Predicting guilt judgments from news stories

Elisa Kreiss
Department of Linguistics
Stanford University
ekreiss@stanford.edu

Abstract

News has become a central part of our lives and with it our assessments and reproductions of these stories. However, the way in which humans assess guilt when given a story about a crime and suspects is unclear. If neural networks can learn to predict guilt judgments from a news story, they can help us form hypotheses about relevant features for human guilt predictions. However, whether neural networks can even learn to predict guilt judgments from a given news story is still an open question. In this paper, I will present a bidirectional LSTM with self-attention that makes a promising first step into that direction.

1 Introduction

In the current world we are constantly confronted with news. We consume it by reading the newspaper, listening to news channels, but also through social interactions, for example conversations with friends. A crucial aspect of news is the uncertainty that comes with different stories. For example news can report on advances in crime investigations that are still in progress, such as the arrest of a suspect. Most importantly, an arrest does not imply that the suspect is in fact guilty or even that the investigation is completed. Reporting is often not simply about guilty or not guilty, but a graded notion of guiltiness presenting a range of evidence from definite to only suggestive.

After observing the news, they are retold and passed on through social interactions. It is essential that the uncertainty which was previously communicated, will propagate to the next person we tell the story to. This person in turn can tell the story to someone and so on. If we had perfect memory, the story would stay the same throughout the reproductions. However, reproduction is not only a lossy but also faulty process. (Bartlett, 1932) introduced this transmission chain paradigm as a scientific method and showed not only that information get lost, but also that they get distorted. Subsequent work found that the person's priors and beliefs shape their reproductions of stories (Allport and Postman, 1945; Kashima, 2000; Griffiths and Kalish, 2007). In other words, people's initial guilt assessments after hearing a story will affect how they will reproduce it. It is therefore relevant which guilt judgments people form and what these are influenced by.

However, guilt judgments are very complex. Assessing the likelihood of someone's guilt could be dependent on how persuasive we find their alibi, how condemning their motive and even prejudices against race or culture of the suspect could play a role. Additionally, the presentation of the news seems relevant, for example is a lot of detail given, what does the author who presents the news think, do the news seem to be written in a biased way. There are many more of these potential aspects that we can reasonably assume to affect our guilt judgments. For less complex questions, one would look for likely predictors and see whether a model which assumes these predictors makes similar predictions as we see in the empirical data. However, this seems infeasible given the complexity of the problem.

A promising way to capture these complex functions is through neural network models. Can neural network models learn to predict human guilt judgments? In particular this paper investigates whether a bidirectional LSTM with attention can model empirically collected guilt ratings of propagated news stories. If so, the features those models learn can inform us about our theories on what the relevant features could potentially be. Generally, neural network predictions can be used to assess the perception of news stories to ensure news stories in fact communicate the intended amount of uncertainty. Failing to do so increases

the problem of unintentional false news propagation. On the long run, these systems can then also be used to reverse engineer which aspects of news stories might affect guilt judgments the most and which might do so less than expected. This can inform the way we write news that corresponds to the listeners' most likely interpretations.

1.1 Related work

The central challenge is to find a promising neural network architecture that can extract relevant features of an input paragraph. This can then be used to predict a value that represents a guilt judgment.

Word embeddings There are a variety of systems that investigate how to find meaningful word representations. In the literature, GloVe word embeddings (Pennington et al., 2014) have been widely used to represent word meanings. Those embeddings are context-independent, i.e., each word only has one respective vector representation. However, this means that for example polysemous words have the same representations and when there is no Part-Of-Speech attached, even the verb and the noun *run* have the same meaning representation.

More recent advances in word embeddings aim to provide context-sensitive word representations to obtain richer and more informative word meaning representations (e.g., ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018)). The BERT word embedding model is a Transformer based network which incorporates the context before and after the critical word to obtain a word representation. Devlin et al. report that using these embeddings boosts performance in all of the 11 natural language tasks it was trained on compared to previous systems.

Paragraph embeddings While word embeddings are useful, we need to arrive at a meaningful **paragraph** representation for our purposes. In other words, the word embeddings need to be combined in a meaningful way to arrive at a more complex representation.

For short phrases, bidirectional LSTMs (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005) have been shown to be a useful tool. However, they have difficulty resolving long-term dependencies, which is why their performance decreases with longer input paragraphs (Hochreiter et al., 2001). (Vaswani et al., 2017) developed a system they call "attention" aiming to resolve these issues in Transformers. However, a similar principle can be applied to LSTM architectures as well.

