What do the GuessWhat?! Guesser models learn and what do they learn it from?

Atanasoska Tamara, and **Ryazanskaya Galina,** and **Verma Bhuvanesh**
University of Potsdam

Abstract

This paper explores the multimodal representations learned by grounded language comprehension models on the GuessWhat?! task. Following Greco et al. (2020), we use dialogue history permutations to investigate whether the models are able to extract salient semantic features from the dialogues. We compare blind and multimodal models with LSTM (LSTM, V-LSTM) and Transformer architectures (RoBERTa, LXMERT). Additionally, we explore the effect that the input features have on the models' ability to learn such semantically rich representations. We find that a significant share of the Guesser accuracy is dependent on the candidate object category being provided as the model input. There is a degree of interaction between the model input and the performance on the permuted history experiments. This holds for all models except LXMERT, which performs relatively well even without object category, possibly due to extensive multimodal pretraining.

1 Introduction

Grounded language comprehension is an area of active research, and several multimodal datasets have recently been proposed for model evaluation. These include purely referential datasets such as ReferIt (Kazemzadeh et al., 2014) as well as visual dialogue datasets such as GuessWhich (Das et al., 2017) and GuessWhat?! (De Vries et al., 2017), which require referring expression resolution as a sub-task in the two-agent visual dialogue game. All these tasks challenge the ability of the models to learn multimodal object representations from the images and the linguistic data in the dataset. We use the GuessWhat?! game (De Vries et al., 2017), specifically, the final Guesser task, as the test-bed for our experiments. The task of the Guesser in the GuessWhat?! game is to identify the target object from a list of candidate objects based on an image and a series of yes/no question-answer pairs. This paper, building upon the experiments conducted by Greco et al. (2020), assesses the effect that the method of encoding the dialogue has on the Guesser accuracy. We also investigate the ability of various models to extract salient semantic features and how this ability interacts with the encoding method and with the method of candidate object representation.

Several papers have stressed the importance of going beyond numeric comparison solely based on task success while performing model assessment in multimodal tasks (Shekhar et al., 2018; Testoni et al., 2019; Greco et al., 2020). In this paper, we adopt the approach proposed by Greco et al. (2020) and explore the effect that dialogue history permutations have on the Guesser performance as a method of semantic analysis of the learned representations. Some suggest that dialogue models might be insensitive to dialogue turn order (Sankar et al., 2019), which is in line with the recent research showing a degree of insensitivity of Transformer-based language models to permutations of the test data (Sinha et al., 2021). The models do not seem to be utilizing the positional information as much as one might have expected and are still able to extract salient features needed to resolve the referential expressions. We reproduce the experiments carried out in Greco et al. (2020), namely reversing dialogue history and removing the last turn of the dialogue. We extend the experiments to include a random shuffling of the dialogue order, aiming thus to explore what exactly the GuessWhat?! Guesser models learn.

The other question that we address in the present paper is what the GuessWhat?! Guesser models learn from, exploring which parts of the model input contribute the most to the task success. The GuessWhat?! Guesser model in the established architecture receives as its input an image and a dialogue history, which are both encoded with a multimodal Encoder. The Guesser classifier then receives a list of representations of the candidate objects in the image and selects the target object based on these representations and the visual dialogue embedding from the Encoder. The candidate object representations in the original GuessWhat?! article use the object category label from the MS-COCO manual object annotations and spatial coordinates of the object (De Vries et al., 2017). The object category is then converted from a one-hot class vector into a dense category embedding using a learned look-up table. This architecture has been used in the majority of the articles on the GuessWhat?! task. However, multiple papers (Zhuang et al., 2018; Strub et al., 2018; Suglia et al., 2020a,b) suggest that the GuessWhat?! models are overly reliant on the object category in their Guesser predictions. Greco et al. (2020) also mentions that removing the object category might strongly affect the performance. In this paper, we decided to assess the degree to which various GuessWhat?! Guesser models rely on the object category and how this reliance affects the performance on permuted dialogue history.

The Encoder components of the Guesser models that we compare, following Greco et al. (2020), vary along two dimensions: the architectural basis (LSTM vs Transformers) and the input modality (language-only (blind) vs multimodal), thus resulting in four models: LSTM, V-LSTM, RoBERTa, and LXMERT. We train each model with and without candidate object category labels to be able to single out its effect on model performance. We compare the Guesser models' accuracy on the ground truth QA pairs, as well as dialogue history with the last turn removed, reversed, and shuffled dialogue history. We thus explore how the reliance on various aspects of the input (dialogue encoding method, presence of the visual input, and presence of the object category) interacts with the ability to extract salient semantic features.

We find that:

- In line with Greco et al. (2020) findings, Transformer-based models perform better than LSTM-based models both on GT and on permuted history, and this effect is stronger than the effect of the input modality. All models seem to be relatively robust to history shuffling, more so than to history reversal. We observe better performance and higher robustness to permutations in RoBERTa than LXMERT.
- We find that all models except LXMERT are highly reliant on object category, as a 16-18% drop in accuracy is observed when the models are trained with no category information. LXMERT is the only model that is robust to this change, with a decrease of only 5.2%. This could be explained by the extensive multimodal pretraining and fine-tuning of the LXMERT model which could compensate for the absence of the category label.
- The object category contributes to the learning of salient features, as the effects of history permutations are significantly less pronounced in the models trained with no category information. Similarly, the difference between the models' performance trained with and without category labels is less pronounced in the dialogue history permutation experiments.

Acknowledging the importance of reproducibility, we deem it necessary to provide the documented code that we used for the present project [1]. The repository follows the code provided by the authors of the Greco et al. (2020), with various code improvements and additional files created to simplify the training and testing, based on the existing code. Additionally, our repository includes extended documentation to improve usability.

2 Related Work

Multimodal language processing aims to solve problems that require the integration of visual and linguistic data (Mogadala et al., 2021). Visual Dialogue tasks such as GuessWhich and GuessWhat?! is an important task in this field, as they build upon language modeling, task-oriented dialogue systems, question answering, as well as referential expression resolution and object recognition (De Vries et al., 2017; Das et al., 2017).

