Modeling Referential Gaze in Task-oriented Settings of Varying Referential Complexity

Özge Alaçam[1], Eugen Ruppert[2], Ganeshan Malhotra[2]*
Chris Biemann[2] and **Sina Zarrieß[1]**
[1]Computational Linguistics, University of Bielefeld, Bielefeld, Germany
[2]Language Technology, University of Hamburg, Hamburg, Germany
{oezge.alacam, sina.zarriess} @uni-bielefeld.de,
{ganeshan.malhotra, eugen.ruppert, chris.biemann} @uni-hamburg.de

Abstract

Referential gaze is a fundamental phenomenon for psycholinguistics and human-human communication. However, modeling referential gaze for real-world scenarios, e.g. for task-oriented communication, is lacking the well-deserved attention from the NLP community. In this paper, we address this challenging issue by proposing a novel multimodal NLP task; namely predicting when the gaze is referential. We further investigate how to model referential gaze and transfer gaze features to adapt to unseen situated settings that target different referential complexities than the training environment. We train (i) a sequential attention-based LSTM model and (ii) a multivariate transformer encoder architecture to predict whether the gaze is on a referent object. The models are evaluated on the three complexity datasets. The results indicate that the gaze features can be transferred not only among various similar tasks and scenes but also across various complexity levels. Taking the referential complexity of a scene into account is important for successful target prediction using gaze parameters especially when there is not much data for fine-tuning.

1 Introduction

In a situated interaction, interlocutors produce and interpret complex communicative signals and intertwine their verbal utterances with non-verbal signals like gaze. For instance, when referring to objects in the visual environment, speakers tend to fixate the target referent, listeners gaze at the objects they believe to be referred to by the speaker and, importantly, listeners monitor the speaker's gaze in case it provides reliable information about the referent (Staudte and Crocker, 2011b). In noisy environments, listeners (that need to resolve references to objects) might even face situations where

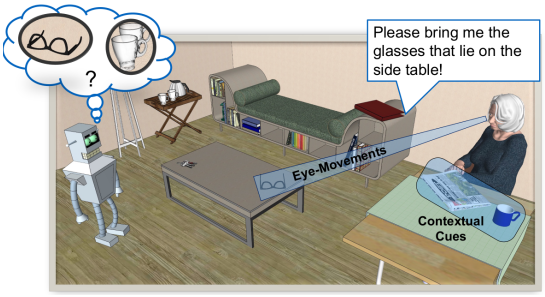


Figure 1: An example task-oriented scenario

gaze provides a more reliable cue than spoken words. For example, in the scenario depicted in Figure 1, "the glasses" in the command illustrates an ambiguous reference and during human-human communication, such ambiguities can be easily resolved via tracking referential gaze.

Referential gaze as a communication modality is a well researched / fundamental phenomenon in Psycholinguistics and Cognitive Science (Koller et al., 2012; Staudte and Crocker, 2011a; Prasov and Chai, 2008). Mainstream Natural Language Processing (NLP) systems — on the other hand — still usually employ language-only approaches, where the performance is highly dependent on the completeness of the language modality. Despite the fact that reference resolution in visual environments has become a very popular task in recent NLP and Computer Vision research (Kazemzadeh et al., 2014; Schlangen et al., 2016; De Vries et al., 2017; Cirik et al., 2018; Yu et al., 2018; Kalpakchi and Boye, 2019; Chen et al., 2020), there is very little work on reference resolution models that exploit eye gaze for this task. They fall short on modeling referential gaze for realistic scenarios for task-oriented communication that go beyond simple settings. One main reason behind this is human gaze's intricate nature of being complex (a multivariate sequence) and multi-functional (e.g. referential gaze, social gaze and so on) (Somashekarappa et al., 2020). In this paper, we propose a novel task of predicting when the gaze is referential during

* Remote research-intern at the Language Technology Group, University of Hamburg

Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022, pages 197–210
November 20–23, 2022.

the communication, aiming at modeling referential gaze for various multimodal settings.

Most recently, daily devices like laptops start to utilize eye-tracking technology (Brousseau et al., 2020; Rogers, 2019; Khamis et al., 2018). As a result, incorporating eye-movements in language comprehension models is an inevitable goal for NLP emerging from these developments, and this motivates systematic research on the interaction of different communicative modalities. However, the collection and pre-processing of eye-movement data is a very costly process, and this is another main reason why there are only a few large eye-movement datasets available (Alaçam et al., 2020; Wilming et al., 2017; Ehinger et al., 2009).

Eye-movements are highly influenced by bottom-up perceptual and top-down conceptual properties of the task (e.g. free viewing, search, etc.) and the properties of the visual environment (Einhäuser et al., 2008; Zelinsky et al., 2006; Henderson, 2003). Besides, their patterns (pupil size, saccade velocity, fixation duration, etc.) are very user-dependent (Rayner et al., 2007). All these factors introduce challenges in (i) learning meaningful patterns from limited data, (ii) generalizing well enough to different kinds of situations of real-world complexities and (iii) successfully incorporating it to NLP systems for reference resolution and meaning recovery. To mitigate these problems, transfer learning can be used to adapt the knowledge obtained from one setting to another, benefiting from its added generalization capabilities.

2 Background

In this paper, we apply transformer-based time-series modeling and transfer learning to the phenomenon of referential gaze. Section 2.1 discusses the background for technical modeling, and Section 2.2 introduces referential gaze.

2.1 Transfer Learning and Time-series Multivariate Classification

Time-series analysis have been generally approached using more traditional machine learning techniques such as XGboost (Chen and Guestrin, 2016), and Dynamic Time Wrapping (Lei et al., 2019). There has been also successful recurrent models like RNNs (LSTMs and GRUs) with additional enhancements to address the intricacies of multivariate time series (Wu et al., 2020; Bianchi et al., 2019). By taking the close relation of the referential gaze with language, LSTM solutions are considered as an adequate baseline for the task.

With the development of the auto-encoder architectures (Vaswani et al., 2017), most machine learning domains are dominated by transformer solutions. Transformer models for uni-variate time-series forecasting and classification has been studied broadly. However, as eye-trackers can record multiple parameters simultaneously (such as velocity, acceleration, pupil size, etc.), this makes the collected data a multivariate time series. Despite the simultaneity, many of these features might have their unique onsets and offsets in regards to changes in the top-down (*mental, cognitive*) or bottom-up (*perceptual*) factors. Thus, modeling referential gaze and classification based on a set of various raw gaze features requires a multivariate approach, which has recently received some attention in the literature.

Liu et al. (2021)'s simple but effective solution of combining a gating mechanism with transformer architectures seems to provide state-of-the-art results for time-series forecasting. A novel approach on supervised and unsupervised representation learning for a series of multivariate tasks (such as regression, classification and forecasting) has been proposed by Zerveas et al. (2021). Pretraining and fine-tuning procedures exhibit high resemblance to language modeling, but they are modified to process multivariate time series. The model only uses an encoder part, this provides great computational power. Their unsupervised pre-training scheme, evaluated on several benchmark datasets, surpasses the performance of all current state-of-the-art supervised methods including their own.

