in the i-th conversation.The input embeddings for the bi-directional LSTM are formed by adding the BERT output embeddings $\mathbf{e}_i$ and speaker embeddings $\mathbf{s}_i$ element-wise. This is denoted as $\mathbf{H} = \mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n$, where $\mathbf{h}_i = \mathbf{e}_i + \mathbf{s}_i$. This approach was inspired by the positional encoding technique used in Transformers (Vaswani et al., 2017) and the work of (He et al., 2021).

The Bi-LSTM is then used to capture all the contextual dependencies within the sequence:

$$\overrightarrow{\mathbf{h}}_i = \overrightarrow{LSTM}(\mathbf{h}_i, \overrightarrow{\mathbf{h}}_{i-1})$$
$$\overleftarrow{\mathbf{h}}_i = \overleftarrow{LSTM}(\mathbf{h}_i, \overleftarrow{\mathbf{h}}_{i+1})$$
$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i]$$

where $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$ are the forward and backward LSTM outputs for the i-th utterance, respectively.

Finally, a linear layer is used to produce the predicted label logits in a similar fashion to the first stage.

4 Experiments Protocol

4.1 Data Pre-processing

We employed the **S**witchboard **D**ialogue **A**ct **C**orpus (SwDa) (Jurafsky et al., 1997), a dataset of transcribed telephone conversations between strangers, annotated with dialogue act labels such as statements, questions, and backchannels, using the **D**ialog **A**ct **M**arking in **S**everal **L**anguages (DAMSL) annotation scheme with 43 labels (Core and Allen, 1997), to train our models for dialogue act classification. The data have already been partitioned into train, validation, and test sets.

Although several alternative exists (Li et al., 2017; Leech and Weisser, 2003; Busso et al., 2008; Passonneau and Sachar., 2014; Thompson et al., 1993; Poria et al., 2018; Mckeown et al., 2013), we choose SwDa as it is the largest available dataset.

Before training our models on the dataset, we performed several pre-processing steps. Firstly, we cleaned the text by removing unnecessary punctuation, non-alphanumeric characters, and converting the text to lowercase. Next, we mapped each speaker to a binary value of 0 or 1. After mapping the speakers, we tokenized the utterances into a sequence of tokens.

For **bi-LSTM** models, we grouped 30 utterances of length 128 each into conversations so that the model can accept multi-utterance conversations as input. We then encoded and padded sequences using the BERT model to obtain a fixed-size representation for each utterance.

4.2 Training

In this section, we provide a comprehensive overview of the training procedures used to train our proposed models.

To accomplish this, we use cross-entropy as our loss function for all models. This function is widely used in classification tasks and measures the dissimilarity between the predicted probability distribution and the ground truth label for each training instance. We then apply backpropagation and stochastic gradient descent to minimize the cross-entropy loss and update the model parameters during training.

To further optimize the training process, we leverage the Adam optimizer proposed by Kingma and Ba (2014). Specifically, we set the learning rate to 0.001 for the GRU encoders and LSTM models, while for the bi-directional LSTMs, we set it to 0.0001.

All models have been implemented in PyTorch and trained with a patience of 5 epochs to a maximum number of 10 epochs on a single NVIDIA P100.

5 Results

Model	Without Speaker	With Speaker
GRU Encoder	63.0	61.3
$LSTM_{Base}$	65.9	-
$BI - LSTM_{avg}$	74.6	-
$BI - LSTM_{cls}$	**88.6**	85.5

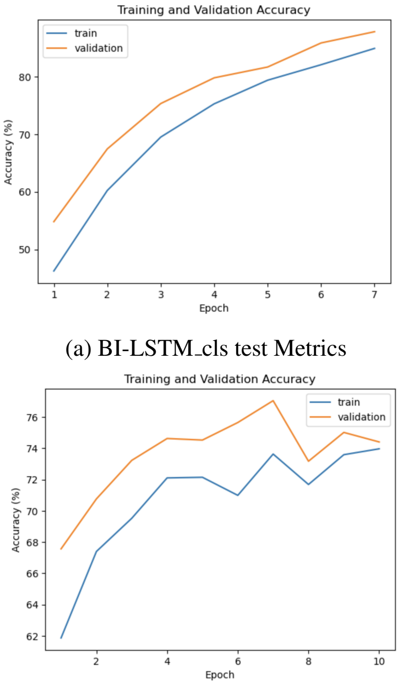
Table 1: Test accuracies of all models.