2.1 GuessWhat?! Task

GuessWhat?! is an asymmetric visual dialogue task, specifically, a two-player game where one agent, called the Questioner, poses a series of yes/no questions to another, called the Oracle, and then the first agent, taking on the role of the Guesser, tries to identify the target object from a list of candidates based on the image and the QA-pairs. The models used for this task usually com-

[1] https://github.com/TamaraAtanasoska/dialogue-history

partmentalize the task into sub-tasks performed by sub-models: the Encoder (encoding of the image and dialogue history); the QGen (generating the questions); the Oracle (replying to the questions); and the Guesser (selecting the target object based on the encoding of the dialogue history, of the image, and in some cases additional candidate object information, such as category labels and object image crops). Mogadala et al. (2021) provide a review of different approaches to the GuessWhat?! task.

2.2 Dialogue History Order

Multiple papers working on the GuessWhat?! task come up with various strategies of testing what the models learn (Shekhar et al., 2018; Greco et al., 2020), investigating the performance on altered data or various data subgroups separately, visualizing learned attention matrices, or exploring the linguistic properties of the generated dialogues. Greco et al. (2020), in particular, explore which turns in the dialogue are the most important to various Guesser models. It has previously been proposed, based on language-only datasets, that the neural models are insensitive to the order of turns in dialogues and might not be using the history effectively (Sankar et al., 2019). Greco et al. (2020) try to assess whether this result applies to the multimodal dialogues as well. Therefore, the authors use a visual dialogue task and compare various Encoders, assessing the contribution of the dialogue history order to the task success. They also claim that in the GuessWhat?! task the order of QA-pairs is not crucial, and humans would be able to guess the target object even if the history was reversed. Greco et al. (2020) wonder, whether the models would also have the ability to identify task-relevant information independently of the position.

The Encoder models Greco et al. (2020) compare vary along three dimensions: the architectural basis (RNN vs Transformers); the input modality (blind vs multimodal); and the utilization of the background model knowledge (models trained from scratch vs pre-trained and fine-tuned). The authors conduct multiple experiments with every model to identify the contribution of every turn to the Guesser accuracy and the ability of each model to identify salient information. They compare the models' performance on the complete unaltered dialogues, the dialogues with the last turn removed (across dialogue lengths), and the dialogues with reversed history. Additionally, the authors analyze the attention distribution by dialogue turn.

Greco et al. (2020) find that Transformer-based models generally perform the best and are more robust to various history permutations. They also show a larger drop in performance in the experiment with no last turn, and pay more attention to it, which the authors take to be a sign of the model's ability to identify the most relevant information. Testoni et al. (2021) demonstrate that people also show lower task success on dialogues with the last turn removed (21%). Importantly, Transformer-based models pay more attention to the information-dense last turn even if it presented the first, as proven by the reversed history experiment. The advantage of Transformer-based models is particularly pronounced on longer dialogues.

2.3 Permuted Language Encoding

Greco et al. (2020) findings are in line with the research showing that Transformer-based models are largely insensitive to permutations of the test and even train data (Sinha et al., 2021). Sinha et al. (2021) question the claim that masked language model (MLM) pretraining allows the models to learn representations of the syntactic structures prevalent in classical NLP pipelines. The MLM's performance is known to be quite robust to permuting downstream test data (Sinha et al., 2020; Pham et al., 2020; Gupta et al., 2021) and even to perform relatively well on permuted downstream train data (Sinha et al., 2020; Gupta et al., 2021). The authors test how syntax-disordering permutations of pretraining data affect model performance, and they find that the MLM's trained on permuted data perform surprisingly well on the downstream tasks and even some parametric probes that were designed to require syntactic insight.

2.4 GuessWhat?! Candidate Object Category Representation

The original GuessWhat?! paper proposes to use candidate object category labels as a part of the Guesser model input (De Vries et al., 2017). The category labels are then encoded and used for target object prediction. However, several papers (Zhuang et al., 2018; Strub et al., 2018; Suglia et al., 2020a,b) point out that the GuessWhat?! models are overly reliant on the object category in their Guesser predictions.

Zhuang et al. (2018) use the GuessWhat?! dataset to assess model performance in referring expression comprehension, and they were the first to point out that other datasets do not include the candidate object category information in the input, and that removing this information results in a significant drop in model performance (up to 13%).

Strub et al. (2018) explore models that use Feature-wise Linear Modulation that alternates between attending to the language input. While testing whether adding visual feedback to the context embedding improves performance, they also experimented with removing the object category and also saw a significant drop in performance. For them, adding visual feedback was more helpful in the setting without the object category, which acted as its surrogate, partially rendering multimodal learning unnecessary for task success.

Finally, Suglia and colleagues explored the issue of category in detail in their 2020 papers (Suglia et al., 2020a,b). They show that the existing models fail to learn truly multi-modal representations, relying instead on gold category labels for objects in the scene both at training and inference time. They claim that this "provides an unnatural performance advantage when categories at inference time match those at training time, and it causes models to fail in more realistic 'zero-shot' scenarios where out-of-domain object categories are involved", as shown by the aforementioned papers. They propose a possible solution to this problem in the form of an "imagination" module based on Regularized Auto-Encoders, that learns context-aware and category-aware object embeddings, instead of relying on category labels.

3 Problem Statement

In this paper, we explore how various components of the Guesser input (dialogue encoding method, visual input, and object category labels) influence the task success and the ability of the model to learn salient semantic features.

Following Greco et al. (2020), we assess the influence of various Encoder architectures on the GuessWhat?! task success and the importance of different turns in the dialogue history. Therefore, we train the relevant components of the GuessWhat?! model and evaluate the resulting performance singling out the influence of the Encoders. To do so, we fix the non-Encoder components, namely, we use the human questions and the answers from the dataset instead of QGen and Oracle, and assess the task success using the same Guesser architecture trained on the outputs of various tested Encoders. The Guesser architecture we use is the one proposed in the original GuessWhat?! paper (De Vries et al., 2017). The Encoders we test are described in detail in section 4.3.

