Developing a Pragmatic framework based on Transformers

Geoffroy de Gournay
geoffroydegournay@hotmail.com

Abstract

Some recent NLU papers focus on building a model attuned to the fact that humans live in a social physical environment and leverage that information. RSA based models have proved to offer an interesting framework for that purpose. These models need to handle the fact that when humans speak or listen, they always reason about other minds. Recent works have been using neural networks representing literal speakers and literal listeners, as building blocks for modelling more sophisticated pragmatic listeners and speakers. The neural speakers found in current RSA studies are encoder-decoders using RNNs. This study hypothesis is that speakers can be modelled more efficiently by using Transformers rather than RNNs. These models should especially help in the hardest cases, when the speaker needs to be more specific and needs to produce longer sentences. The intuition here, is that the powerful attention mechanisms used by Transformers, should help remembering the complex grounding aspects impacting language modelling throughout the whole process of building a sentence.

1 Introduction

Some recent NLU papers focus on building a model attuned to the fact that humans live in a social physical environment and leverage that information. Counterfactual reasoning about alternative utterances has been central to explanations of these pragmatic aspects of language (Grice, 1975). One popular account of pragmatic reasoning is the Rational Speech Acts (RSA) model (Goodman & Frank, 2016). RSA based models have proved to offer an interesting framework for that purpose. These models need to handle the fact that, when humans speak or listen, they always reason about other minds. Recent works have been using neural networks representing literal speakers and literal listeners, as building blocks for modelling more sophisticated pragmatic listeners and speakers.

RSA offers an interesting approach for grounding but has some limitations. In this model, one need to know the probability of all the alternative possible descriptions of a given context. As soon as the task considered involves a large enough vocabulary and a grammar that is complex enough, this calculation becomes practically unrealizable. Moreover, it is far from certain that humans, when pragmatically formulates utterances, actually consider all the alternative possible descriptions of a given context. In natural language processing, RSA-based models for pragmatic referring expression generation, use approximate inference methods, either sampling a subset of possible utterances from a learned model (Andreas & Klein, 2016; Monroe et al., 2017) or reasoning incrementally (Cohn-Gordon et al., 2019).

Many tasks in NLP have been revolutionized following the building of the Transformer model (Vastwani et al., 2017). Different models like BERT have used a derivation of that architecture and have outperformed existing models at several tasks. Natural language generation models like OpenAI’s GPT-2 have shown an impressive ability to produce human-like utterances. These results suggest that any task involving language modelling should consider the possibility to use a Transformer-based architecture.

The neural speakers found in current RSA studies are encoder-decoders using RNNs. This study first hypothesis is that speakers can be modelled more efficiently by using Transformers rather than RNNs. These models should especially help in the hardest cases, when the speaker needs to be more specific and needs to produce longer sentences.

The intuition here is that the powerful attention mechanisms used by Transformers, should help remembering throughout the whole process of building a sentence, the complex grounding aspects impacting language modelling. More specifically a speaker-based listener will outperform a neural literal listener in finding target colors using an adequate a description.

The next hypothesis is that this neural captioner, by being modified to make a system that is pragmatically informative, would lead to a new model outperforming the non-pragmatic baseline. An approach based on beam search performed by a neural speaker is proposed in this study. The hypothesis here is that it can offer an interesting and relative efficient way to approximate pragmatic reasoning.

Building a scalable model that proves to work in a relatively simple task could open the way to interesting developments in further studies or practical applications. The gain in performance achieved by the Transformer-based model on this task could be even more impressive with more complex datasets with pictures or videos. This model can be used for describing discriminatively pictures or videos (speaker) or selecting a target within a set of objects using a description (pragmatic listener).

The current study is performed using the Stanford English Colors in Context corpus (SCC). An RNN-based model and a Transformer-based model are trained, and their relative performance are compared. Their performance is analyzed across all the data and a more specific comparison in the 'close', 'far' and 'split' context is done as well. The performance of the pragmatic speaker-based listener vs a literal neural listener is also analyzed. Finally, a literal beam search approach is compared to the greedy one in order to analyze a pragmatic speaker.

This study finds that a Transformer-based model outperforms significatively the RNN-based model when used to model a speaker-based pragmatic listener. On the other hand, there was no real gain in performance by using the Transformer when the model is used as a speaker. The pragmatic speaker modeled using beam-search gave the worst performance.