(Lin et al., 2017) introduce a bidirectional LSTM with self-attention that receives word embeddings as an input and outputs a paragraph representation. The bidirectional LSTM receives the GloVe embeddings of the paragraph as input. The self-attention layer is the weighted sum of all of the LSTM's hidden states. But instead of using a single weighted sum to arrive at a vector, they suggest to perform this multiple times with different weight vectors to arrive at an attention matrix. Furthermore, they introduce a penalization term which ensures that each attention vector in this attention matrix focuses on different semantic features of the input paragraph. Which semantic features are more important than others can in turn be learned again, dependent on the task it needs to solve. Lin et al. report that the model outperforms previous models in author profiling, sentiment classification and textual entailment.

2 Dataset

For training and testing the model, I use the Annotated Iterated Narration Corpus (AINC) collected by (Kreiss et al., 2019). The corpus was created for investigations on how news stories change when they are propagated from one person to another. First the authors created five stories of approximately 850 words, which are called the *seed* stories. All of them report a crime and an arrest of one or more suspects. Each of these five seed stories exists in a weak and a strong evidence condition. For example, when a suspect was arrested because of camera footage, this footage was described as being of "very poor quality" in the weak evidence condition and "very high quality" in the strong evidence condition. Each story was given to a participant who were asked to read and afterwards reproduce it. The reproduced story was then given to the next participant who again reproduced it. This pattern was repeated for 5 generations of participants. The data collection therefore followed the transmission chain paradigm, introduced by (Bartlett, 1932). Following this schema, each story and condition was reproduced in 5 different chains over 5 generations, resulting in 250 reproductions, and therefore 260 stories overall (see Figure 1).

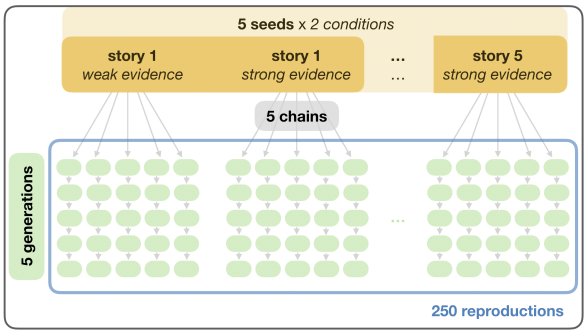

Figure 1: Corpus collection schema as presented in (Kreiss et al., 2019).

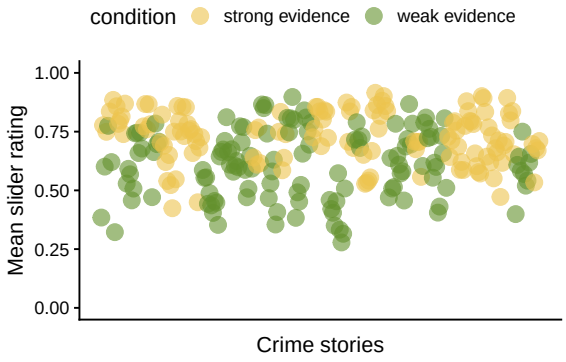

Figure 2: A point represents the mean subject guilt rating for a story in the corpus. They are color-coded with respect to their condition.

After the corpus collection, Kreiss et al. annotated the corpus with human judgments, mainly about guilt but also for example perceived subjectivity of the story writing. Participants were recruited on Amazon Mechanical Turk and indicated their response on a continuous slider, here underlyingly coded as ranging from 0 to 1. Most interestingly for this project, they asked "How likely is it that the suspect is / the suspects in the crime are guilty?". Each story receives approximately 20 ratings, ranging from 0 to 1. For the purpose of this work, we will consider the mean rating for each story as the guilt judgment label. The labels of each story are shown in Figure 2. In contrast to the raw ratings, their means only range between 0.27 and 0.92. Even though this range is smaller than in the raw data, there is still a lot of variance to explain.

Subsequent analyses revealed that there are correlations with the number of hedges in the story, the length of the stories and of course with the evidence condition and therefore content expressed.

In summary, the AINC is a corpus of reproduced news stories, annotated by human guilt judgments. One of the challenges, this dataset poses, is its size. There are only 260 labeled examples to train the model on. Can a bidirectional LSTM with attention pick up on the patterns in human judgments with only very limited available data? If so, neural network models might present an exciting opportunity to test hypotheses about what could influence human judgments.

