	TINY	SMALL
Nbs of heads	1	6
N_d	2	4
N_u	2	4
T	50	50
C	5	5
$\mathcal{T}_d$ nbs of heads	6	6
Inner dimension	768	768
Model Dimension	768	768
Vocab length	32000	32000
$\mathcal{T}_d$: Emb. size	768	768
d_k:	64	64
d_v:	64	64

Table 8: Architecture hyperparameters used for the hierarchical pre-training.

A Additional Details on data composing SILICONE

In this section, we illustrate the diversity of the dataset composing SILICONE. In Figure 3, we plot two histograms representing the different utterance lengths for DA and E/S. As expected, for spoken dialog, lengths are shorter than for written benchmarks (e.g., GLUE).

B Additional Details for Models

In this section we report model hyper-parameters and as well as additional descriptions of our baselines. For all models we use a tokenizer based on WordPiece (Wu et al., 2016).

We also provide a concrete example of corrupted context for the MLM Loss.

B.1 Hierarchical pre-training

We report in Table 8 the main hyper-parameters used fo our model pre-training. We used GELU (Hendrycks and Gimpel, 2016) activations and the dropout rate (Srivastava et al., 2014) is set to 0.1.

B.2 MLM Loss example

In this section we propose a visual illustration of the corrupted context Figure 4 by the MLM Loss.

B.3 Experimental Hyper-parameters for SILICONE

For all models, we use a batch size of 64 and automatically select the best model on the validation set according to its loss. We do not perform exhaustive grid search either on the learning rate (that is set to 10^{-4}), nor on other hyper-parameters to perform a fair comparison between all the models. We use ADAMW (Kingma and Ba, 2014; Loshchilov and Hutter, 2017) with a linear scheduler on the learning rate and the number of warm-up steps is set to 100.

B.4 Additional Details on Baselines

A representation for all the baselines can be found in Figure 5. For all models, both hidden dimension and embedding dimension is set to 768 to ensure fair comparison with the proposed model. The MLP used for decoding contains 3 layers of sizes $(768, 348, 192)$. We use RELU (Agarap, 2018) to introduce non linearity inside our architecture.

C Additional Experimental Results

In this section we report the detailed results on SILICONE, including the ones presented in Table 4. We report results on two new experiments: importance of pre-training time for both a TINY and SMALL model, we report the convergence time of a TINY model and finally we extend subsubsection 5.2.3 by reporting results on IEMO.

C.1 Detailed Results on SILICONE

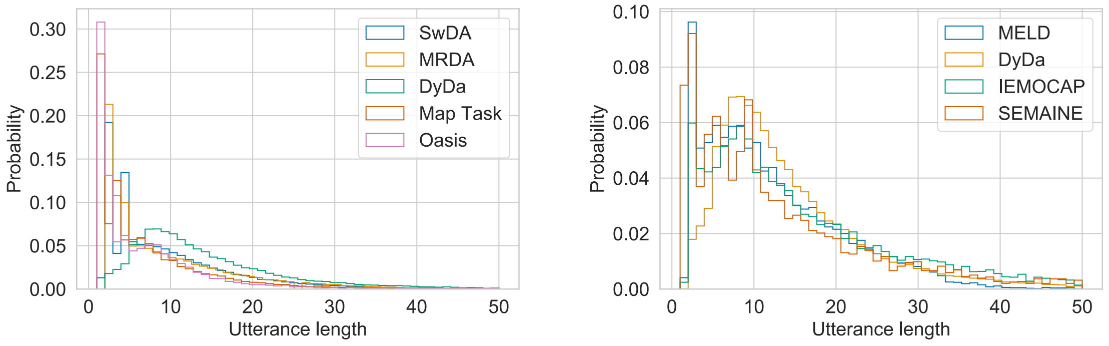
We show in Table 9 the results on the SILICONE benchmark for all the models mentioned in the paper.

C.2 Improvement over pre-training

In this experiment we illustrate how pre-training improves performance on SEM (see Figure 6). As expected accuracy improves when pre-training.