Moreover, transformer architectures can extract patterns from low-level features without extensive feature engineering because of their multi-layer structure and effective attention mechanisms. This might have particular advantages for eye-movement processing since many approaches uses fixation-based parameters where a series of rule-based assumptions are needed to define a fixation. And each researcher and each eye-tracking device producer might come up with their own criteria. Being able to work on low-level features might eliminate these inconsistencies.

2.2 Referential Gaze

Prior research indicates that incorporating eye movements of a speaker or a listener improves the

performance of many NLP tasks, e.g. in predicting / resolving which entity is being referred to in a complex visual environment (Mitev et al., 2018; Koleva et al., 2015). As shown by Koleva et al. (2015), listener gaze can be highly beneficial to predict which entity is being referred to in the sentence and to understand the intention of the listener when the targets and their referentially possible competitors are located close-by. A gaze-contingent system may react to changes in its environment by tracking the probability of the fixations per each item in the scene over time. However, Henderson et al. (2009) point out that the success of such a system is dependent on utilizing an effective combination of several fixation parameters. A study by Klerke and Plank (2019) indicates that globally-aggregated measures can also capture the central tendency or variability of gaze data instead of customizing towards individual participants.

Only a few studies embed a set of eye-movements (e.g. velocity, acceleration, pupil size) into a rich vector space (Sood et al., 2020; Takmaz et al., 2020; Park et al., 2019; Karessli et al., 2017). Nevertheless, those models are limited to relatively simple scenes or reading activities (e.g. CMCL Shared Task 2021-2022 (Hollenstein et al., 2021; Barrett and Hollenstein, 2020). Situated language understanding in a referentially complex environment imposes a different level of challenge as it requires more complex visual search due to ambiguity resolution among possible options.

3 Approach

We investigate the modeling of eye-movements and ask whether different referential complexities need individual referential gaze models or whether we can use transfer learning (pre-training on larger collections and fine-tuning on task-specific dataset). We build a model that predicts when the gaze of a participant is referential, i.e., when she looks at the target object referred to by the speaker. For a low-complexity scene (i.e., with few objects) and an unambiguous verbal description, this task can be considered as straightforward, since the user will quickly identify the target and not have to visually search for it. In a complex visual scene — with occluded objects and complex or ambiguous verbal descriptions — eye-movements can provide highly distinctive information to resolve ambiguities, but may also show more complex and challenging gaze patterns in return. Therefore, in this study, we minimize the contribution of accompanying linguistic and contextual information and focus on the influence and capabilities of gaze features.

3.1 Task

We frame the learning problem as a supervised sequence tagging task where the input is a multivariate time series (i.e., with multiple eye-movement parameters) and the output is a sequence of binary labels. The label indicates whether the participant's gaze is currently referential. Thus, we train our model to predict for each time frame whether the gaze of the participant is on the target object while the spoken sentence unfolds.

Given that verbal descriptions of referents vary in their complexity, different labeling schemes for "target objects" can be adopted. To illustrate, the second referring expression in Table 1 has a single *global target*, i.e., *cage_1*, as the object of the intended action. But, the expression mentions further referents (*table_1* and *man_1*, see Figure 2a) which are *local targets* that are likely to be gazed at as well. To account for this, we distinguish two different task settings: (i) in *Task-A*, we consider time frames as referential, where the gaze is on the global target; in (i) *Task-B*, we label all time frames as referential where the gaze is on a global or local target object.

3.2 Referential Complexity

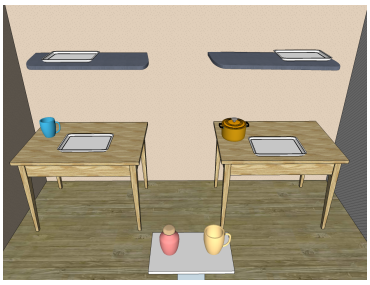
Referential complexity is a complex notion in itself and has been investigated in different fields and with different terminologies, cf. Clarke et al. (2013). In this study, we use the complexity classification provided by Alaçam et al. (2020)'s Eye4Ref Benchmark to account for reproducibility. Thus, we investigate three complexity levels — LOW, MEDIUM and HIGH — which differ in the way the scene and descriptions are composed. Sample stimuli and the basic descriptive statistics of each complexity level are given in Figure 2. In the LOW referential complexity, the focus lies on identifying the target and the targeted location with no ambiguity. In the HIGH and MEDIUM conditions, for each mentioned object in the scene, there are also distractor objects that share properties with the targets (e.g. type or color). Unlike other two, the HIGH condition contains not only objects but also people and actions.

Table 1: Sample (translated) sentences with varying complexities *(Experiment language is German)*

Complexity	Sentence
High	It is a book on a couch that he reads quietly.
High	It is a cage on a table that he moves. (Figure 2a)
Medium	Bring me the blue mug from the counter. (Figure 2b)
Medium	Bring me the mug, the blue mug, from the counter. (Figure 2b)
Low	Put the mug on the counter, on the blue one.
Low	Put the mug on the counter, next to the blue one. (Figure 2c)

(a) Set-1: HIGH RC: 28 participants, 36 scenes, 548K timestamps

(b) Set-2: MEDIUM RC: 27 participants, 46 scenes, 565K timestamps



(c) Set-3: LOW RC: 21 participants, 17 scenes, 290K timestamps

Figure 2: Sample scenes and descriptive statistics for the three referential complexity (RC) Eye4Ref datasets

4 Experiments

4.1 Data

We use the Eye4Ref Benchmark that consists of three datasets (Alaçam et al., 2020) where gaze data was collected from human participants on referentially complex situated settings using a *SR Eyelink 1000 Plus* eye tracker with a sampling rate of 1000 Hz. For all datasets, the eye-tracking data of the participants were recorded while they are presented with images and accompanying spoken descriptions like (*put object X onto Y*) and descriptions like (*there is an object X that Y interacts with*), as summarized in Table 1. The language of the experiments is German. For simplicity, in our illustrations, we use the translated sentences from the dataset. Referential complexity of the studies is defined in terms of visual manipulations (e.g. number of objects, visibility of the target items, presence of distractor objects that share the same class with the target objects) and linguistic ones (e.g. the position of the disambiguating word in a sentence). For the details of the dataset and data collection, please refer to (Alaçam et al., 2020).