2 Related Work

Various studies combining machine learning with probabilistic pragmatic reasoning models have for goal to model the fact that people reason about other minds and to show that modelling that aspect improves performance of both speakers and listeners at different tasks.

Monroe et al., 2017, Andreas et Klein, 2016, Cohn-Gordon et al., 2018 and White et al., 2020 build models around a version of the Rational Speech Acts (RSA) model (Frank and Goodman, 2012; Goodman and frank, 2016).

All these papers use RNN literal listeners and speakers to overcome the problem that the set of messages and the interpretation used in RSA models could be, in practical contexts, extremely large and potentially infinite. With that set up, they don't need to use domain knowledge in the form of a hand-written grammar or hand-engineered listener. To overcome the RSA models issues, Monroe et al., 2017 and Andreas et Klein, 2016 use generated samples of utterances when Cohn-Gordon et al., 2018 fully addresses this problem by implementing RSA at the level of characters rather than at the level of utterances or words. White et al., 2020 proposes an amortized speaker trained to directly optimize the RSA objective by using a pretrained neural listener when calculating its loss function.

Vaswani et al., 2017 present a powerful tool, the Transformer, that could be used to rethink and potentially improve the models of speakers and listeners described in the articles mentioned previously. There is still a connection to grounding in this article, although it might be not as clear as with the other ones. It deals with a local form of grounding, found in the sentence itself, giving to words a context captured thanks to attention mechanisms. Further research has shown that transformers can also deal efficiently with visual grounding as found in images and video. The wide range of applications of transformers in NLU make this paper an interesting one to consider, when working on a project focusing on grounding.

All these previous works related to RSA use RNN-based models. The current study explores if there are any gains that can be achieved by using a Transformer-based model.

The current study uses a Bayesian approach for modelling the pragmatic listener. A neural speaker-based listener directly infers the target color by calculating the probability that a color is a target implied by the speaker model it is based on. This approach is used by Cohn-Gordon et al., 2019 as a convenient way to evaluate its speaker, but there is

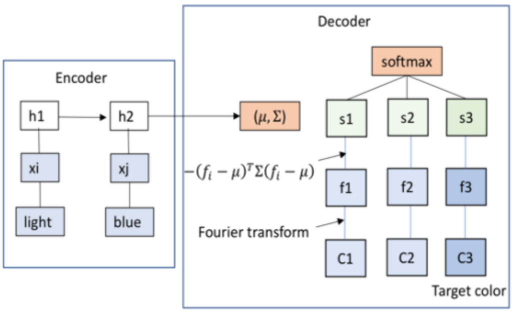


Figure 1: Neural LSTM literal listener L_0.

no detailed analysis of the listener in this article. On the pragmatic speaker side, the current study has a neural speaker performing a simple beam-search algorithm.

3 Data

The dataset used for this study is the Stanford English Colors in Context corpus (SCC) (Monroe et al. 2017). The SCC corpus is based on a two-player interactive game. The two players share a context consisting of three colors patches, with the display order randomized between them so that they can't use positional information when communicating. The speaker is privately assigned a target color and asked to produce a description of it that will enable the listener to identify the speaker's target. The listener makes a choice based on the speaker's message, and the two succeed if and only if the listener identifies the target correctly. The two players played repeated reference games and could communicate with each other in a free-form way.

To obtain this corpus of natural color reference data, 9767 participants from Amazon Mechanical Turk were recruited to play 1059 games of 50 rounds each. To ensure a range of difficulty, an equal number of trials from three different conditions were randomly put in place: 1) close, where all three colors were similar but still sufficiently distinct to be perceptible, 2) split, where one distractor is close to the target color, but the other distractor is easily differentiated, and 3) far, where all colors were easily differentiated one from the other.

The resulting filtered dataset consists in a corpus of 53,365 speaker utterances across 46,994 rounds in 948 games. The three conditions are equally represented, with 15,519 close trials, 15,693 split trials, and 15,782 far trials. For each entry the following fields are provided: game identifier, round number, worker identifier, round condition, time of message, the three colors and their position as seen by the speaker and as seen by the listener, which color was the speaker target, which color was selected by the listener, and time of listener's click.