Being interested in the ability of each model to identify salient information, we explore whether the models pay attention the most important dialogue turns to and whether permuting the order of turns impairs a model's performance. The particular tasks relating to this exploration include modifying the test sets. One modification is aimed at comparing the dialogues with and without the last turn (**no-last-turn**), which, according to Greco et al. (2020) and Testoni et al. (2021) contains a lot of task-relevant information. The drop in performance would, therefore, be indicative of the model's sensitivity to the salient information and the size of the drop could indicate the ability of the model to gather information from the remaining turns. The other experiment compares the performance on the unaltered dialogues to the performance on the **reversed** dialogues. Finally, we also compare the performance on the unaltered dialogues to the performance on a test set with randomly **shuffled** dialogue history. Both these modifications test the Guesser models' ability to identify the salient information independently of its dialogue position.

To assess the degree of the Guesser model's reliance on the category label we create a dataset with no category information (**no-cat**). We train all the models on it and compare their performance to their category-aware counterparts. As we are also interested in the interaction between the model's ability to identify salient information and its reliance on the category label, we create the permuted dialogue history test sets with category labels removed and carry out a comparison between the drops in the performance of the models trained with and without category information.

All dataset modifications are discussed in greater detail in the next section.

4 Datasets

The dataset we use as a test bed is a subset of GuessWhat?! dataset collected by De Vries et al. (2017). The dataset consists of real-world images taken from the MS-COCO dataset (Lin et al., 2014) along with dialogues collected via Amazon Mechanical Turk. In the dialogues, one person is assigned the role of the Guesser, having to select the target object from the ones on the image by posing yes/no questions to the other player, assigned the role of the Oracle. Following Greco et al. (2020), we filtered the dataset to only include the dialogues in which the Guesser successfully selected the target object and to only include the dialogues with 10 questions or less (90K training set, 18K both in validation & testing). For an in-depth analysis of the dataset features and their correlations, please, refer to Greco et al. (2020).

4.1 Dialogue History Permutations

There were three modified test sets created for the experiments. The first of the experimental datasets modified dialogue history, excluding the last turn of the dialogue, henceforth **no-last-turn**. The motivation for this is that the last question is often long and includes many details, thus containing a lot of task-relevant information, as people are also reliant on the information in the last turn, as shown in Testoni et al. (2021). The impact of the removal of the last turn on the Guesser accuracy could show the model's sensitivity to the importance of the last turn.

The second experimental dataset included all the question-answer pairs from the original dialogue but in **reversed** order. This modification of the dataset allows us to test how much influence the correct dialogue order has on the Guesser accuracy. If the reversed order task success proves comparable to the unaltered dataset, it would imply that the model is robust to the changes in turn order. Greco et al. (2020) claim that a good Encoder should be able to identify the salient information independently of the position in the dialogue history.

The third experimental dataset included the entire dialogue history as well, but in a randomly **shuffled** order. We decided to create this subset to expand the reversed history experiment and explore if the Transformer-based models would be robust to such changes, as the literature seems to suggest (Sinha et al., 2021).

4.2 Category Removal

To assess the degree to which the Guesser models rely on the category information for prediction, we experimented with removing this information. Specifically, we chose to uniformly replace all category labels and ids with the same dummy category "no_category". As the candidate object still had their unique ids, the effect of this replacement would reflect only the contribution of the candidate object category label to the task success.

We re-created the experimental test sets described above for the **no-cat** subset, resulting in three more experimental sets: **no-cat + no-last-turn**, **no-cat + reversed**, and **no-cat + shuffled**.

4.3 Models

The experiments we conduct feature 4 models, each consisting of two parts: the Encoder, generating the representation of the dialogue history and, in some cases, of the image; and the Guesser, selecting the target from a list of candidate object representations based on the Encoder output. Since we expanded the experiments by Greco et al. (2020), we used the code from their repository[2] with modifications and bug-fixes, that we documented in detail. We also proposed improvements to the code in the repository which we describe separately in section 7.

The Guesser architecture was the same across the models. The Guesser received as the input the category of each candidate object (either real, or dummy category), its' spatial coordinates, and a representation of the dialogue history with an optional representation of the image, both obtained from the Encoder. Additionally, LXMERT model received an object crop of the image for each candidate object. The candidate object information is then put through a Multi-Layer Perceptron (MLP). The MLP had three layers with 264, 512, and 512 nodes respectively. The embedding from the Encoder was dot-multiplied with the candidate object representations obtained from the MLP and put through softmax to obtain object probabilities.

As mentioned above, the Encoder models varied along two dimensions: input modality (blind vs multimodal) and base architecture (LSTM vs Transformer). We thus had four Encoder model architectures: LSTM, V-LSTM, RoBERTa, and LXMERT. All of the Encoder models featured a linear layer with *tanh* activation that scaled the output of the Encoder to the size of the Guesser input (512 nodes)[3].

The multimodal models also required image preprocessing, specifically, V-LSTM required the ResNet image features generation for all the datasets, and LXMERT required Faster R-CNN image features. Since LXMERT model received

[2] https://github.com/claudiogreco/aixia2021

[3] All the specific model hyperparameters are described in respective configuration files in the project repository at https://github.com/TamaraAtanasoska/dialogue-history/tree/main/model-repos/aixia2021/config/SL

the object crops of the image for each of the candidate objects, it also required the visual features for the object crops.

All the models were trained on a remote machine provided by the University of Potsdam, with 12G of GPU space (NVIDIA-SMI 510.54 with CUDA 11.6).

The next sections describe in detail the Encoder models used in the present paper.

4.3.1 Blind Models: LSTM

The LSTM Encoder architecture follows the one proposed in De Vries et al. (2017), encoding only the dialogue history using one layer of unidirectional Long-Short Term Memory units with 512 nodes. The Greco et al. (2020) repository featured a `PyTorch` version of the LSTM Encoder, and we chose to use this version for the experiments. The LSTM-based Guesser model was trained for 30 epochs.

4.3.2 Blind Models: RoBERTa

The RoBERTa Encoder uses BERT (Transformer) architecture which is modified to be more robustly optimized Liu et al. (2019). We used a pretrained RoBERTa model (`roberta-base` provided in `transformers` library), having 12 self-attention layers with 12 heads each. The model was trained for a masked language modeling task for 500K steps on English text. The RoBERTa-based Guesser model was trained for 30 epochs.