4.2 Gaze Feature Vector and Labels

Employing a simple approach which uses only one selected gaze parameter (e.g. gaze location at one point of time) may yield successful results only if the number of objects is limited (low referential complexity). Furthermore, a lot of assumptions need to be made to decide when the aggregated group of eye-movements forms a fixation or saccade. Thus, regarding the goals of the project addressing various complexities, an elaborated parameter selection is required to establish crossmodal mapping. We use a time-series format that requires fewer assumptions on the raw data. For computational efficiency reasons, we use binning, where each bin corresponds to a cumulative sampling for 20 ms such as average fixation duration, gaze velocity, or list of targeted area of interest (AOIs). Eye4Ref provides pre-processed data for each scene and participant in each dataset. For each timestamp (20 ms bin), all linguistic, contextual and gaze features are provided in a CSV format. The number of the features (on average 230 values) is dependent on the number of items in the scene. Forty-five of them correspond to participant- and study-related information as well as the set of eye-movement parameters. Approximately 180 values correspond to one-hot encoded fixation location parameters addressing all the objects in the respective scene, indicating whether the gaze is fixated on that object. For our purposes, we have reduced the size of this scene-specific vector part to two scene-agnostic binary output variables: whether the gaze is (i) on the target object or (ii) on a communicatively relevant object (all referents). The dimension of the final fixed-sized feature vector is 16, consisting of only gaze and scene information such as gaze acceleration, velocity, pupil diameter,

object count of the scene as a general referential complexity measure, etc (see Appendix A.1). In order to be able to generalize better, gaze coordinates of the eye-movements are not included in the training since this information would be only useful in static images, where the objects have a fixed location.

4.2.1 Normalization Parameters

One of the manipulated variables in this study is the scope of normalization for the eye-movement parameters. We normalized the continuous scale gaze features in three ways: (i) within participant (across items), (ii) within dataset (across participants and items), and (iii) across all datasets. These parameters are directly retrieved from the original dataset. Since eye-movements are highly task and participant dependent, one common approach is to train models for each user and each task, which is a big challenge for incorporating eye-movements. On the other hand, with the advancements deep learning methods, this problem can be overcome through transfer learning and pre-training. This experimentation allows us to investigate to what extent a normalization scope should be extended for a successful transfer.

4.3 Splits and Testing Conditions

Each complexity set has been split item-wise into training (80 %), validation (10 %) and test (10 %) sets. This means that each set has distinct items in their repertoire. To investigate the effect of size and diversity of training data, we introduce the COMBINED condition, where the new sets are created by concatenating the respective subsets of all conditions. In the end, we obtain 16 train–test combinations (Appendix A.5).

- Within-complexity tests: training and testing on the same complexity e.g. $\text{Train}_{\text{LOW}} \rightarrow \text{Test}_{\text{LOW}}$
- Data-diversity tests: training on COMBINED and testing on each complexity condition e.g. $\text{Train}_{\text{COMBINED}} \rightarrow \text{Test}_{\text{LOW}}$
- Cross-complexity tests: training and testing in a cross-complexity way e.g. $\text{Train}_{\text{LOW}} \rightarrow \text{Test}_{\text{MEDIUM}}$

5 Model Architectures

To establish the performance of within- and cross-complexity performances, we employ two deep learning approaches; (i) LSTM as a sequential base model and (ii) transformer architecture (Vaswani et al., 2017). We use a transfer learning approach to establish the compatibility of different complexities. This is done because there are not many large datasets available and we want to study options of how an available benchmark (Eye4Ref) can be utilized as a baseline that is then adapted on a small set of individual, task-specific data. Transfer learning only trains the final layer (*the output layer* and all dense layers are frozen), thus the input representation stays the same. Therefore, we further experiment with fine-tuning the layers to arrive at an input encoding that better fits the small target data. The full code, and model summaries are provided under supplementary material.

Baseline LSTM Model We experimented with two variations of a bi-directional LSTM architecture (Hochreiter and Schmidhuber, 1997). Since we are dealing with a sequence classification task, attention mechanisms can help to improve the performance of our model by guiding the model to give more weight to the relevant time-frames in the sequence. In the second variation, we use a variant of self-attention (Bahdanau et al., 2015) known as the Sequential Self Attention by Keras. The details of the models are provided in Appendix A.3.

Time-series Transformer Model (TST) Inspired by Zerveas et al. (2021), we utilize their working solution (TST for classification) as our Transformer architecture[1]. For the sake of systematicity, the scope of this paper is restricted to supervised pretraining and further fine-tuning, by leaving unsupervised pretraining to future studies.

For input, we create sequences of 25 timestamps, spanning 500 ms of input data. We use class weights to treat the imbalance in the size of the datasets. If the model predicts a referential gaze for a timestep sequence, then the most visited area-of-interest during that period is accessed and compared against the true label. The final representation vectors corresponding to all time steps are concatenated into a single vector (an input vector). For the classification problem, the predictions are passed through a softmax function to obtain a distribution over classes, and its cross-entropy with the categorical ground truth labels will be the sample loss.

[1]For the details, please visit the original paper. The modified code is available at https://gitlab.com/alacam/referential-gaze-modeling

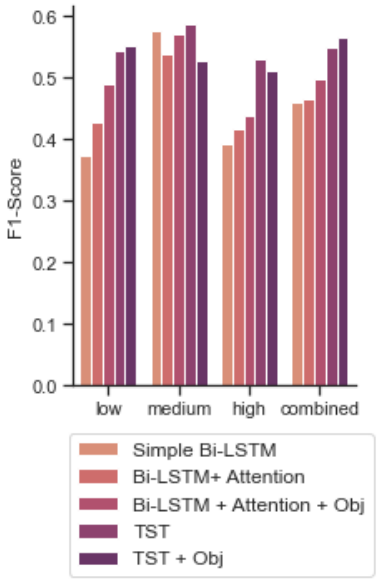


Figure 3: F1-Scores for various model variations on the MEDIUM complexity model

Each training sample, which is a multivariate time series of varying length l and 18 different variables, constitutes a sequence of l feature vectors such as $[x1, x2, ..., xl]$. The original feature vectors are linearly projected onto a 50- (for LSTM) and 64-dimensional vector space, (for TST) where d is the dimension of the model.

The first setting (*Supervised Pretraining*) is a simple use of pretrained models after training the models on the training sets via supervised learning. The parameters are provided in Appendix A.3 and runtime details in Appendix A.4. In the second variation (+*fine-tuned*), we do fine-tuning on the pre-trained model by further training with a lower learning rate on the validation set for 20 epochs. With this, we aim to improve the results by incrementally adapting the pre-trained gaze features to new data.

6 Results

6.1 Model Variations

Before looking into transfer learning, we test for appropriate model architectures for learning gaze parameters. We choose MEDIUM condition to compare the variations; (i) simple bi-LSTM , (ii) attention-based LSTM, (iii) attention-based LSTM with object count parameter, (iv) TST (time series transformer) without object count and (v) the previous condition with object count. As shown in Figure 3, on MEDIUM condition, incorporating attention mechanism is crucial for LSTM architecture. In addition, including the number of objects in the scene as a complexity feature boosts the performance. When the object count (as an indicator of referential complexity) is excluded from the feature vector during training with the LSTM model, the F1-Scores drops on average by .06, while both COMBINED and MEDIUM benefit from this parameter. TST variations beats the LSTM models in all conditions. Yet, unlike the LSTM model, including object count only benefits the low and combined condition with small margin but impairs the medium and high condition.