The two players played repeated reference games and could communicate with each other in a free-form way. This opens up the possibility of modeling these repeated interactions as task-oriented dialogues. However, in this study, most of this structure is ignored. The corpus is treated as a bunch of independent reference games played by anonymous players, and the listener and their choices are entirely ignored. There are cases where the speaker made a sequence of utterances for the same trial. Following Monroe et al., 2017 these are concatenated into a single utterance.

4 Models

The starting point of RSA is a model of a literal listener that speaks a language and makes a guess about what the world is. This guess is made on the basis of the language he speaks and the message he receives. A pragmatic speaker is then derived from that literal listener. This pragmatic speaker is not just reasoning about the language but rather about a listener who reasons about the language. Finally, a pragmatic listener is derived from the pragmatic speaker.

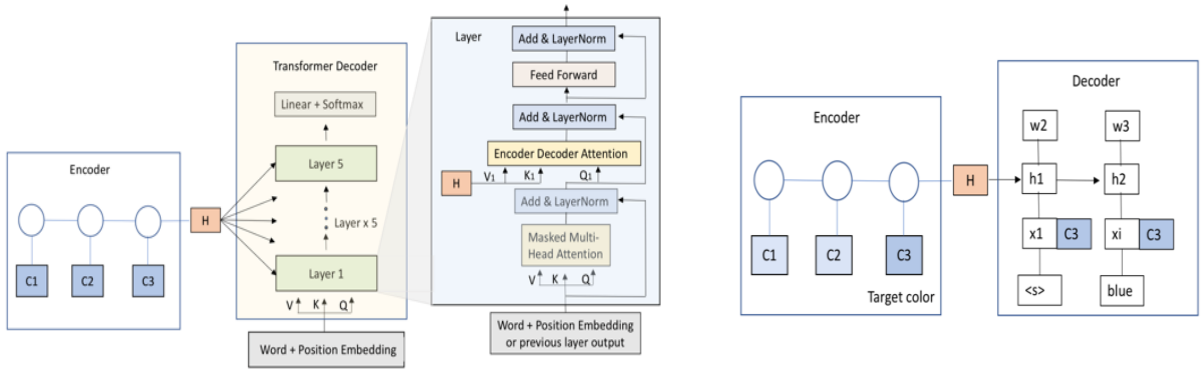
Following Monroe et al., 2017, an alternative formulation of a pragmatic listener L_1 can be started from a literal speaker S_0:

$$\mathrm{S_0(u|c, Lex\ ) \propto Lex(u, c)e^{-\kappa(u)}} \qquad (1)$$

$$\mathrm{L_1(c|u, Lex\ ) \propto S_0(u|c, Lex)P(t)} \qquad (2)$$

where c is a color in a context C, u is a message drawn from a set of possible utterances U, P is a prior over colors, $\varkappa$ is a cost function on utterances and $Lex(u, c)$ is a semantic interpretation function that takes the value 1 if u is true of c, 0 otherwise. This is the approach that is followed here, focusing on a neural speaker rather than on a neural listener as a starting point.

To apply the RSA model to discriminative color captioning, two neural models with different architectures are considered here. These models represent the literal speaker $S_0(u|c, C; \theta)$, θ being the vector of parameters that define the trained



(a) S_0^{TRA} : GRU context encoder and Transformer description decoder with 5 layers and 2 attention heads. The details of the composition of a layer are given on the right of the figure.

(b) S_0^{GRU} : GRU context encoder and GRU description decoder.

Figure 2: The neural base speaker agents. The S_0 agent process the target color C_3 in context and produces tokens of a color description sequentially. Each step in production is conditioned by the context representation H and the previous word produced.

model.

These neural models will serve several purposes. They are firstly used to derive a pragmatic listener, effectively choosing the target color within a context, considering the probability that a literal speaker would formulate the description that's been made for such a context of colors.

$$L_1(c|u, C; \theta\,) = S_0(u|c, C; \theta) \qquad (3)$$

This listener uses Bayes' rule to calculate, from the neural speaker, the probability of each color to be the target color given a context of two distractor colors and the description u given. L_1can be used as a literal agent to evaluate the performance of a speaker. Neural speakers can also be used as speaker S_0themselves, or as a basis for modeling a pragmatic speaker S_1.

$$S_1(u|c, C; \theta\,) = \frac{L_1(c|u, C; \theta)}{\sum_{u'} L_1(c|u', C; \theta)} \qquad (4)$$

The summation in (4) is done over alternative referents in the context. In the original RSA formulation S_1 is derived from L_0 but the focus of this study being rather on improving the neural listener, S_1 is derived from L_1.