4.3.3 Multimodal Models: V-LSTM

The multimodal representation for the LSTM-based architecture was achieved by concatenating the linguistic and visual representation and scaling the result with the MLP of the Guesser. The visual representation was obtained using frozen ResNet-152 features. The files with the image features were generated with a dedicated script, resulting in 2048 visual features. The LSTM component consisted of one layer of unidirectional LSTM units with 1024 nodes. The V-LSTM model was trained for 30 epochs. This architecture is similar to the supervised learning version of the visually-grounded dialogue state encoder (GDSE) architecture proposed by Shekhar et al. (2018). In the code of the Greco et al. (2020) repository, there was an option to generate and train V-LSTM with object crop image features, but the model in the paper seems to have been trained without it, so we opted for a more precise replication.

4.3.4 Multimodal Models: LXMERT

The Transformer-based multimodal Encoder used LXMERT architecture Tan and Bansal (2019). This model represents the image features with position-aware object embeddings from the 36 most salient image regions identified using Faster R-CNN. The text is represented with position-aware word embeddings. Both modalities are processed by a Transformer-based architecture with self-attention (5 layers for visual and 9 for textual embedding). The two representations are combined using a multimodal transformer with cross-modal attention with 5 layers. The model was trained on a multi-objective multimodal task pool consisting of 5 tasks. The pretrained model used for the experiments was obtained from the link provided in the Greco et al. (2020) paper repository [4]. This model is pre-trained on five vision-and-language tasks whose images come from MS-COCO and Visual Genome (Lin et al., 2014; Krishna et al., 2017).

The LXMERT model required the generation of image features for Faster R-CNN as a preprocessing step both for the entire image and the object crops. However, the files required for the generation process were absent from the original repository, as they were too large for GitHub. Upon request, the authors of the Greco et al. (2020) paper provided the image features and we relied on them for model training.

4.4 Evaluation

We used the GuessWhat?! task success, that is the accuracy of the Guesser model on the human dialogues (and their modifications) as the evaluation metric. We interpret the drops in accuracy on modified datasets as indications of the contribution of the modified feature, and similarly, we interpret the drops in the accuracy of the models with modified inputs as indicative of the impact of the input feature of interest. We used the train-validation-test split provided by the authors of the original dataset (De Vries et al., 2017).

5 Experiments

The experiments we conducted are presented below in four sections. Each section includes experiment results of the models trained with (**cat**) and without (**no-cat**) candidate object category labels. Section 5.1 covers the results on the unaltered

[4] `https://github.com/claudiogreco/aixia2021/tree/main/lxmert`

dialogues, while sections 5.2, 5.3, and 5.4 cover the **no-last-turn**, **reversed**, and **shuffled** dialogue history permutation experiments respectively. Beside comparing the results of the models trained on **cat** and **no-cat** datasets, we compare our results with those reported by Greco et al. (2020), wherever applicable. The training process of the four category-aware Guesser models (**cat**) is shown in the appendix A.

5.1 Replication Results

Table 1 reports the performance of the different Encoder models in our experiment as compared to the results reported in Greco et al. (2020). It also shows the effect of category removal on the Guesser performance.

The results of the replication experiment, even though not matching the results reported by Greco et al. (2020) exactly, are very close to them (within 1%) for all models except LXMERT, which shows slightly better performance (by 6%). Importantly, the trends in the task success are generally preserved, that is, Transformer-based models outperform the LSTM-based models, which perform very similarly to each other. However, the LXMERT performs better in our experiment than RoBERTa, while in Greco et al. (2020) the reverse is true. As the authors provided the random seed value in the configuration, the slight differences found might stem either from the difference in batch size (we had to make batches smaller to adjust for our limited computational resources) or from the differences in the way the random seed is handled by the `PyTorch`, as any slight difference in that could result in a different performance due to the stochastic nature of the models.

As for the effect of category removal, all models but LXMERT when trained on **no-cat** dialogues when compared to the category-aware counterparts, show a very large drop in performance 16-18%, while LXMERT performs relatively well (a drop of 5.2%). This could be explained by the fact that LXMERT Encoder has extensive multimodal pretraining that could potentially compensate for object category removal.

5.2 Experiment 1: No Last Turn

The experiment on the influence of the last turn on task success required running the models on the modified test set and comparing the results with the ones on the unaltered test set. Table 2 reports the results of the different Encoders as reported in Greco et al. (2020) and in our experiment, as well as the results on the **no-last-turn** dataset produced by the **no-cat** Guesser models.

In the **no-last-turn + cat** results, RoBERTa performs the best and shows the least drop in performance, followed by LXMERT, while the accuracy and the drop in accuracy of LSTM and V-LSTM are very similar. For LSTM and V-LSTM, the performance appears quite similar between the original article by Greco et al. (2020) and our replication (with differences of 1.1% and -2.3% respectively), while in the performance of the Transformer-based models there is a significant difference (the replication accuracy being 7.3% better for RoBERTa and 6.8% better for LXMERT). Importantly, there are also differences in the size of the drop in performance (reported in parenthesis): while LSTM performance drops by the same 18% on the **no-last-turn** test set, for the multimodal V-LSTM and LXMERT there is a difference in the size of the drop between the replication and the original article (2.8% and 1.2% respectively), and for RoBERTa the drop is significantly more pronounced in the original paper (the difference in the size of the drop being 6.5%).

While the overall performance trends for this experiment are quite similar, we observed a closer performance of LSTM and V-LSTM, and, interestingly, we observed RoBERTa still outperform the other models by some margin (4.5%), unlike the findings of the original experiment. As the decrease in performance is the most pronounced in LXMERT, we could name it as the most sensitive to the removal of the last turn, while Greco et al. (2020) reports RoBERTa to be more sensitive to it than LXMERT. In our experiment, conversely, RoBERTa was the least sensitive to this modification.

Here it is worth noting that in addition to the differences in batch size and possible changes due to random initialization, we are comparing the average of the results reported for the games of lengths 3, 5, and 8, to the results averaged across all the game lengths. This could the source of discrepancy in the performance patterns. Unfortunately, the result averaged across all lengths on the test set with the last turn removed is not reported in the Greco et al. (2020) article.