6.2 Normalization Parameters

Figure 4 shows each TST model's performance on the three normalization scopes. Within-participant normalization is the most simple approach where each parameter collected within a trial are "min-max" normalized producing values between 0 and 1. For within-study normalization (WS), "min-max" normalization is applied by taking all the trials collected for each study separately. Across-study (AS) normalization is the most comprehensive approach since all gaze parameters are normalized by taking all produced values for that parameter in the entire benchmark. WP normalization produces comparable scores to more global approaches. Using more sophisticated methods seems to be beneficial especially for fine-tuning and the long-tail conditions such as LOW and HIGH. These results indicate that if the training size is limited or has different referential complexity than the target set, applying more global way of normalization might be preferred.

6.3 Within-Complexity Results

On target referent prediction (Task-A), the negative class has a proportion between 87 and 92%, rendering the task of identifying the sparser positive class somewhat difficult. When we take all referents into account (Task-B), the most frequent negative class has a share between 68 and 75%. All within-complexity test results beat their (most frequent class) baseline on the accuracy metric ($\text{Low}_{\text{baseline}}$: 0.683, $\text{Medium}_{\text{baseline}}$: 0.755, $\text{High}_{\text{baseline}}$: 0.728, $\text{Combined}_{\text{baseline}}$: 0.73), indicating that even with gaze information alone, communicative object prediction is possible.

Within-complexity results are provided in Figure 5 (details in Appendix A.6). Since further fine-tuning does not make sense for the within complexity conditions, fine-tuning values are marked

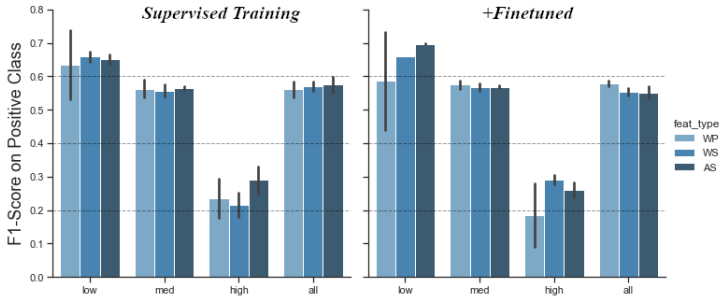
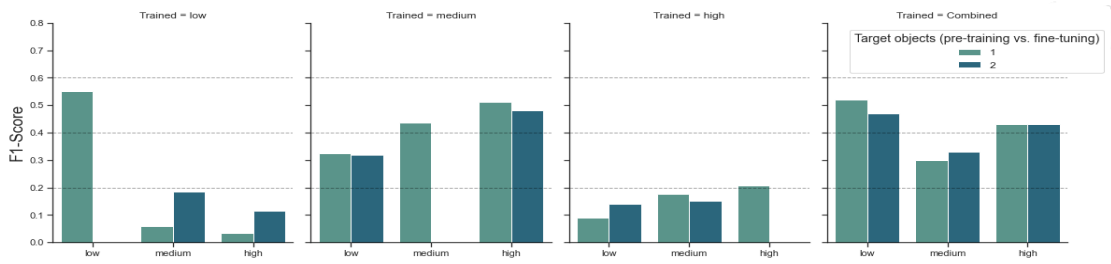
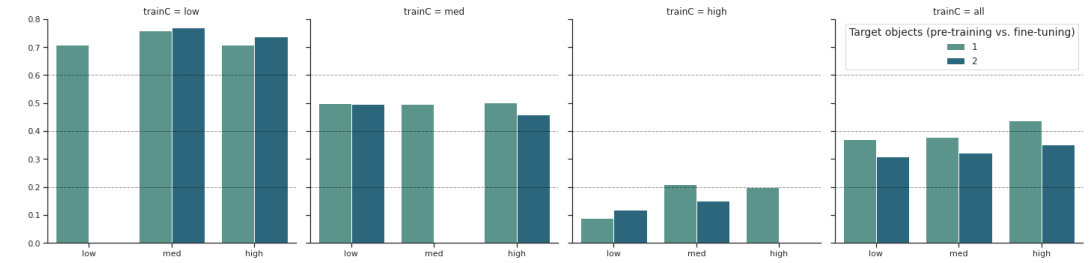


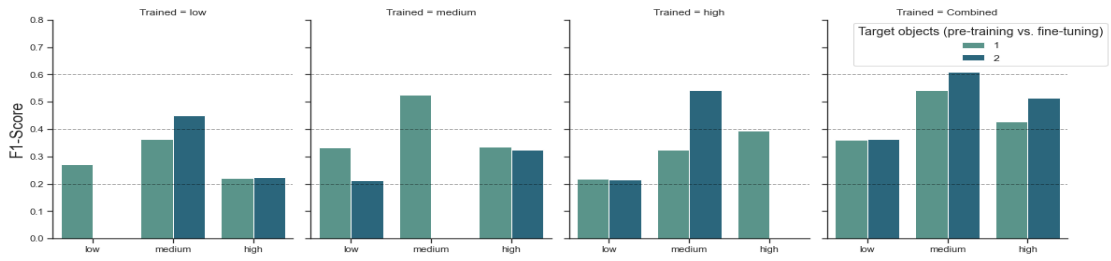
Figure 4: F1-scores (on the positive class) for varying normalization scopes



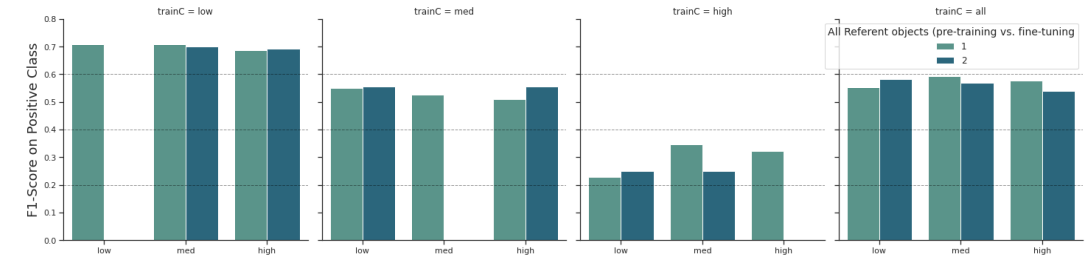
(a) Task-A: LSTM Model for Global Target Objects



(b) Task-A: TST Model for Global Target Objects



(c) Task-B: LSTM Model for Local Target Objects



(d) Task-B: TST Model for Local Target Objects

Figure 5: F1-scores on the positive class for Task-A and Task-B. Light green corresponds to pre-training, and the dark green to fine-tuning.

as empty in the graphs and as non-applicable (NA) on the Tables. Here we interpret the results from pre-training and testing on the same conditions. For the Task-A (referential gaze on target), transformer model (TST) produces a better performance overall; the within complexity train-test cycle resulted in $+0.15\%$ better for the LOW case compared to the LSTM Model, $+0.06\%$ for the MEDIUM. And

there is slight decrease -0.02% for the HIGH condition. The results for the Task-B is less conclusive. While TST performs better for the LOW condition to a substantial degree $+0.20\%$, LSTM's performance surpasses TST on the HIGH condition $+0.22\%$. And there is no difference in terms of performance on the MEDIUM condition.