C.3 Multi level Supervision for pre-training MELD

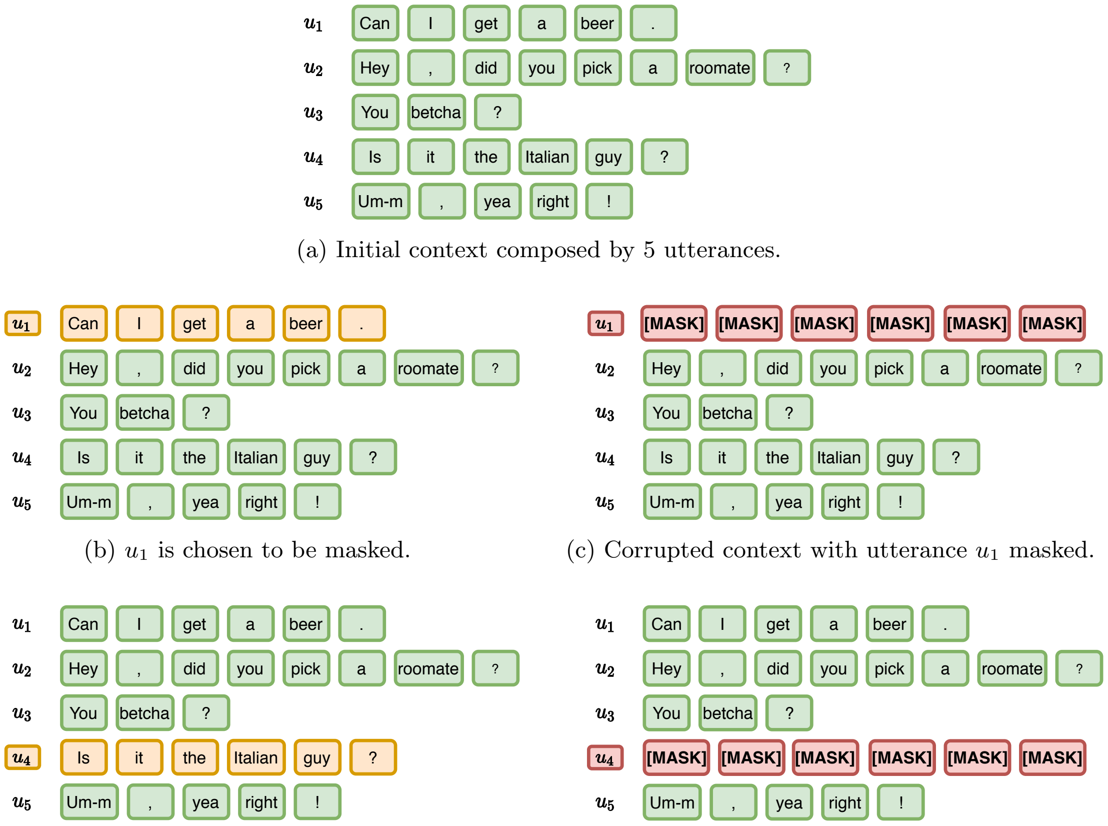
In this experiment we report results of the experiment mentioned in subsubsection 5.2.3. In this experiment we see that the training process seems to be noisier for fractions lower than 40%. For larger percentages, we observe that including higher supervision (at the dialog level) during pre-training leads to a consistent improvement.



(a) `SILICONE DA`

(b) `SILICONE S/E`

Figure 3: Histograms showing the utterance length for each dataset of `SILICONE`.


u_1 Can I get a beer .
u_2 Hey , did you pick a roomate ?
u_3 You betcha ?
u_4 Is it the Italian guy ?
u_5 Um-m , yea right !
[MASK] [MASK] [MASK] [MASK] [MASK] [MASK]

(a) Initial context composed by 5 utterances.