Table 1 shows that the $BI - LSTM_{cls}$ model outperforms its variants in terms of accuracy. Furthermore, there is only a small difference in the accuracies of the GRU encoders.

Model	Precision	Recall	F1 Score
GRU Without speaker	63.1	63.0	59.3
GRU With speaker	63.8	61.2	58.3
LSTM Base Encoder	65.1	66.0	63.7
$BI-LSTM_{avg}$	75.7	74.6	71.6
$BI-LSTM_{cls}$ without speaker	89	88.6	87.9
$BI-LSTM_{cls}$ with speaker	85.1	85.5	83.9

Table 2: Models Weighted Metrics Comparison

Table 2 shows the precision, recall, and F1 score for various models evaluated in the study, including GRU models with and without speaker-level embeddings, a base LSTM encoder, and different variations of a bi-directional LSTM encoder that leverages the 'cls' representation or an average of the LSTM layer.



(a) BI-LSTM_cls test Metrics

(b) BI-LSTM_avg test Metrics

Figure 1: BI-LSTM_avg and BI-LSTM_cls Visual comparaison

Figure 1 shows a visual comparison of the BI-LSTM-avg and BI-LSTM-cls models in terms of loss and accuracy metrics.

6 Discussion

6.1 GRU Encoders

During our experimentation with the GRU model, we observed a case of underfitting, as depicted in Figure 10. Despite this challenge, we were able to achieve a reasonably high accuracy of 63.0 on the test set, as shown in Figure 3 in the appendix. However, due to computational constraints, we were unable to conduct an extensive hyperparameter search, which could have further optimized the model's performance.

Moreover, our results indicate that such information did not significantly improve the model's performance, thereby suggesting that knowledge of the speaker in the previous context is not helpful.

6.2 LSTM Encoder

The LSTM model outperformed the GRU model, even though it exhibited slight underfitting due to computational and time constraints (figure 8). This model captures dependencies between utterances without considering the conversation level. It achieved an accuracy of 65.6 on the test set (figure 4 appendix).

6.3 Bi-LSTM Encoders

Our study shows that incorporating conversation context, including previous and next utterances, results in significant improvement in classification performance compared to previous models.

Specifically, our results indicate that the bi-directional LSTM encoder that leverages the pooled output of the `[CLS]` token, denoted as $BI-LSTM_{cls}$, outperformed other models, including $BI-LSTM_{avg}$ that averages the output of the LSTM layer. The reason for the superior performance of $BI-LSTM_{cls}$ compared to $BI-LSTM_{avg}$ can be attributed to the fact that the `[CLS]` token captures more information about the input sentence compared to the output of the LSTM layer. The `[CLS]` token is specifically trained in BERT to represent the whole sentence and can provide a stronger representation of the input utterance.

Furthermore, we noted that the training of $BI-$

$LSTM_{cls}$ had to be terminated at the seventh epoch due to overfitting. This behavior might be due to the relatively small number of utterances per conversation, which was limited to 30 in our study.

Interestingly, we found that incorporating speaker turns, contrary to previous literature, did not improve classification accuracy.

7 Conclusion

In conclusion, our study suggests that incorporating conversation context, including previous and next utterances, can result in significant improvement in classification performance compared to models that do not consider context. Our experiments indicate that the bi-directional LSTM encoder that leverages the pooled output of the `[CLS]` token, $BI - LSTM_{cls}$, outperformed other models, achieving an accuracy score of 88.6

On the other hand, our experiments with GRU and LSTM encoders revealed underfitting due to computational and time constraints, resulting in lower accuracy scores of 63.0% and 65.6%, respectively.