6.4 Effect of Data Diversity

Results from training on COMBINED and testing on each condition shows the effect of using a larger and well representative dataset that contains various referential complexity settings, shown in the right-most graphs in Figure 5. Here, we observe that with rich data variety, without transfer learning, good results on both target and any referent prediction can be achieved. For the Task-A, the COMBINED condition provides the second best solution for the HIGH condition (almost comparable to the MEDIUM). In terms of model architecture, the TST model displays an advantage over LSTM in supervised learning from rich data. On the other hand, with further fine-tuning, LSTM results approach and even exceed the TST scores.

6.5 Cross-Complexity Results

Figure 5 shows the F1-scores (on the positive class) when transferring the LSTM and TST models across complexities (see Appendix A.6 for further details). The light green bars show results for pre-trained models, and the dark ones refer to the fine-tuned models. Overall, the most striking result is that the TST model trained in the LOW condition transfers very well to the MEDIUM and HIGH condition, even without fine-tuning. In effect, the overall best results on MEDIUM and HIGH are achieved by the TST model that is trained on the LOW condition. This is the case for both Task-A and Task-B (see the leftmost column in Figure 5). Generally, the TST model seems to benefit little from fine-tuning which may indicate that this additional training step introduces overfitting. This seems to be the pattern while testing on lower complexities than the training one (e.g. $\text{Train}_{\text{High}}$–$\text{Test}_{\text{Medium}}$, or $\text{Train}_{\text{Combined}}$–$\text{Test}_{\text{Low}}$) In contrast, the fine-tuning is highly instrumental on the LSTM's performance. Unlike TST, LSTM model is better at generalizing from the MEDIUM condition.

Moreover, training on the HIGH condition and testing on conditions of lower complexity does not seem to be successful in any model (see the third column in Figure 5). Overall, training on the

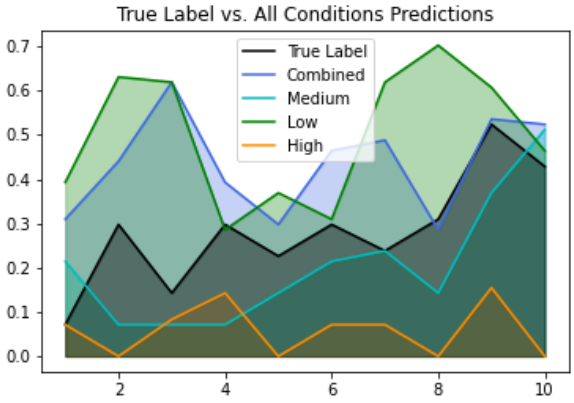


Figure 6: Aggregated model predictions on one image (medium condition) against the aggregated truth labels (from all participants)

MEDIUM condition achieves a medium accuracy which remains at a medium level in the other conditions. This pattern indicates that it is important to do the pretraining of gaze embeddings on a condition where the model can achieve high accuracy in referential gaze predictions. This leads to gaze representations that can be transferred well to other conditions.

At first glance, a stronger prediction performance on the LOW complexity is expected compared to other conditions. However, only TST model performs in line with this assumption. The number of objects is relatively small and has a low range (10 to 12) for all the scenes in that condition. It is possible that after detecting the relevant items, other objects are also being looked at by the participants until the trial ends (non-referential gaze). LSTM recurrent mechanism might be less sensitive about distinguishing referential gaze from other kinds of eye behaviors (like free viewing).

6.6 Scene-Specific Analysis

Further scene-specific analysis on the predictions provides insights about the temporal dimension of such predictions. However, first it should be noted that each participant might look at the referent objects at different point of time even while they are looking at the same image and hearing the same audio. This means that each participant produces unique ground truth labels (Appendix A.7). This makes the error analysis extremely challenging on referential gaze data. To address this issue, we believe that a sound method for error analysis will need to be developed and tested with care.

Although a full-scale error analysis is not in the scope of this study, we can look at the aggregated

data of all participants who saw the same scene. Figure 6 shows ground truth and TST models' predictions on a specific image. For each time interval (in the range of 1 to 10), we have aggregated the data collected from all participants in this condition as ground truth and model predictions respectively. For the sake of readability, individual model comparisons to ground truth have been presented separately per condition in the Appendix, Figure 9 to Figure 12.

This preliminary analysis supports our quantitative findings on transferring our referential gaze model from Section 6.5. Thus, the models trained on the LOW and COMBINED conditions (green and blue line) achieve the most stable prediction over the course of the sequence. Furthermore, the analysis indicates that the temporal dimension of the prediction is central. While the models exhibit difficulty to predict a referential gaze in the beginning of the sentence, the predictions become more reliable towards the end, except for the HIGH condition. When we look at the more stable second-half, we observe that the only under-generating model (producing false negatives) is still the HIGH condition. On the other hand, over-generation (false-positives) occurs more frequently with COMBINED and LOW conditions in the first half.

6.7 Summary

We now summarize the main findings from our investigation into the modeling of referential gaze. First of all, our results give some clear indications with respect to choice of model architecture and normalization procedures. The time-series transformer model (TST) outperforms the more basic LSTM architecture in most settings. Normalization of gaze features affects performance and across-study normalization is beneficial for low-resource or transfer settings.

Our results also clearly reveal that transferring gaze features between conditions and settings is far from trivial. Within-complexity results show that referential gaze prediction is possible from gaze features alone. All models beat the majority baselines in Task-A and Task-B (Section 6.3). Across-complexity results, however, demonstrate that some of the models are highly tuned to their specific communicative setting and do not generalize well.

The most robust models, in terms of generalization capabilities, are the TST model trained jointly on all conditions (COMBINED), and the TST model trained on the LOW condition only. Thus, our main finding is that gaze embeddings learned with models that achieve high accuracy in referential gaze prediction transfer well to other settings, even when they are trained on small amounts of data. We believe that this points into a very promising direction for future work on integrating NLP models with gaze processing.