3 Model

In the following section, I will explain the model architecture and how it was trained.

3.1 Architecture

To predict guilt judgments, I used a bidirectional LSTM with a simple attention mechanism. Figure 3 visualizes the model architecture.

The model receives a news story as an input to a BERT word embedding layer. The story has previously been transformed to all lower-case letters and has been tokenized using the Bert-Tokenizer implementation in the python package *pytorch-pretrained-bert*[1]. The BERT word embeddings for each token were extracted from the Google-pretrained base BERT word embeddings (uncased) with 768 hidden layers.

The BERT embeddings are the input for a bidirectional LSTM with 200 hidden dimensions.

The LSTM's hidden states are fed through an attention layer which outputs a vector of dimensionality 400. We arrive at the attention vector by taking the weighted sum of the LSTM's hidden states. The attention layer is implemented in the following way: Each hidden state is first reduced in dimensionality from 400 to 50 dimensions using a Tanh activation function. This again is linearly transformed to a dimensionality of 1, i.e., a single number. We will then take the Softmax over all numbers obtained from different hidden states. Now each hidden state has a state vector (dimensionality 400) and a number associated with it. All hidden states taken together, we have a vector with the single numbers we obtained from the hidden states (1 x input length) and a matrix with all hidden states (input length x 400). We then take the dot product of the two and arrive at the final atten-

[1] Huggingface Inc., https://github.com/huggingface/pytorch-pretrained-BERT, June 09 2019

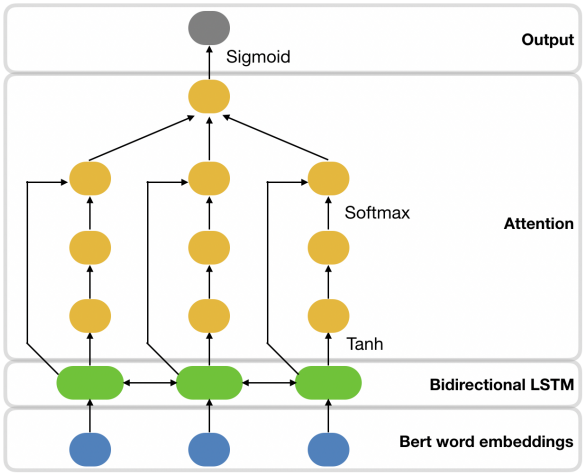

Figure 3: Proposed model architecture to predict guilt judgments (between 0 and 1) from news stories.

tion vector of size 400.

For a prediction, this is then linearly transformed into a single number and fed through a Sigmoid function to ensure a prediction between 0 and 1.

In summary, the model receives a news story as an input. BERT word embeddings for the story are used as an input to a bidirectional LSTM. The weighted sum over the LSTMs hidden states builds up the attention layer. The attention vector is fed through a linear layer and a Sigmoid function to ensure that the prediction will be between 0 and 1.

3.2 Training

From the 260 stories with their respective mean suspect guilt rating, 26 were randomly held back as a final testing set. The guilt rating is the mean obtained from the previously described data collection and annotation. Crucially, the model does not learn a classification, but regression task, which is why the target values range between 0 and 1. The remaining 234 stories were used for training the model.

To assess the model performance during development, we need to exclude more stories from the training data to be able to use it as dev-testing data. Therefore, the model was trained using 10-fold cross-validation to make the most use of the limited data. The results can also inform us, whether there is strong variation given a different training/dev-testing split. It is very likely that given the small size of the dataset (and therefore dev-test set), the variation can be quite high. Given that there were 234 stories for training, each fold had between 23 and 24 stories, each associated with one guilt rating that functioned as the target label.

For training, the model uses mean squared error (MSE) as the loss function, stochastic gradient descent as the optimizer and a learning rate of 0.1. The whole model was implemented using pytorch (Paszke et al., 2017). It was trained for 30 epochs, for each of the 10 training/dev-test configurations in the cross-validation.

4 Results

This work investigates whether a bidirectional LSTM with self-attention (as described in Section 3.1) can predict human guilt judgments from news stories. In each step of the 10-fold cross-validation, the model was trained on 206/207 stories and 23/24 were held out for dev-testing. Each story has an associated suspect guilt rating, which is the mean rating obtained from human judgments. This value functions as the target label.