As for the interaction between the category and the last turn removal, the differences in the performance of the different Encoders on the **no-last-**

	evaluation	LSTM	V-LSTM	RoBERTa	LXMERT
Greco et al. (2020)	validation	65.6 (19)	68.7 (7)	65.2 (9)	65.1 (12)
	test	64.7	64.5	67.9	64.7
	no-cat	-	55.2 (9.3)*	-	-
present	validation	64.4 (8)	64.9 (11)	68.3 (13)	71.6 (26)
	test	65.3	65.0	68.7	70.7
	no-cat	46.7 (18.6)	46.5 (18.5)	52.1 (16.6)	65.5 (5.2)

Table 1: Validation and test accuracy of the four Encoder models in our replication experiment compared to the results reported in Greco et al. (2020). The numbers in parentheses in the validation accuracy indicate the best epoch. The **no-cat** row reports the model performance when trained without category information. The numbers in parenthesis indicate the drop in performance brought on by category removal. * The information on the no-category model performance was obtained through personal communication with the authors on their work-in-progress.

		LSTM	V-LSTM	RoBERTa	LXMERT
Greco et al. (2020)	cat	46.2 (18.5)	49.8 (14.7)	44.7 (23.2)	43 (19.7)
present	cat	47.3 (18)	47.5 (17.5)	52.0 (16.7)	49.8 (20.9)
	no-cat	35.1 (11.6)	35.9 (10.6)	39.9 (12.2)	46.4 (19.1)

Table 2: Test accuracy of the four Encoder models on the dataset without the last turn (**no-last-turn**) as reported in Greco et al. (2020) and our replication experiment. The numbers in parenthesis indicate the drop in performance as compared to the performance on the unaltered dialogues. The **no-cat** row reports the model performance when trained without category information.

turn test set are *almost erased* for the **no-cat** models, the drop in performance being ~11% for all models except for LXMERT (~20%), which is the most sensitive to the last turn removal, being the least sensitive to the category removal. The performance of LXMERT trained with and without category as well as the drops in performance are very close.

The difference between the models trained with and without candidate object category labels is *less* pronounced in the **no-last-turn** test set: it is ~12% for all models but LXMERT, as compared with ~17% on the unaltered test set. The decrease is also smaller for the LXMERT, which shows a drop of only 3.4% (vs 5.2% on unaltered test set), once more being the most robust to the category removal.

5.3 Experiment 2: Reversed History

The experiment on the influence of the turn order inversion on GuessWhat?! task success required running the models on the **reversed** dialogue history test set and comparing the results with the ones on the unaltered test set. The results of the experiment both compared with Greco et al. (2020) and with **no-cat** models are reported in table 3.

Once again, RoBERTa is the best-performing model (67.2%) with the least drop in performance (1.5%), followed by LXMERT (65.9 accuracy and 5.7% drop) and V-LSTM (53.2 accuracy and 11.8% drop). LSTM in the replication experiment shows the worst performance (49.2%) and the largest drop in accuracy (16.1%) when the dialogue history is reversed, thus showing the least ability to identify salient information.

There are significant differences between the patterns observed in the original versus the replicated results: while RoBERTa performs very similarly in terms of absolute performance, the drop in performance, and compared to other models, the V-LSTM results are less similar (the difference being 1.9% in task success and -1.4% in performance drop), with LSTM showing the largest difference (6.8% in task success and 7.4% in performance drop). Once more, we observe 5.6% better LXMERT performance as compared with the results reported by Greco et al. (2020), while the drop in performance brought by dialogue history inversion is more similar (the difference being 1.3%).

It is not quite clear what could cause the differences in the results, besides the batch size and possible random initialization differences. One possible explanation for the large difference seen in the performance of the LSTM model is that we chose to use the `PyTorch` code in the newer, paper-specific repository, over the older code in the De Vries et al. (2017) repository.[5]

[5] In the README, the older repository was reported to

		LSTM	V-LSTM	RoBERTa	LXMERT
Greco et al. (2020)	cat	56 (8.7)	51.3 (13.2)	66.5 (1.4)	60.3 (4.4)
present	cat	49.2 (16.1)	53.2 (11.8)	67.2 (1.5)	65.9 (5.7)
	no-cat	39.3 (7.4)	40.7 (5.8)	50.3 (1.8)	59.6 (5.9)

Table 3: Test accuracy of the four Encoder models on the dataset with the **reversed** dialogue history as reported in Greco et al. (2020) and our replication experiment. The numbers in parenthesis indicate the drop in performance as compared to the performance on the unaltered dialogues. The **no-cat** row reports the model performance when trained without category information.

There is some interaction between the effects of category removal and dialogue history inversion. The drop in performance brought on by the reversal of the turns is much *less* pronounced in the LSTM-based models trained with no category information (the difference being -8.6% and -6% for LSTM and V-LSTM, respectively). However, the Transformer-based models trained on **no-cat** data show very similar decreases in performance on the **reversed** test set to the ones obtained for their category-aware counterparts.

The difference between **cat** and **no-cat** models is *less* significant on the **reversed** test set. RoBERTa is showing the largest drop of almost 17%, followed by V-LSTM (12.5%) and LSTM (9.9%), while LXMERT is once more the least affected by category removal (6.3%). The difference in sensitivity to category removal between the different Encoder models' is the strongest on the **reversed** test set.

5.4 Experiment 3: Shuffled History

The experiment on the influence of the turn order shuffling on the Guesser accuracy required running the models on the **shuffled** dialogue history test set and comparing the results with the ones on the unaltered test set. Table 4 reports the results of the shuffling experiment both for Guesser models trained with and without candidate object category labels.

Interestingly, shuffling brings *smaller* decreases in performance, than dialogue history reversal, though the pattern is still the same. LSTM showing the largest drop (16.1 **reversed** vs 8.2% **shuffled**), followed by V-LSTM (11.8 and 5.9% performance drop, respectively), then LXMERT (5.7 and 2.5%), RoBERTa being the best model (1.5 and 0.7% drops in performance on **reversed** and **shuffled**, respectively). This might come from the fact that shuffling preserves some of the order, which could be significant on shorter dialogues.

As for the **shuffled + no-cat** test set, the drops in performance from shuffling are uniformly small for all Encoder models (1 to 4%), with LSTM again experiencing the largest drop (3.7%), and RoBERTa - the smallest (1.2%). The effects of shuffling are, thus, significantly *less* pronounced on the models trained with no category data.