7 Conclusion

Attending to referential gaze of the interlocutors is fundamental to face-to-face communication, yet still mostly ignored by the NLP community. In this study, we experiment with two deep learning methods (LSTM and transformer) to model referential gaze. We target gaze-only reference resolution and test how we can transfer the gaze features among various scene settings. Depending on the task (target or all-referent prediction) and the complexity level, the models exhibit different advantages. While TST is successful at generalizing from low complexities to higher ones and without the need of extra fine-tuning step, LSTM beats TST at generalizing from the MEDIUM conditions. But its performance is positively affected by fine-tuning.

One of the challenges of eye-movement modeling originates from being highly individual, task and environment dependent, making the generalization is more challenging. The results on different levels of gaze parameter normalization indicate that long-tail conditions clearly benefit from using more globally normalization. Within-complexity comparisons show that gaze features based on one scenario can be useful for similar new scenes. However, adopting among various complexities using pretrained models (with or without fine-tuning) displays encouraging results. Yet these result also confirm the challenging nature of the task and provide stepping stone for modeling referential gaze. Especially, the results are not trivial considering that we only use low-level gaze features. In addition to the gaze parameters, including the number of objects in the scene as a feature improves referential gaze prediction, indicating that this information makes the model more sensitive to various referential complexities.

References

Özge Alaçam, Eugen Ruppert, Amr R. Salama, Tobias Staron, and Wolfgang Menzel. 2020. Eye4ref: A multimodal eye movement dataset of referentially complex situations. In *Proceedings of the12th International Conference on Language Resources and Evaluation (LREC)*, page 2396–2404, Marseille, France.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*, pages 1–15, San Diego, CA, USA.

Maria Barrett and Nora Hollenstein. 2020. Sequence labelling and sequence classification with gaze: Novel uses of eye-tracking data for natural language processing. *Language and Linguistics Compass*, 14(11):1–16.

Filippo Maria Bianchi, Lorenzo Livi, Karl Øyvind Mikalsen, Michael Kampffmeyer, and Robert Jenssen. 2019. Learning representations of multivariate time series with missing data. *Pattern Recognition*, 96:106973.

Braiden Brousseau, Jonathan Rose, and Moshe Eizenman. 2020. Hybrid eye-tracking on a smartphone with cnn feature extraction and an infrared 3d model. *Sensors*, 20(2):543.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Zhenfang Chen, Peng Wang, Lin Ma, Kwan-Yee K Wong, and Qi Wu. 2020. Cops-ref: A new dataset and task on compositional referring expression comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10086–10095.

Volkan Cirik, Louis-Philippe Morency, and Taylor Berg-Kirkpatrick. 2018. Visual referring expression recognition: What do systems actually learn? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 781–787, New Orleans, Louisiana.

Alasdair DF Clarke, Micha Elsner, and Hannah Rohde. 2013. Where's wally: The influence of visual salience on referring expression generation. *Frontiers in psychology*, 4:329.

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.

Krista A Ehinger, Barbara Hidalgo-Sotelo, Antonio Torralba, and Aude Oliva. 2009. Modelling search for people in 900 scenes: A combined source model of eye guidance. *Visual cognition*, 17(6-7):945–978.

Wolfgang Einhäuser, Ueli Rutishauser, Christof Koch, et al. 2008. Task-demands can immediately reverse the effects of sensory-driven saliency in complex visual stimuli. *Journal of vision*, 8(2):2–2.

John M Henderson. 2003. Human gaze control during real-world scene perception. *Trends in cognitive sciences*, 7(11):498–504.

John M. Henderson and Tim J. Smith. 2009. How are eye fixation durations controlled during scene viewing? further evidence from a scene onset delay paradigm. *Visual Cognition*, 17(6-7):1055–1082.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Nora Hollenstein, Emmanuele Chersoni, Cassandra L. Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus. 2021. CMCL 2021 shared task on eye-tracking prediction. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 72–78, Online.

Dmytro Kalpakchi and Johan Boye. 2019. SpaceRefNet: A neural approach to spatial reference resolution in a real city environment. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 422–431.

Nour Karessli, Zeynep Akata, Bernt Schiele, and Andreas Bulling. 2017. Gaze embeddings for zero-shot image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4525–4534.

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.

Mohamed Khamis, Florian Alt, and Andreas Bulling. 2018. The past, present, and future of gaze-enabled handheld mobile devices: survey and lessons learned. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–17, Barcelona, Spain.

Sigrid Klerke and Barbara Plank. 2019. At a glance: The impact of gaze aggregation views on syntactic tagging. In *Proceedings of the Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*, pages 51–61, Hong Kong, China.

Nikolina Koleva, Martín Villalba, Maria Staudte, and Alexander Koller. 2015. The impact of listener gaze on predicting reference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for*

Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, volume 2, pages 812–817, Beijing, China.

Alexander Koller, Konstantina Garoufi, Maria Staudte, and Matthew Crocker. 2012. Enhancing referential success by tracking hearer gaze. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 30–39, Stroudsburg, PA, USA.

Qi Lei, Jinfeng Yi, Roman Vaculin, Lingfei Wu, and Inderjit S Dhillon. 2019. Similarity preserving representation learning for time series clustering. In *International Joint Conferences on Artificial Intelligence*, volume 19, pages 2845–2851, Macao China.

Minghao Liu, Shengqi Ren, Siyuan Ma, Jiahui Jiao, Yizhou Chen, Zhiguang Wang, and Wei Song. 2021. Gated transformer networks for multivariate time series classification. *arXiv:2103.14438*.

Nikolina Mitev, Patrick Renner, Thies Pfeiffer, and Maria Staudte. 2018. Using listener gaze to refer in installments benefits understanding. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society*, pages 2122–2127, Madison, Wisconsin, USA.

Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. 2019. Keras tuner. `https://github.com/keras-team/keras-tuner`.

Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. 2019. Few-shot adaptive gaze estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9368–9377, Seoul, Korea.

Zahar Prasov and Joyce Y Chai. 2008. What's in a Gaze? The Role of Eye-gaze in Reference Resolution in Multimodal Conversational Interfaces. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 20–29, Gran Canaria, Spain.

Keith Rayner, Xingshan Li, Carrick C Williams, Kyle R Cave, and Arnold D Well. 2007. Eye movements during information processing tasks: Individual differences and cultural effects. *Vision research*, 47(21):2714–2726.

Sol Rogers. 2019. Seven Reasons Why Eye-tracking Will Fundamentally Change VR. Retrieved on 15.05.2020.

David Schlangen, Sina Zarrieß, and Casey Kennington. 2016. Resolving references to objects in photographs using the words-as-classifiers model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1213–1223, Berlin, Germany.

Vidya Somashekarappa, Christine Howes, and Asad Sayeed. 2020. An annotation approach for social and referential gaze in dialogue. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 759–765, Marseille, France.