We also found that incorporating speaker turns did not significantly improve classification accuracy.

Future research in the area of dialogue act classification will focus on ensuring fairness in the models developed for this task. It is important to ensure that the models are not biased towards certain groups of people and do not perpetuate discriminatory practices (Colombo et al., 2022; Pichler et al., 2022). This requires careful consideration of the data used to train the models and the evaluation metrics used to assess their performance. Additionally, research will explore the use of alternative approaches to training models, such as adversarial training and data augmentation, to improve the overall fairness of the models.

References

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.

Mark G Core and James Allen. 1997. Coding dialogs with the damsl annotation scheme. In *AAAI Fall Symposium on Communicative Action in Humans and Machines*, volume 56, pages 28–35, Cambridge, MA. AAAI Press.

Daniel Jurafsky, Elizabeth Shriberg, Debra Biasca, Kirsten Breckenridge, Tree Fox, Robert Katz, Rachel Martin, and Patti Price. 1997. Switchboard swbd-damsl shallow-discourse-function annotation coders manual. Technical Report CSLI-97-3, Center for the Study of Language and Information.

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

Alex Graves, Santiago Fern'andez, and J"urgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, volume 3697 of *Lecture Notes in Computer Science*, pages 849–854. Springer.

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.

Gary Mckeown, Michel Valstar, Roddy Cowie, Maja Pantic, and M. Schroder. 2013. The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *Affective Computing, IEEE Transactions on*, 3:5–17.

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

Kyunghyun Cho, Dzmitry van Merrienboer, Bougares, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Lee and Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827*.

Liu and Lane. 2017. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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

Harshit Kumar, Arvind Agarwal, Riddhiman Dasgupta, Sachindra Joshi, and Arun Kumar. 2017. Dialogue act sequence labeling using hierarchical encoder with crf. *arXiv preprint arXiv:1709.04250*.

Chandrakant Bothe, Cornelius Weber, Sven Magg, and Stefan Wermter. 2018. A context-based approach for dialogue act recognition using simple recurrent neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Cornelius Weber Chandrakant Bothe, Sven Magg and Stefan Wermter. 2018. Conversational analysis using utterance-level attention-based bidirectional recurrent neural networks. *arXiv preprint arXiv:1805.06242v2*.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations.

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.

Colombo, Chapuis E, Manica M, Vignon E, Varni G, and Clavel C. 2020. Guiding attention in sequence-to-sequence models for dialogue act prediction. *arXiv preprint arXiv:2002.08801*.

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*.

Emile Chapuis*, Pierre Colombo*, Matteo Manica, Matthieu Labeau, and Chloe Clavel. 2020. Hierarchical pre-training for sequence labelling in spoken dialog. *Finding of EMNLP 2020*.

Guokan Shang, Antoine J.-P. Tixier, Michalis Vazirgiannis, and Jean-Pierre Lorre. 2020. Speaker-change aware CRF for dialogue act classification. In *Proceedings of the 28th International Conference on Computational Linguistics*.

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

Bill Noble and Vladislav Maraev. 2021. Large-scale text pre-training helps with dialogue act recognition, but not without fine-tuning. In *Proceedings of the 14th International Conference on Computational Semantics*.

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

Zihao He, Leili Tavabi, Kristina Lerman, and Mohammad Soleymani. 2021. Speaker turn modeling for dialogue act classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2150–2157.

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*.