The difference between **cat** and **no-cat** performance is *slightly less* pronounced in the **shuffled** condition as compared to the unaltered dialogues but follows the same pattern: all models but LXMERT show a very large drop in performance (14 to 17%), while LXMERT is quite robust to the absence of category labels (a drop of 5.8%).

6 Discussion

For the baseline performance of the four Encoder models, we observed similar results to the ones reported in Greco et al. (2020), Transformer-based models showing better Guesser accuracy than the LSTM-based ones. Unlike Greco et al. (2020), we observe the best performing model on the unaltered dialogues to be LXMERT, not RoBERTa. The difference between the Transformer and LSTM-based models is larger than that between multimodal and blind models, which could be due to the fact that we use pretrained Encoders for Transformer-based models while the LSTM-based ones are trained on the GuessWhat?! data only.

The experiments on altering the dialogue history show that the Transformer-based Encoder models are more robust to history permutations than the LSTM-based Encoder models. RoBERTa outperforms LXMERT on the **reversed** history test set as well as on the **no-last-turn** one. LXMERT proves the most sensitive to the last turn removal, while still performing quite well in terms of absolute accuracy. Generally, we observe closer performance

have some bugs, which, if the authors used it, could have affected the LSTM results. However, in this case, it is not clear why no large performance differences were observed in the other experiments.

	LSTM	V-LSTM	RoBERTa	LXMERT
shuffled	57.1 (8.2)	59.1 (5.9)	68.0 (0.7)	68.2 (2.5)
shuffled + no-cat	43.0 (3.7)	43.6 (2.9)	50.9 (1.2)	62.4 (3.1)

Table 4: Test accuracy of the four Encoder models on the dataset with the **shuffled** dialogue history. The numbers in parenthesis indicate the drop in performance as compared to the performance on the unaltered dialogues. The **no-cat** row reports the model performance when trained without category information.

of RoBERTa and LXMERT than that reported in Greco et al. (2020). Surprisingly, shuffling the dialogue history brings much smaller drops in performance, than dialogue history inversion. This could be due to the fact that some of the dialogues are quite short and shuffling preserves more of the dialogue order than inversion. Generally, our observation that Transformer-based models outperform LSTM-based ones in dialogue permutation experiments is in line with the recent findings on Transformers being robust to such permutations (Sinha et al., 2020; Pham et al., 2020; Gupta et al., 2021; Sinha et al., 2021). We could draw from this that Transformer-based models are better able to identify salient features independent of their position in the dialogue. However, this could also be due to the pretraining of Transformer-based models. Exploring the attention of these models in the way proposed by Greco et al. (2020) could more reliably confirm or refute this conclusion. Exploring the performance of Transformer-based models trained on GuessWhat?! data exclusively could also clarify this point. We leave this to future research.

As for the degree to which the Guesser models are reliant on the candidate object category labels for the Guesser prediction, we find it to be very high (16-18% on unaltered dialogue history), which is in line with the studies that claim the models rely on this information rather than multimodal integration of other training data (Strub et al., 2018; Zhuang et al., 2018; Suglia et al., 2020a,b). This is true of all the models but LXMERT, which is significantly less dependent upon category labels, as it shows a much smaller drop in performance when they are removed (5.2%). The difference between **cat** and **no-cat** models is less pronounced in the history permutation experiments. The average performance drop 14% for unaltered dialogues, 9.8, 11.4, and 13.1% for **no-last-turn**, **reversed**, and **shuffled** test sets, respectively. The differences in the performance drop brought on by category removal between the different Encoder models are the largest on the **reversed** dialogue history experiment.

The performance drops brought on by history permutations are similarly less pronounced in the **no-cat** models. The average performance drop for **no-last-turn** was 18.3% in **cat** and 15.9% in the **no-cat** condition, 8.8 vs 5.2% on the **reversed** test set, and 4.3 vs 2.7% on the **shuffled** test set. The differences in the size of performance drop brought on by history permutations between the different Encoder models are also smaller in the **no-cat** experiment, which could be partially explained by the generally lower performance of the models trained on **no-cat** data. The differences in the absolute Guesser accuracy between the Encoders are higher in the **no-cat** condition on the unaltered dialogue history, as well as both **revrsed** and **shuffled** test sets while being approximately the same for the **no-last-turn** condition. The differences between the absolute accuracy of the Encoder models in the permuted experiments are larger if the object categories are unavailable, while the sizes of the decrease in performance are smaller and so are the differences in the size of the decrease between the Encoder models.

Summing up, Transformer-based models seem to be more robust to dialogue history permutations. The model architecture or pretraining seems to have a larger effect on the task success than the incorporation of visual features, as the difference between Transformer- vs LSTM-based models is larger than that between multimodal and blind models, which could be the result of pretraining. Even though RoBERTa is slightly better than LXMERT at identifying salient information, it is, unlike LXMERT, strongly reliant on the object category labels, as it is a blind model. LXMERT is the only model that is relatively robust to category removal, unlike the other three models which largely rely on the candidate category labels for Guesser predictions. This, as well, could be the result of multimodal pretraining of this particular Encoder model.

7 Code-base Contributions

We introduced code improvements in both repositories we used for our experiments.

In the original GuessWhat?! repository[6] we have fixed the code inconsistencies around the updated dataset and upgraded the project documentation by providing an environment setup to run the code as it is in 2022. Together with some other minor changes, we packaged and added these improvements back to the original repository, making us official contributors. We hope that these improvements help the next generations of researchers.

The newer Aixia2021 repository[7] was our main experiment repository. We changed a lot of the original files to suit our specific requirements, partially removing the unused files and introducing novel computation methods. Taking our limited computational resources into consideration, we worked on optimizing and improving the performance of the models, which resulted in 30% faster training for the V-LSTM model, and some minor 1-2 minute improvements for the Transformer-based models. We think that the improvements to the Transformer-based models would be much more apparent if we had more powerful hardware to test on as those models were already using close to 100% of the GPU, to begin with. Additionally, we removed all deprecation errors and unsafe computation warnings that were present in the original repository. All the models currently train error-free. Some of the more significant changes that make the entire train-test cycle easier include creating the image features for the respective datasets as part of the training cycle instead of it being a separate preprocessing step, as well as automatically saving the best epoch and optionally triggering testing on it. For a more detailed list of these improvements, please look at the GitHub pull request history here[8].