Ekta Sood, Simon Tannert, Philipp Müller, and Andreas Bulling. 2020. Improving natural language processing tasks with human gaze-guided neural attention. *Advances in Neural Information Processing Systems*.

Maria Staudte and Matthew W Crocker. 2011a. Investigating joint attention mechanisms through spoken human–robot interaction. *Cognition*, 120(2):268–291.

Maria Staudte and Matthew W. Crocker. 2011b. Investigating joint attention mechanisms through spoken human–robot interaction. *Cognition*, 120(2):268–291.

Ece Takmaz, Sandro Pezzelle, Lisa Beinborn, and Raquel Fernández. 2020. Generating image descriptions via sequential cross-modal alignment guided by human gaze. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4664–4677, Online.

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 *31st Conference on Neural Information Processing Systems (NIPS 2017)*, pages 1–11.

Niklas Wilming, Selim Onat, José P Ossandón, Alper Açık, Tim C Kietzmann, Kai Kaspar, Ricardo R Gameiro, Alexandra Vormberg, and Peter König. 2017. An extensive dataset of eye movements during viewing of complex images. *Scientific data*, 4(1):1–11.

Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. 2020. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv:2001.08317*.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. 2018. MattNet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315.

Gregory Zelinsky, Wei Zhang, Bing Yu, Xin Chen, and Dimitris Samaras. 2006. The role of top-down and bottom-up processes in guiding eye movements during visual search. In *Advances in Neural Information Processing Systems*, volume 18, pages 1569 – 1576.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2114–2124.

A Supplementary Material

Ethics Statement

The data used in this study involves (transcribed) verbal descriptions and eye-movements. No personal data that includes, e.g., name, age and education are shared.

Limitations

One of the main limitations is the limited number of eye-movement data samples that address real-world complexities. Having more diversity in terms of linguistic and visual manipulations is crucial to arrive at better generalization. This bottleneck can be overcome with an increase in the number of benchmark datasets. Another crucial component of this research is the multivariate time-series representation learning. As touched upon in Section 2.1, despite this topic attracting more attention, still it is at early stages to model intricacies of such time-series data.

In this study, the object count is used as an referential complexity parameter. However, it is not an expressive parameter especially for the LOW condition and draws the model further away from the other conditions, which differ substantially in terms of this parameter. This is probably also the reason why fine-tuning does not benefit the LOW condition. On the other hand, as expected, the HIGH complexity contains too much noisy data to be easily generalizable for LOW complexity models. Adapting gaze features between different extremes (e.g. $\text{Train}_{\text{LOW}} \rightarrow \text{Test}_{\text{HIGH}}$) is not as successful as adapting between similar complexities. The results highlight the importance of incorporating referential complexity, which also increases the gain from transfer learning by making models implicitly adaptive to referential complexity. However, the coarse-grain complexity definition provided in the original dataset is one of the main limitations to fairly evaluate the effect of this parameter. In further studies, we will focus on feature-based and more sophisticated referential complexity detection approaches.

A.1 Feature Vector and Labels

- Label for the target referent (1: if the gaze is on the target)
- Label for the all referents (1: if the gaze is on the any of the referents)
- Average time in blink
- Average time in saccade
- Resolution X
- Resolution Y
- Average pupil size
- Acceleration magnitude on X-axis
- Acceleration direction on X-axis
- Acceleration magnitude on Y-axis
- Acceleration direction on Y-axis
- Velocity magnitude on X-axis
- Velocity direction on X-axis
- Velocity magnitude on Y-axis
- Velocity direction on Y-axis
- ObjectCount'

A.2 Normalization

- Within-participant (WP) normalization
- Within-study (WS) normalization
- Across-study (AS) normalization

A.3 Architectures and Best Hyper-parameters

LSTM base model has 50 LSTM nodes. After the LSTM layer, we use two dense layers with 20 and 10 nodes respectively. For the binary classification on the single output layer, we use Sigmoid activation. Overall, the model contains 15,441 parameters. Best meta parameters after grid search; Learning rate = 0.0001; Loss = binary cross-entropy; Optimizer = Adam; Batch size = 128; Epochs = 100.

For the TST model, RAdam optimizer has been used. TST model size is set to 64-dimension. We used the implementation provided in the original Pytorch TST Library (Zerveas et al., 2021). Best meta parameters after grid search; Learning rate = 0.0001; Loss = binary cross-entropy; Optimizer = RAdam; Batch size = 64; Epochs = 50.

A.4 Runtime Settings

The experiments were conducted on a GPU server featuring 32 cores, 256 GB memory and 4 Geforce 1080Ti GPUs. No GPU parallelization was used. The average running time (including data input, model training and transfer learning on all test sets) is 75 minutes for the simplest condition with LSTM and 12 minutes with TST.

Hyperparameter Search The *Keras Tuner* library[2] (O'Malley et al., 2019) is used for finding best hyperparameters for different prediction tasks. We utilize the Random Search tuner with 100 epochs for LSTM and 50 for TST per run. A summary of the best performing model parameters can be found in Appendix A.

[2] https://www.tensorflow.org/tutorials/keras/keras_tuner

Table 2: Best hyperparameters of LSTM for the prediction tasks for (i) the target object, (ii) all communicatively relevant objects including the target

	Target Referent				All Relevant Referents			
	Low	**Medium**	**High**	**Combined**	**Low**	**Medium**	**High**	**Combined**
Learning rate	0.01	0.01	0.001	0.001	0.0001	0.001	0.001	0.001
LSTM nodes (units)	30	30	50		50	40	30	50
Dense-1 (units)	11	18	14		17	16	18	16
Dense-2 (units)	10	10	10		10	10	15	10

Table 3: Transfer learning with TST on within-class and between-class testing for all referents prediction task (Normalization Type: WP). (F1-scores on the positive class; Underlined values indicate best performance between models for each training set, bold values are the best on each test set.)

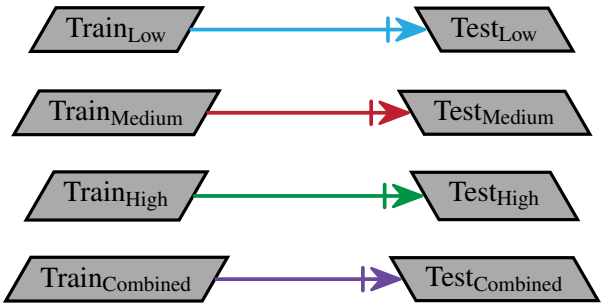
	Testset	**Low**		**Medium**		**High**	
	Model	**Pretrained**	**+Fine-tuned**	**Pretrained**	**+Fine-tuned**	**Pretrained**	**+Fine-tuned**
Training	**Low**	0.683	NA	0.675	0.659	0.722	0.699
	Medium	0.544	0.586	0.571	NA	0.529	0.559
	High	0.201	0.274	0.299	0.226	0.284	NA
	Combined	0.568	0.575	0.577	0.569	0.522	**0.589**

Table 4: Transfer learning with LSTM on within-class and between-class testing for all referents prediction task. (F1-scores on the positive class; Underlined values indicate best performance between models for each training set, bold values are the best on each test set.)