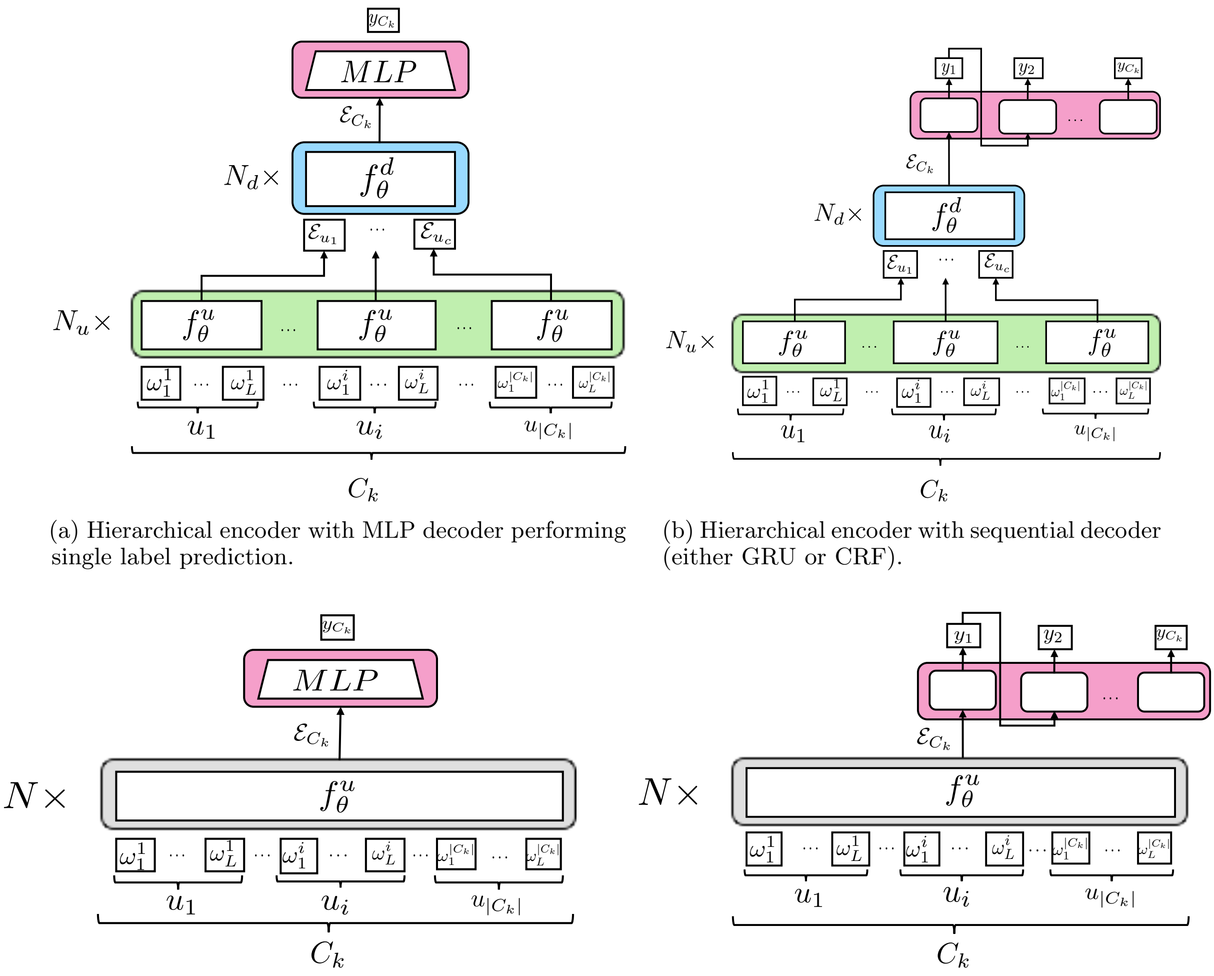
(b) u_1 is chosen to be masked.

(c) Corrupted context with utterance u_1 masked.

(d) u_4 is chosen to be masked.

(e) Corrupted context with utterance u_4 masked.

Figure 4: This figure shows an example of corrupted context. Here p_C is randmoly set to 2 meaning that two utterances will be corrupted. u_1 and u_4 are randomly picked in 4b, 4d and then masked in 4c, 4e.



(a) Hierarchical encoder with MLP decoder performing single label prediction.

(b) Hierarchical encoder with sequential decoder (either GRU or CRF).

(c) BERT encoder with MLP decoder performing single label prediction.

(d) BERT encoder with sequential decoder (either GRU or CRF)

Figure 5: Schema of the different models evaluated on `SILICONE`. In this figure f_θ^u, f_θ^d and the sequence label decoder (g_θ^{dec}) are respectively colored in green, blue and red for the hierarchical encoder (see Figure 5a and Figure 5d). For BERT there is no hierarchy and embedding is performed through f_θ^u colored in grey (see Figure 5c, Figure 5d)