Since the target labels only range between 0.27 and 0.92, one potential problem might be that the model only learns to predict the overall mean of the labels in the training data. We would also expect this outcome if the model does not find any features that it can take as predictors for guilt ratings. This is why, the baseline model to test the dev-test data on will simply predict the mean of the training data labels.

Figure 4 shows the mean squared error (MSE) loss for each training epoch. Firstly, the training loss (in blue) is approaching a loss of 0 toward the end of the training in all cross-validation configurations. Thus the amount of training epochs seems to be sufficient. Crucially, the dev-testing loss (in orange) is overall lower after the last training epoch than the baseline (in yellow). Therefore, the model obtains information from the dataset that it can use to improve its predictions. However, there is also a high amount of variation between the different cross-validation steps. The mean of the training labels alone (i.e., the baseline) only has a very small loss on cross-validation configuration 1 and the trained model can barely beat it. This is in clear contrast to cross-validation step 3, where the baseline model loss is very high on the dev-test data and the the trained model can easily surpass it. Those two cases exemplify the high variation that comes with the different splits of training and

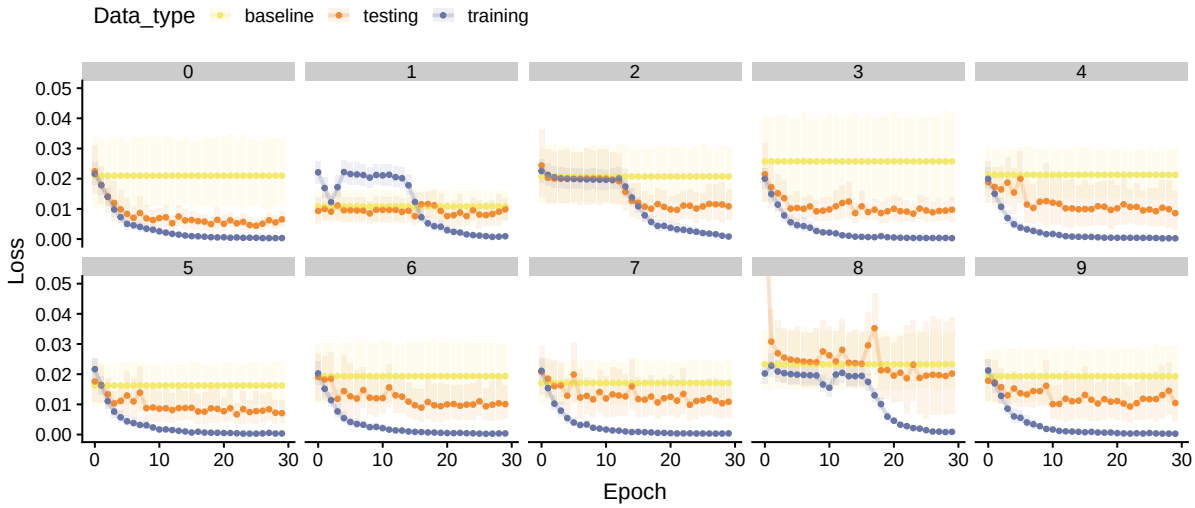

Figure 4: Loss (mean squared error) over epochs (x axis), faceted over cross-validation configurations. The performance of the model on the training set (in blue) approaches zero. The performance of the model on the dev-test set (in orange) generally outperforms the baseline (in yellow).

dev-testing data[2].

But looking at the MSE loss alone, is not on its own sufficiently informative to assess how well the model actually learns to predict the underlying distribution. Figure 5 shows the correlation between the actual target label (on the x axis) and the model prediction (on the y axis) for cross-validation step 0. Before training, the training and dev-testing data has a high variance around the perfect correlation line (r=1). The model only predicts values between 0.48 and 0.79. However, the underlying labels range from 0.31 to 0.92. Even though the Pearson correlation still predicts a rather high correlation ($r = 0.65$) on the testing set, the mean squared error is comparably high with 0.022.

Already qualitatively, the model predictions after training (Figure 5) seem far more informative. Now the model predictions have approximately the same range as the target labels (between 0.33 and 0.91). The Pearson correlation itself is now 0.85, and the mean squared error reduced to 0.007.

Since the baseline model does not learn over training epochs, its predictions stay the same.