GRU-based speaker: The GRU-based speaker S_0^{GRU}, consists in a GRU context encoder and a GRU description decoder (Figure 2 (b)). The colors of the context are transformed into Fourier representation space and are passed through a GRU with 150-dimensional hidden state. The context is reordered to place the target color last, minimizing the length of dependence between the most important color and the output. Sentences are tokenized, as described previously, and the caption $u = (c_0, c_1, \dots, c_T, c_{T+1})$ is a sequence of $T + 2$ tokens, with $c_0 =< SOS >$ and $c_{T+1} =< EOS >$. The final hidden state of the encoder is used as the initial encoder hidden state. The GRU decoder is initially fed with the final hidden state of the encoder and use a 100-dimensional glove representation of words concatenated with the 54-dimensional Fourier representation of the target color for training. At each step a linear transformation and softmax are applied to the output to produce the following token of the description.

Transformer-based speaker: Following recent advances in language modeling the alternative speaker candidate studied here is a Transformer-based speaker S_0^{TRA} (Figure 2 (a)). This model is a modification of the GRU-based speaker, keeping the same encoder but using the decoder architecture of a Transformer as a decoder. This model uses multiheaded self-attention both to propagate information along the sequence of utterance tokens, as well as to fuse visual and textual features. During training, the forward

	Number of	condition		
	rounds	far	split	close
All data	46994	33.60%	33.40%	33.00%
Train	31210	32.90%	33.60%	33.50%
Dev	8358	31.90%	33.20%	34.90%
Train Speaker	16275	32.50%	33.70%	33.80%
Train Listener	16227	31.90%	33.60%	34.50%
Hyper	5031	31.60%	33.50%	34.90%

Table 1: All the subsets of the data have a balanced repartition between the "far", "split" and "close" conditions. Hyper is the subset on which training is done for hyperparameters tuning.

model receives two inputs: color features from the decoder, and a caption describing the target color. As previously, a 54-dimensional Fourier space representation of colors is used, and the same tokenizing process is applied to the utterances. The encoder has a 128-dimensional hidden state. The model is trained to predict $u_{1:T+1}$token-by-token, starting with <sos>. First, the tokens of u are converted to vectors via word embedding and positional embeddings. Next, these vectors are processed through a sequence of 5 Transformer layers. Each layer performs masked self-attention with 2 heads over token vectors and color vectors and applies a two-layer network to each vector. These three operations are each followed by a layer normalization. After the last Transformer layer, a linear transformation and softmax are applied to the output to produce a probability distribution over the vocabulary for the next token.

LSTM-based Literal Listener L_0: This agent is derived from (Monroe et al., 2017). It runs a bidirectional LSTM over the utterance to predict a Gaussian distribution over colors space with mean vector μ and covariance matrix Σ (Figure 1). The captions are embedded using 100-dimensional glove embedding and the LSTM uses a 50-dimensional hidden state. Colors are embedded using the 54-dimensions Fourier Transformed representation. A score is computed for each color representation using the following formula:

$$\mathrm{score} = (c - \mu)^{T}\Sigma(c - \mu) \qquad (5)$$

A softmax function is applied to the three scores calculated to produce a probability distribution.

Pragmatic Listener: The pragmatic listener is based on a pretrained speaker S_0. It uses Bayes' rule to obtain from S_0 the posterior probability of each color being the target when the two others are the distractors given a full utterance u. When choosing which color is the target, L_1 calculates: $c^* = \mathrm{argmax}_{c \in C} S_0(u|c, C; \theta)$ where c is the target color and C the context of c and two distractors.

Pragmatic Speaker: The pragmatic speaker S_1 is based on L_1 (4), and by construction of L_1 (3), S_1 is ultimately built from a pretrained speaker S_0. In order to approximate the normalization in the pragmatic speaker (which cannot be performed directly because it sums over all alternative utterances) beam search is performed. When building a beam of alternative possible partial sentences, the speaker chooses, within a restricted number of alternative sentences, the one that would effectively make choosing the target color, the most likely option for a pragmatic listener L_1. In this framework, L_1 effectively considers how likely is it that a literal speaker would produce such a sentence given a context of colors. The question, that a speaker S_1 using beam-search considers, is how likely is it that a pragmatic listener would find the target color for each alternative formulation provided by the beam-search.