Although we contributed a lot of code changes, we consider the clear and detailed documentation the biggest contribution to both of the repositories we worked on, as well as guaranteeing error-free runs if one follows it.

8 Conclusions

In this paper, we explored the interaction between the ability of different GuessWhat?! Guesser models to identify salient information in the dialogues and the degree of model reliance on various aspects of the model input. Following (Greco et al., 2020), we compared four models, namely, LSTM, V-LSTM, RoBERTa, and LXMERT. We ran a series of history permutation experiments and found a degree of interaction between the reliance on the category label and the performance on permuted data, which could, however, be partially explained by the lower performance of the models trained with no category labels.

We were able to generally reproduce the history permutation experiments, although we obtained better LXMERT performance than that reported in (Greco et al., 2020). Extending the history permutation experiments to dialogue history shuffling confirms the general finding that Transformer-based models seem to be more able to identify salient information independently of its position. We found RoBERTa to be most robust to history permutations, closely followed by LXMERT, which, unlike RoBERTa, was also robust to candidate object category removal.

As suggested in the literature, the GuessWhat?! Guesser models seem to be over-reliant on the category labels (Strub et al., 2018; Zhuang et al., 2018; Suglia et al., 2020a,b). However, LXMERT, unlike other models, is quite robust to category removal, which is likely explained by the extensive multimodal pretraining that allows the model to 'guess' the information that the other models can only get from the category label.

We note that the difference between the LSTM-based and the Transformer-based Encoders that we observed throughout our experiments could stem from pretraining, rather than purely architectural superiority. This could mask the importance of the visual input. Unfortunately, limitations on computational resources forced us to leave the experiments with training the Transformer-based Encoders from scratch to further research.

Acknowledgements

We would like to express our gratitude to the people who made this project possible. Firstly, we would like to thank Alberto Testoni for all the helpful communications concerning both the theoretical questions and the code that we needed to run to

[6] https://github.com/GuessWhatGame/guesswhat

[7] https://github.com/claudiogreco/aixia2021

[8] https://github.com/TamaraAtanasoska/dialogue-history/pulls?q=is%3Apr+is%3Aclosed

replicate the experiments, as well as his co-authors, Claudio Greco and Raffaella Bernardi. We would also like to thank Ravi Shekhar for his comments on the GDSE repository. Secondly, we would like to thank the MLCog Support team, and Philipp Sadler in particular, for helping us with GPU management. Finally, we would like to thank Explosion for their willingness to provide GPU resources.

References

Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. 2017. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2951–2960.

Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5503–5512.

Claudio Greco, Alberto Testoni, and Raffaella Bernardi. 2020. Which turn do neural models exploit the most to solve guesswhat? diving into the dialogue history encoding in transformers and lstms. In *NL4AI@ AI* IA*, pages 29–43.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. Bert & family eat word salad: Experiments with text understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12946–12954.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Aditya Mogadala, Marimuthu Kalimuthu, and Dietrich Klakow. 2021. Trends in integration of vision and language research: A survey of tasks, datasets, and methods. *Journal of Artificial Intelligence Research*, 71:1183–1317.

Thang M Pham, Trung Bui, Long Mai, and Anh Nguyen. 2020. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? *arXiv preprint arXiv:2012.15180*.

Chinnadhurai Sankar, Sandeep Subramanian, Christopher Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. *arXiv preprint arXiv:1906.01603*.

Ravi Shekhar, Aashish Venkatesh, Tim Baumgärtner, Elia Bruni, Barbara Plank, Raffaella Bernardi, and Raquel Fernández. 2018. Beyond task success: A closer look at jointly learning to see, ask, and guesswhat. *arXiv preprint arXiv:1809.03408*.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *arXiv preprint arXiv:2104.06644*.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2020. Unnatural language inference. *arXiv preprint arXiv:2101.00010*.

Florian Strub, Mathieu Seurin, Ethan Perez, Harm De Vries, Jérémie Mary, Philippe Preux, and Aaron CourvilleOlivier Pietquin. 2018. Visual reasoning with multi-hop feature modulation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–800.

Alessandro Suglia, Ioannis Konstas, Andrea Vanzo, Emanuele Bastianelli, Desmond Elliott, Stella Frank, and Oliver Lemon. 2020a. Compguesswhat?!: A multi-task evaluation framework for grounded language learning. *arXiv preprint arXiv:2006.02174*.

Alessandro Suglia, Antonio Vergari, Ioannis Konstas, Yonatan Bisk, Emanuele Bastianelli, Andrea Vanzo, and Oliver Lemon. 2020b. Imagining grounded conceptual representations from perceptual information in situated guessing games. *arXiv preprint arXiv:2011.02917*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