Qualitatively, these plots look very similar throughout all cross-validation configurations and can be compared in Figure 7 and 8 in the Appendix. However, quantitatively the correlation on the dev-test set changes. Those differences are

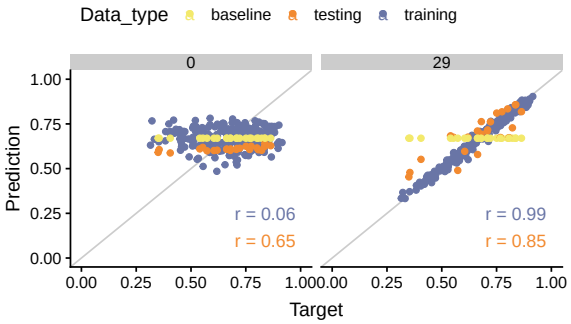

Figure 5: Correlation between target label (x axis) vs. model prediction (y axis) before and after training.

mainly driven by outliers.

As a final performance evaluation, we can examine the target-prediction correlation on the held-out test set (see Figure 6). For predictions on the test set, I used the final model weights obtained by the first cross-validation configuration. This setup was chosen, because of the low dev-testing loss and the high correlation of the dev-test set with the target labels.

The Pearson correlation on the held-out test set is still 0.84 and almost identical with the performance on the dev-test set. This fairly high correlation and the fact that the high correlation reproduces with the held-out testing data indicates that the model learns something meaningful that can generalize well.

[2]However, I cannot exclude that also the random parameter initialization plays a relevant role here. But at least the differences in the baseline are definitely caused by the different splits.

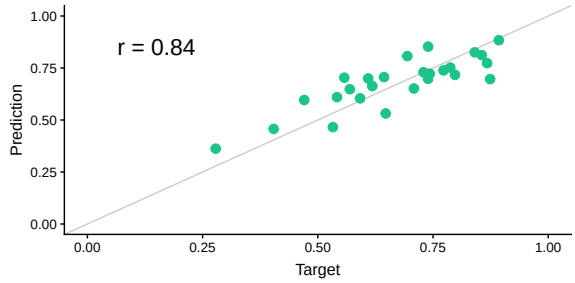

Figure 6: Testing label (x axis) vs. model prediction (y axis) after training on the held out test set (26 data points) using the parameter settings obtained after the 30th epoch from the first cross-validation. Pearson correlation of 84%.

5 Discussion

The results show that when given a news story, a bidirectional LSTM with self-attention can already capture something essential about human guilt judgments. Especially considering the limited amount of data (only 236 data points in the training set), the model can already extract meaningful features for guilt judgment predictions.

So far, the model has not been subjected to hyperparameter tuning. It is likely that this will further boost the model's performance. Furthermore, it is plausible that training and testing the model on the guilt ratings directly instead of their means can further increase the performance, since only showing the means hides a vast amount of information about the underlying distribution of ratings which can be very meaningful. For example, a news story where participants highly vary in their judgments should influence the model predictions less than a story with low variance.

The introduced model is a promising first step to develop a better understanding of human guilt judgments. I am confident that in the future models like these can be used to inform hypotheses about what humans consider for their assessment. This can in turn inform us about how different reporting on a news story will influence these assessments.

Acknowledgments

I'm particularly thankful for the valuable feedback from Chris Potts and Sebastian Schuster. Furthermore, I would like to thank Judith Degen and the ALPS lab for their helpful ideas and questions.

Project authorship statement

This is an individual project and therefore, E. K. was responsible for all parts of the projects.

References

Gordon W Allport and Leo J Postman. 1945. Section of psychology: the basic psychology of rumor. *Transactions of the New York Academy of Sciences*, 8(2 Series II):61–81.

Frederic C Bartlett. 1932. Remembering: An experimental and social study. *Cambridge: Cambridge University*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Thomas L Griffiths and Michael L Kalish. 2007. Language evolution by iterated learning with bayesian agents. *Cognitive science*, 31(3):441–480.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

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

Yoshihisa Kashima. 2000. Maintaining cultural stereotypes in the serial reproduction of narratives. *Personality and Social Psychology Bulletin*, 26(5):594–604.