This approach is only an approximation of what an RSA framework requires. In theory all the alternative possible formulations of the literal speaker would need to be considered. In practice, as soon as the vocabulary and the grammar become large enough it becomes impossible to perform such calculation.

5 Experiments

5.1 Evaluation methods

The experiments are conducted on the SCC data set of Monroe et al., 2017. The data are split between a 'Train' (64%), 'Dev' (16%), and 'Test' (20%) subsets.

The 'Train' subset is split between a 'Train Listener' and 'Train Speaker' subsets so that the speaker S_0 used to build the pragmatic listener L_1 , that will be used for evaluation, doesn't use the data on which the training of the evaluated speaker is done. Separate data are necessary to avoid the speaker production model effectively having access to the system evaluating it. A last subset, 'Hyper', is used for hyperparameters tuning. It is built from 15% of the training subset. After preprocessing, the data are organized following the repartition described in Table 1. As the repartition was calculated on raw lines of data, some rounds got split in the process, but the impact on the study should be neglectable. It explains why the number of rounds between datasets doesn't add up exactly. The different subsets of data have a balanced repartition of conditions between 'far', 'split' and 'close', which is what is needed for the current study.

Colors are represented in the data in RGB space as a three-dimensional vector. They are first converted to HSV vectors and are then transformed into a Fourier basis representation as in Monroe et al. (2016). The representation for a color $(\mathrm{h}, \mathrm{s}, \mathrm{v})$ is given by:

$$\mathrm{f} = \left[\mathrm{Re}\{\hat{\mathrm{f}}\}, \qquad \mathrm{Im}\widehat{\{\mathrm{f}\}}\right] \ \mathrm{j}, \mathrm{k}, \mathrm{l} = 0..2 \qquad (5)$$

where $\hat{\mathrm{f}}_{\mathrm{jkl}} = \exp[-2\pi \mathrm{i}(\mathrm{j}\mathrm{h}^{*} + \mathrm{k}\mathrm{s}^{*} + \mathrm{l}\mathrm{v}^{*})]$
and $(\mathrm{h}^{*}, \mathrm{s}^{*}, \mathrm{v}^{*}) = \left(\frac{\mathrm{h}}{360}, \frac{\mathrm{s}}{200}, \frac{\mathrm{v}}{200}\right)$.

Utterances are preprocessed as in Monroe et al., 2017: lowercasing, tokenizing by splitting off punctuation as well as the endings -er, -est, and -ish and replacing the tokens that don't appear in the training split with <unk>.

Initially an hyperparameters optimization is done for all the models studied. For that purpose, models are trained on 'Hyper' and evaluated on 'Dev'. Both Adam and Adadelta, adaptative variants of stochastic gradient descent (SGD) are used. Both the option of using 100-dimensional glove embedding or random embedding for tokens are considered. When performing hyperparameter search with Adadelta higher learning rates (around 0.2) where used than when performed with Adam (around 0.001). The following options were kept:

- GRU-based speaker: Glove embedding, 150-dimentional hidden states for both the encoder and decoder, batch size of 64, early stopping, Adam optimizer, learning rate of 0.005.
- Transformer-based speaker: no Glove embedding, encoder and decoder 128-dimensional hidden states, 2 attention heads, 5 Transformer layers, batch size of 128, Adam optimizer, learning rate of 0.001, 512-dimentional feedforward layer.
- Literal LSTM-based listener: Glove embedding, 50-dimentional hidden states for the encoder, batch size of 32, Adam optimizer, learning rate of 0.005.

The memory of the GPU used was not sufficient to experiment with 3 or more attention heads for the Transformer-based speaker. Models used as listeners are trained on 'Train Listener' and models used as speakers are trained on 'Train Speaker'.

Evaluate models as listeners: The metric used to evaluate the listeners is accuracy.

Evaluate models as speakers: Three analysis are performed to evaluate the performance of the model as a speaker S_0. When a listener model is used for the evaluation of the speaker, it is trained on 'Train Listener', which is an independent set of data from 'Train Speaker', the one used for S_0. Firstly, a pragmatic listener L_1, derived from S_0 is used.