	Avg	`SwDA`	`MRDA`	`DyDA`$_{\texttt{DA}}$	`MT`	`Oasis`	`DyDA`$_{\texttt{e}}$	`MELD`$_{\texttt{s}}$	`MELD`$_{\texttt{e}}$	`IEMO`	`SEM`
BERT-4layers (+MLP)	69.45	77.8	90.7	79.0	88.4	66.8	90.3	49.3	50.4	43.0	58.8
BERT (+MLP)	72.79	79.2	90.7	82.6	88.2	66.9	91.9	59.3	61.4	45.0	62.7
BERT (+GRU)	69.84	78.2	90.4	80.8	88.7	63.7	90	50.4	48.9	45.0	62.3
BERT (+CRF)	72.8	79.0	90.8	88.3	67.2	81.9	91.5	59.4	61.0	44.2	61.5
$\mathcal{HR}$ (+MLP)	69.77	77,5	90,9	80,1	82,8	64,3	91.5	59,3	59.9	40.3	51.1
$\mathcal{HR}$ (+GRU)	67.54	78.2	90.9	79,9	84,4	63,5	91.5	50,7	50.4	35.2	50.7
$\mathcal{HR}$ (+CRF)	70.5	77.8	91,3	79,7	87,5	65,3	91,1	62,1	57,4	42.1	50.7
$\mathcal{HT}(\theta_{MLM}^{u,d})$ (TINY)	73.3	79.3	92.0	80.1	90.0	68,3	92.5	62.6	59.9	42.0	66.6
$\mathcal{HT}(\theta_{MLM}^{d})$ (TINY)	72.4	78.5	91.8	78.0	89.8	66.0	92.5	62.6	59.3	42.0	63.5
$\mathcal{HT}(\theta_{MLM}^{u})$ (TINY)	72.4	78.6	91.8	79.0	89.8	65.0	91.8	61.8	58.1	39.2	68.9
HBERT (w) θ_{BERT_milmil} (TINY)	70.8	77.6	91.4	79.3	88.3	65.8	91.9	58.0	56.3	40.0	59.1
$\mathcal{HT}(\theta_{MLM}^{u,d})$ (SMALL)	74.32	79.2	92.4	81.5	90.6	69.4	92.7	64.1	60.1	45.0	68.2
$\mathcal{HT}(\theta_{GAP}^{d})$ (TINY)	71.58	78.6	91.8	78.1	89.3	64.1	91.6	60.5	55.7	42.2	63.9
$\mathcal{HT}(\theta_{GAP}^{u})$ (TINY)	71.52	78.5	90.9	79.0	88.9	66.3	92.0	59.2	57.5	39.9	63.0

Table 9: Performances of all mentioned model with different decoders such as MLP, GRU, CRF `SILICONE`. The datasets are grouped by label type (`DA` vs `E/S`) and order by decreasing size.

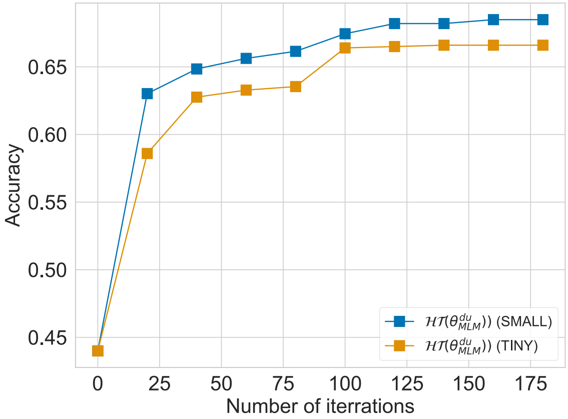


Figure 6: Illustration of improvement of accuracy during pre-training stage on `SEM` for both a `TINY` and `SMALL` model.

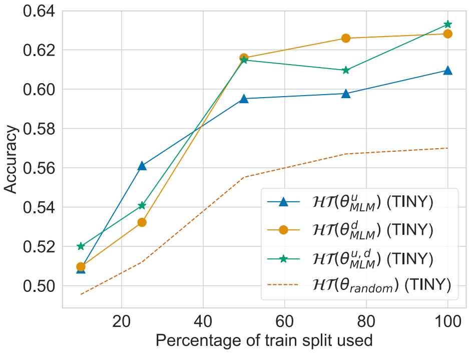


Figure 7: A comparison of different parameters initialisation on $\texttt{MELD}_\texttt{s}$. Training is performed using a different percentage of complete training set. Validation and test set are fixed over all experimentation. Each score is the averaged accuracy over 10 random runs.

D Negative Results on `GAP`

We briefly describe few ideas we tried to make `GAP` works at both the utterance and dialog level. We hypothesise that:

- giving the same weight to the utterance level and the dialog level (see Equation 3) was responsible of the observed plateau. Different combinations lead to fairly poor improvements.

- the limited model capacity was part of the issue. Larger models does not give the expected results.