Elisa Kreiss, Michael Franke, and Judith Degen. 2019. Uncertain evidence statements and guilt perception in iterative reproductions of crime stories. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 41.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

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

6 Appendices

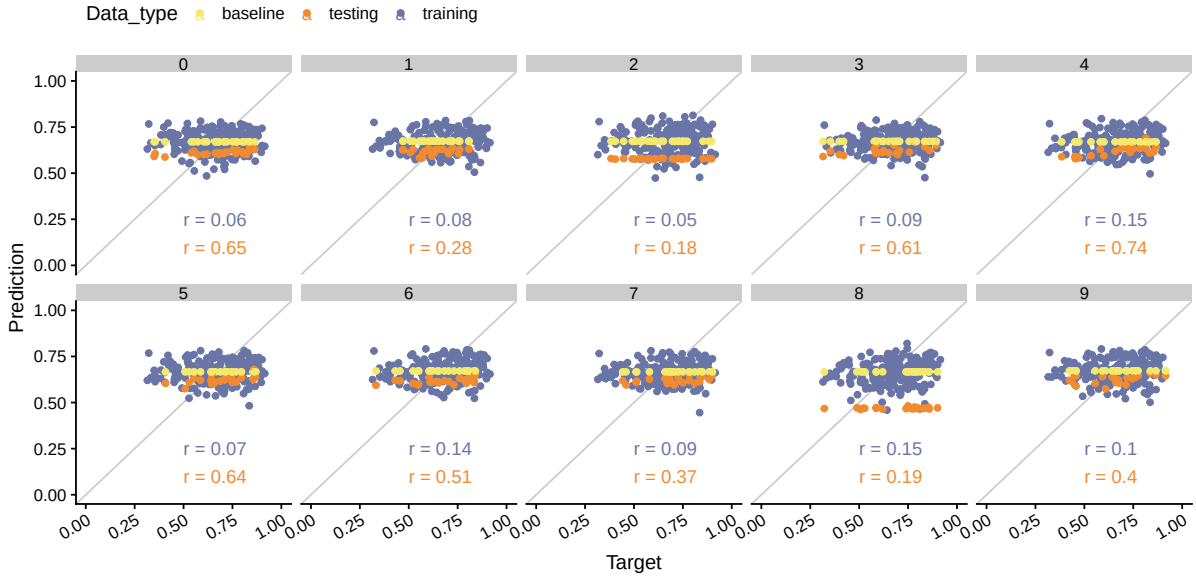

Figure 7: Testing label (x axis) vs. model prediction (y axis) before training; faceted over cross-validation configurations.

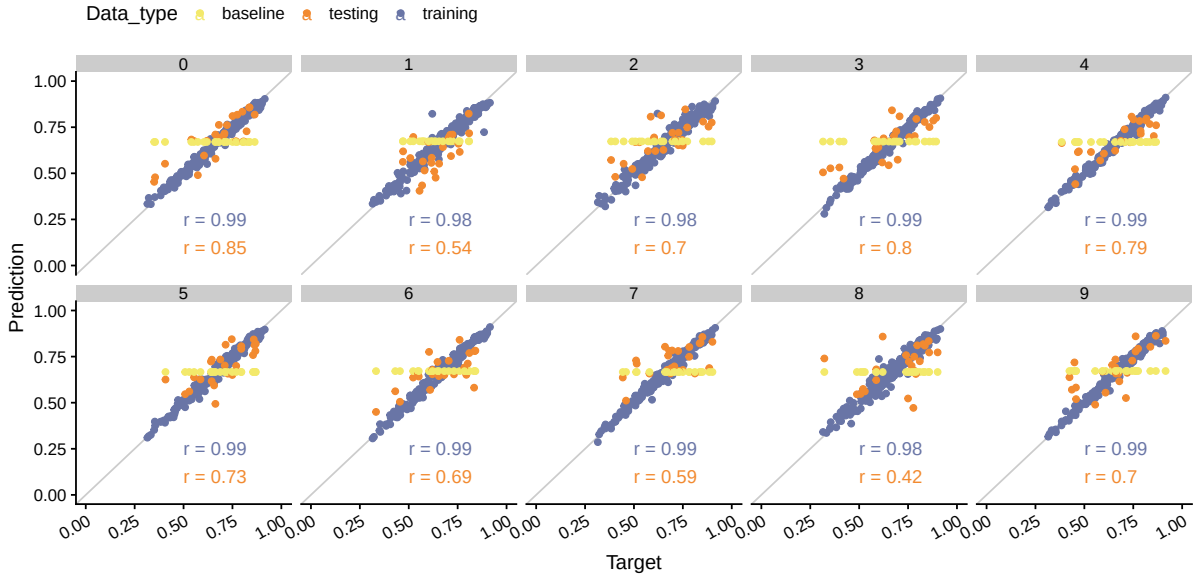

Figure 8: Testing label (x axis) vs. model prediction (y axis) after training; faceted over cross-validation configurations.