The capacity of L_1 to correctly interpret the sentences produced by S_0 is measured using accuracy. L_1 uses Bayes' rule to obtain from a neural speaker S_0 the posterior probability of each target color given a full caption. Secondly, the same accuracy is calculated but using the neural literal listener L_0 this time.

The last analysis looks at the quality of the language produced. Language should be concise, giving the most information needed with the fewest possible words. The BLEU score is considered to complete the accuracy score. This score gives a metric to check that the sentences uttered by the speaker are close to what human would produce. It allows to measure how good the system would be in communicating with humans.

Model	Dev	Test
L_0	76.7	76.7
L_1^{GRU}	81.0	80.6
L_1^{TRA}	82.7	82.9

Table 2: Accuracy results for listener models. Models are trained on ‘Train Listener’.

The Human BLEU score is calculated using human utterances in the Data as gold outputs. The L_1 BLEU score is calculated using sentences produced by the speaker model as gold outputs. This is not a traditional way of using this score, but it gives an indication of the stability of the model, when it is used as a speaker, trained on one set of data, and as a listener trained on another set of data. It also gives a way to put into context the results obtained with the Human BLEU score by looking at the spread between the two scores.

5.2 Results

Listeners: The accuracy of the three listener models are provided in Table 2. The three models are trained on ‘Train Listener’ and evaluations are done on ‘Dev’ and finally on ‘Test’. Both L_1^{GRU} and L_1^{TRA} outperform significatively L_0 on both dev and test data. L_1^{TRA} is the best model for a listener. L_1^{TRA} significatively outperforms L_1^{GRU}.

Speakers: The different speakers were trained on ‘Train Speaker’. Listeners used to calculate ‘L_0 accuracy’, ‘L_1accuracy’ and ‘L_1BLEU’ are trained on ‘Train Listener’. Speakers have no access to the systems evaluating them. It avoids having a model better at communicating with itself using its own language than with others. As seen in Table 3, S_0^{GRU} , is the best of the three speakers, the difference with S_0^{TRA} being fairly small. Surprisingly, S_1^{TRA}, is the worst by far of the three models.

6 Analysis

The neural speakers considered here are not really literal agents. They are already pragmatics in the sense that they’ve been trained on human data, built in a reference game. This data should already have a pragmatic dimension embedded in it and as long as the agents considered here are trained on such data, they already have learned some pragmatic behavior.

It is interesting to look at the different metrics across the different conditions: ‘close’, ‘split’ and ‘far’ (Table 4). As one would expect, for all the models studied here, the performance is positively correlated to the difficulty of the task. When colors are getting close, the sentences need to be longer, more detailed and semantically more complex. It is also interesting to notice that it is on the most difficult task, in the ‘close’ condition, that L_1^{TRA} outperforms the most L_1^{GRU}. This fact is important as it confirms the intuition that the advantages of a Transformer based architecture increases with more complex tasks. The mechanism of self-attention allows the Transformer to look at other positions in the input sequence for indications that lead to a better encoding for this word in a potentially more powerful fashion than does the hidden state for an RNN. On top of self-attention, the multi-headed attention expands the model’s ability to focus on different positions and allows it to project the input embeddings in different subspaces.

Model	$\boldsymbol{S_0^{GRU}}$	$\boldsymbol{S_0^{TRA}}$	$\boldsymbol{S_1^{TRA}}$
L_0 accuracy	84.8/85.1	84.9/84.2	79.3
L_1accuracy	90.6/91.3	90.0/90.1	83.4
Human BLEU	50.1/50	49.2/49.7	
L_1 BLEU	80.7/80.8	79.0/78.7	74.7

Table 3: Accuracy and BLEU scores for speaker models. Speakers are trained on ‘Train Speaker’, listener used for evaluating the speakers are trained on ‘Train Listener’. The first result is the one calculated on ‘DEV’, the second on ‘TEST’. No test calculation was done for S_1^{TRA} .