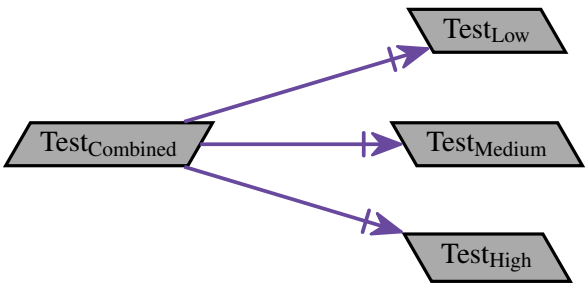
	Testset	**Low**		**Medium**		**High**	
	Model	**Pretrained**	**+Fine-tuned**	**Pretrained**	**+Fine-tuned**	**Pretrained**	**+Fine-tuned**
Training	**Low**	0.479	NA	0.412	0.412	0.329	0.379
	Medium	**0.489**	0.369	**0.569**	NA	0.437	0.413
	High	0.305	0.383	0.372	0.415	**0.505**	NA
	Combined	0.463	0.416	0.546	0.519	0.423	0.411

A.5 Train-Test conditions

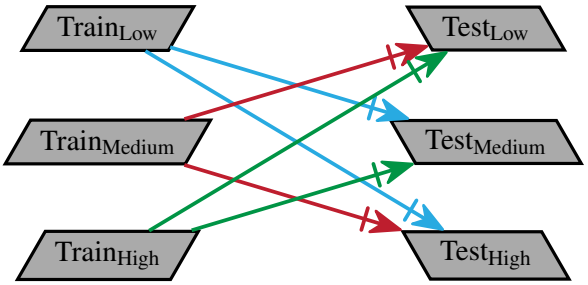
A.5.1 Within-complexity Conditions



A.5.2 Effect of Data Diversity



A.5.3 Transfer Learning conditions



A.6 Results Tables

The detailed scores for both models are presented in Tables 3 and 4.

A.7 Scene-specific Participant Analysis

In the following Figures, a ground truth and COMBINED model's predictions on test trials coming from two participants have been visualized. Both trials belong to same test image from MEDIUM condition and prediction results are taken from COMBINED model. As mentioned before, each participant produces different pattern and when we take the all participants and scenes in the study in

interaction with controlled parameters of this study, such analysis becomes highly complex.

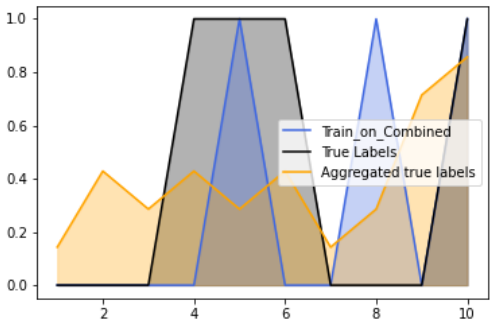


Figure 7: Participant-23 in MEDIUM condition (Scene 16), $\text{Train}_{\text{Combined}}$

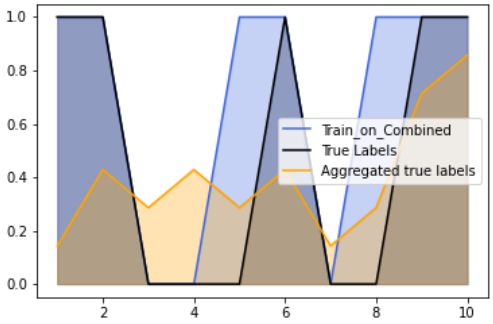


Figure 8: Participant-6 in MEDIUM condition (Scene 16), $\text{Train}_{\text{Combined}}$

A.8 Scene-specific Aggregated Analysis

The following figures illustrate individual TST model comparisons to the ground truths on a specific image separately per condition. For each time interval (in the range of 1 to 10), the model predictions for each participant are aggregated and compared against the ground truth.

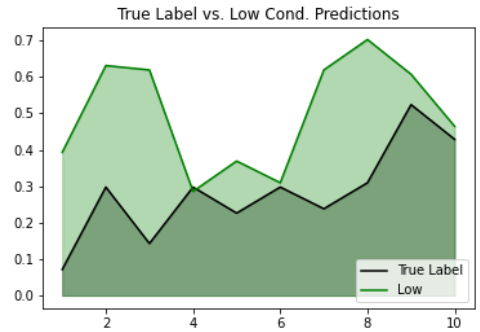


Figure 9: Aggregated $\text{Train}_{\text{Low}}$ model predictions on one image (medium condition) against the truth labels

A.9 Code Repository

The code and its documentation is available in this GitLab repository: https://gitlab.com/alacam/referential-gaze-modeling.

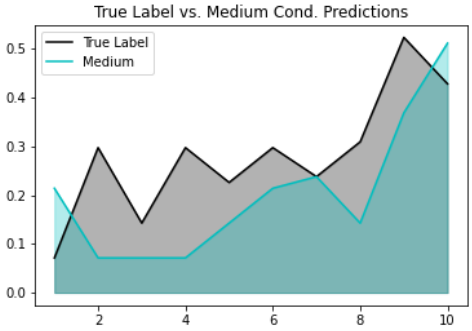


Figure 10: Aggregated $\text{Train}_{\text{Medium}}$ model predictions on one image (medium condition) against the truth labels

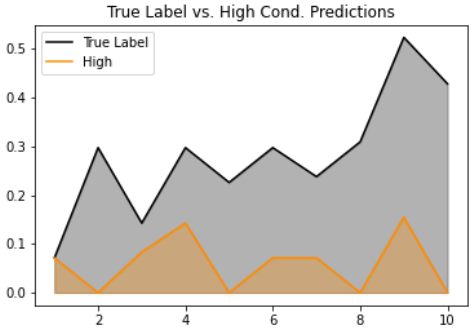


Figure 11: Aggregated $\text{Train}_{\text{High}}$ model predictions on one image (medium condition) against the truth labels

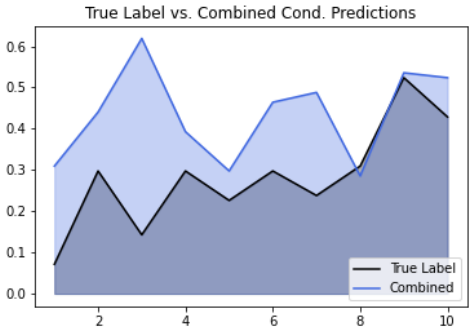


Figure 12: Aggregated $\text{Train}_{\text{Combined}}$ model predictions on one image (medium condition) against the truth labels