A Training accuracy of models

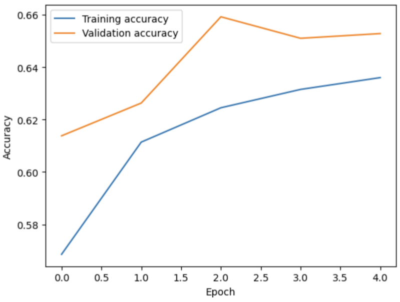


Figure 2: Training and validation accuracy with epochs: GRU encoder

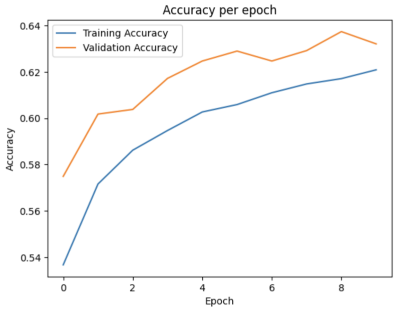


Figure 3: Training and validation accuracy with epochs: GRU encoder with speaker

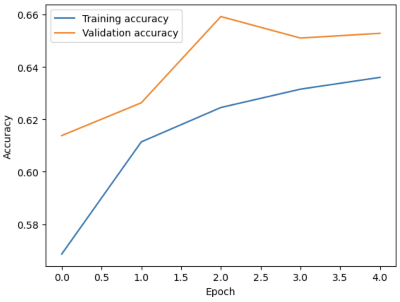


Figure 4: Training and validation accuracy with epochs: LSTM encoder

B Trainig loss of models

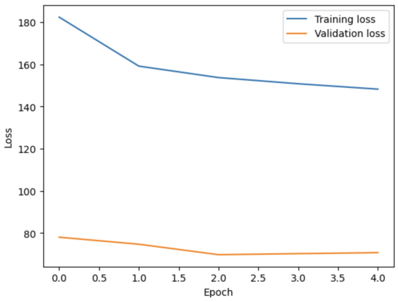


Figure 5: Training and validation loss with epochs: GRU encoder

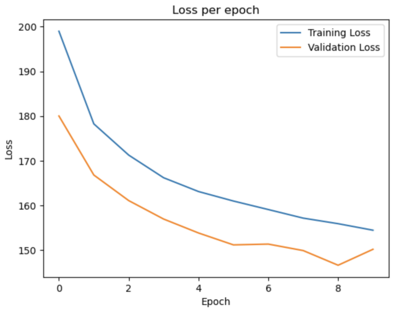


Figure 6: Training and validation loss with epochs: speaker-level GRU encoder

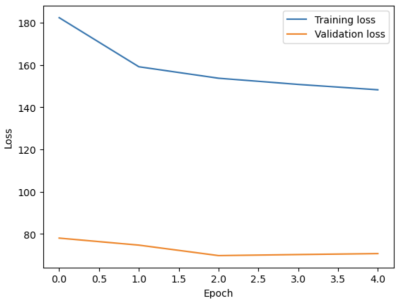


Figure 7: Training and validation loss with epochs: model with LSTM encoder

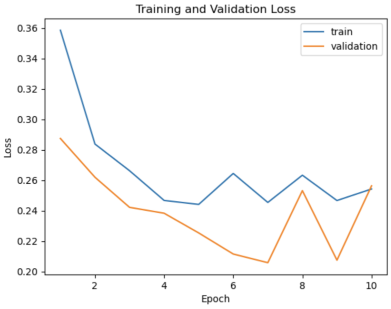


Figure 8: Training and validation loss with epochs: bi-LSTM_avg encoder

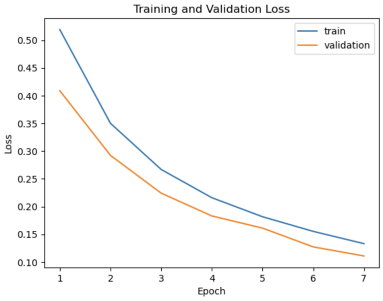


Figure 9: Training and validation loss with epochs: bi-LSTM_cls encoder

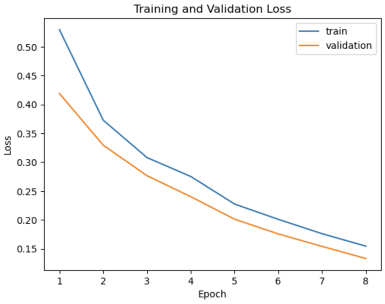


Figure 10: Training and validation loss with epochs: speaker aware bi-LSTM_cls encoder