One important point is that there is an asymmetry in the current study given the structure of the dataset. The purpose is to learn a task from humans and be as good as them at this task if not better. So, the models are trained on data built with human utterances made while playing at a reference game. When the analysis is focused on the model considered as a listener, the accuracy can be compared precisely with the human’s one. The model takes human utterances as input and has to decide which color is the target, exactly as the human listener does and the performance of both humans and the model can be compared. On the other hand, when the focus is on the model considered as a speaker, the evaluation is less natural. Ideally, the model would play versus a human, and one would be able to compare the captioning ability of the model versus the one of humans.

	$\boldsymbol{L_1^{GRU}}$	$\boldsymbol{L_1^{TRA}}$	$\boldsymbol{S_0^{GRU}}$	$\boldsymbol{S_0^{TRA}}$
Close	70.9	73.2	75.2/82.8	76.2/83.3
Split	81.0	83.1	82.8/90.6	82.6/89.3
Far	92.0	92.7	97.3/99.2	96.9/98

Table 4: Accuracy across conditions. For S_0, the results are given for L_0/L_1.

Here, alternative approaches need to be taken to evaluate the model. By looking at the speaker-based listener L_1 and a neural literal listener L_0, trained on different sets of data, a relative 'fair' and independent assessment of the performance is made. Using the same model as a basis for a literal listener, as long as the speaker and listener are trained on segregated data, is somewhat in line with the position of human beings, being both speakers and listeners sharing among them some conceptual structures. Adding the independently modeled listener add some strength to the conclusions in case there would be some hidden flaws in the approach previously described. The human BLEU score presents the advantage to keep a 'human connection' in our data as it is calculated using human made sentences. It is good to keep that metric when the other ones are calculated with machine evaluating machine produced data. The L_1 BLEU score emphasize that last point. This metric is shown here not really as a performance metric but to be compared with the Human BLEU score. The spread between the two is a good indication that there must be some conditions where the models studied here, diverge in the structure of the sentence they produce from the ones one could expect to be produced by a human (Table 3).

It would be interesting to have a human evaluation of the performance of the speaker models considered here, in order to assess more precisely their communication skills. A deeper analysis of why the pragmatic speaker S_1^{TRA}, doesn't outperform the other speakers could also lead to interesting conclusions. Lastly, the fact that the Transformer-based architecture significantly outperforms the GRU one on the listener side and not on the speaker side would be interesting to explore. Anyhow, this study comforts the idea that, when working on more complex backgrounds, using a Transformer encoder coupled with a decoder should improve substantially the performance, at least in terms of accuracy.

7 Conclusion

The present study has analyzed comparatively a Transformer-based and an RNN-based architecture as building blocks for captioning or interpreting human utterances in a grounded context. It has shown that Transformer based models could offer an interesting alternative to RNN-based ones in a grounded task and could be used as a good foundation on which can be built pragmatic agents.

Acknowledgments

We would like to thank Professor Christopher Potts of Stanford Linguistics for discussing ideas with us about the project direction.

Authorship

All persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take responsibility for the content. Specifically, this project consisted of literature review, designing the experiment, developing the code, training the model, analyzing results, writing this paper.

Appendices

The GitHub repository for this project code is available at: github repository

References

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* pages 1173–1182. Association for Computational Linguistics.

Reuben Cohn-Gordon, Noah Goodman, and Chris Potts. 2018. Pragmatically informative image captioning with character-level inference. *In Proceedings of the 2018 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies, NAACL- HLT, volume 2,* pages 439–443.

Karan Desai, and Justin Johnson. 2020. Virtex: Learning Visual Representations from Textual Annotations. arXiv preprint arXiv:2006.06666

Goodman, N. D., & Frank, M. C. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, 20(11), 818-829.

Grice, H. P. (1975). Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and semantics: Vol. 3: Speech acts*. New York: Academic Press.

Will Monroe, Robert XD Hawkins, Noah D Goodman, and Christopher Potts. 2017. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 5:325–338.

Ashish Vaswani, Noam Shazeer, Niki Parmar, JakobUszkoreit, Llion Jones, Aidan N Gomez, ŁukaszKaiser, and Illia Polosukhin. 2017. Attention is all you need. *In Advances in Neural Information Processing Systems*. pages 6000–6010.

Julia White, Jesse Mu, Noah D. Goodman. 2020. Learning to refer informatively by amortizing pragmatic reasoning. *In Proceedings of the 42nd Annual Meeting of the Cognitive Science Society*.