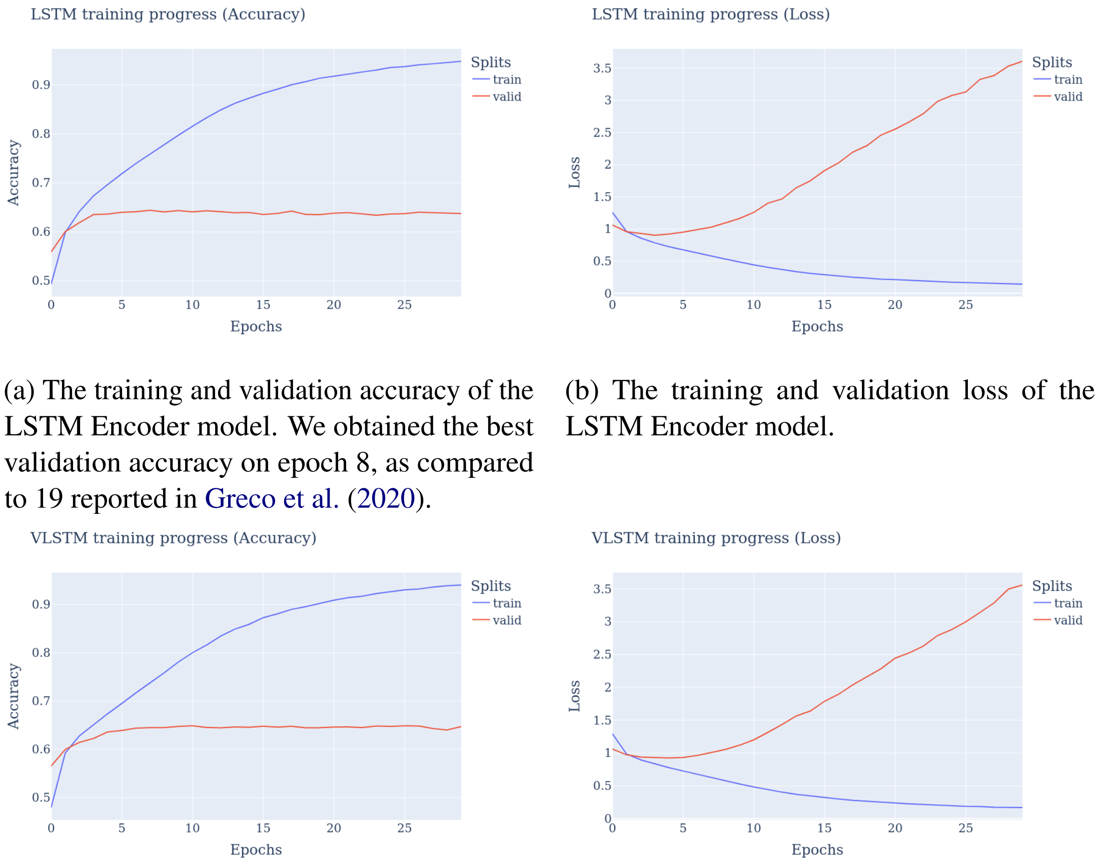
Alberto Testoni, Claudio Greco, and Raffaella Bernardi. 2021. Artificial intelligence models do not ground negation, humans do. guesswhat?! dialogues as a case study. *Frontiers in big Data*, 4.

Alberto Testoni, Ravi Shekhar, Raquel Fernández, and Raffaella Bernardi. 2019. The devil is in the detail: A magnifying glass for the guesswhich visual dialogue game. In *Proceedings of the 23rd SemDial workshop on the semantics and pragmatics of dialogue (LondonLogue)*, pages 15–24.

Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton Van Den Hengel. 2018. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4252–4261.

A Supplementary Graphs

This appendix includes the plots of the training progress of the models obtained with Weights & Biases (`wandb`).

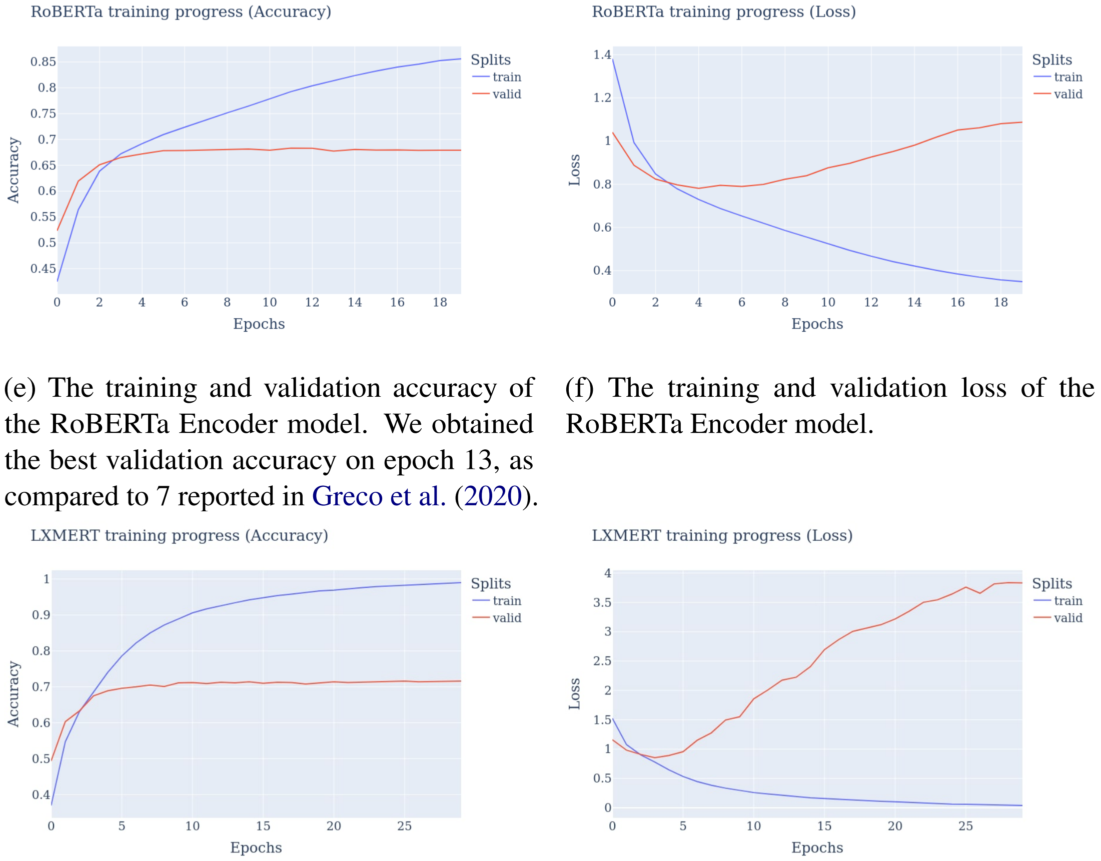


(a) The training and validation accuracy of the LSTM Encoder model. We obtained the best validation accuracy on epoch 8, as compared to 19 reported in Greco et al. (2020).

(b) The training and validation loss of the LSTM Encoder model.

(c) The training and validation accuracy of the V-LSTM Encoder model. We obtained the best validation accuracy on epoch 11, as compared to 9 reported in Greco et al. (2020).

(d) The training and validation loss of the V-LSTM Encoder model.



(e) The training and validation accuracy of the RoBERTa Encoder model. We obtained the best validation accuracy on epoch 13, as compared to 7 reported in Greco et al. (2020).

(f) The training and validation loss of the RoBERTa Encoder model.

(g) The training and validation accuracy of the LXMERT Encoder model. We obtained the best validation accuracy on epoch 26, as compared to 12 reported in Greco et al. (2020).

(h) The training and validation loss of the LXMERT Encoder model.