Attention Weights as a Window into How Crime Narratives Shape Predictions about Guilt

Anonymous ACL submission

Abstract

Neural network attention mechanisms have led to performance gains on a wide variety of tasks in natural language understanding, and they have also provided clues as to how these complex networks represent the language they process, opening up new avenues for linguistic investigation. In this spirit, we apply neural networks with attention to the task of modeling human judgments about guilt based on short crime narratives. The networks are successful at the task, and probing their attention weights helps illuminate the ways in which they make use of markers of certainty and uncertainty in different crime narratives, suggesting new hypotheses about how crimes might be more effectively reported in the news media.

1 Introduction

Deep learning models are increasingly applied to language tasks not only to develop technologies but also to derive new insights into language use. These more scientific applications raise the question of whether the models are processing data in a motivated way; their "black box" nature is often an obstacle to answering this question (Alishahi et al., 2019). Recently, attention mechanisms, which capture the strength of association between components of these networks (Bahdanau et al., 2015; Luong et al., 2015), have not only improved performance, but also increased interpretability, potentially opening the door to new linguistic applications of these models ([ek: paper that challenges this idea a little:](Serrano and Smith, 2019)).

In this paper, we apply neural networks with attention to a task that has both linguistic and societal import: predicting whether the reader of a short narrative crime report will conclude that the main subject is guilty or innocent. Our networks achieve strong performance on this task, as defined by the dataset of Kreiss et al. (2019), which leads us to explore their learned parameters by inspecting how they make predictions for new, carefully controlled examples. Our central finding is that, where the evidence is weak, the networks make robust use of markers of certainty and uncertainty, as indicated by the large attention weights for such terms. In contrast, where the evidence is strong, these markers contribute less (have smaller attention weights). To the extent that humans adopt similar reading strategies, this can inform how the news is reported, in that it gives us insights into the contribution of words like *allegedly* and *possibly* in shaping readers' construal of newspaper articles.

2 Related work

2.1 Predicting guilt

The challenge of predicting guilt judgments from text sources has not yet received a lot of attention. However, Fausey and Boroditsky show that using agentive language increases blame and financial liability judgments people make. Their results suggest that even subtle linguistic changes in crime reports can shape people's judgments of the events. More recent work has focused on predicting guilt verdicts from the Supreme Courts in the Philippines (Virtucio et al., 2018) and Thailand (Kowsrihawat et al., 2018) on the basis of facts presented and law texts. Kowsrihawat et al. argue for a recurrent neural network with attention to make these predictions. Note that this work is not concerned with the linguistic basis of subjective moral guilt judgments, but Court guilt verdicts on the basis of a given law text. It is our goal to train a network on subjective guilt ratings and then probe its parameters with respect to their linguistic basis.

2.2 Interpretation of hedges

In the literature, researchers have categorized and labeled (un)certainty markers in numerous ways, e.g., (Lakoff, 1972; Prince et al., 1982; Brown et al., 1987). For a summary we refer readers to consult (Fraser, 2010). For this work, we use "hedge" as an umbrella term for all subclasses of uncertainty markers. For our purposes, "hedge" refers to any marker that introduces uncertainty "within the propositional content" (e.g., His feet were **sort of** blue.), or "in the relationship between the propositional content and the speaker" (e.g., **I think** his feet were blue.).

There is also extensive prior literature on how hedges affect the perception of the speaker and proposition, painting a highly varied picture on the interpretation of hedges (Erickson et al., 1978; Durik et al., 2008; Bonnefon and Villejoubert, 2006; Rubin, 2007; Jensen, 2008; Ferson et al., 2015). These studies suggest that hedges affect people's judgments of credibility in differing ways, for example an increase in the number of hedges decreases the credibility of witness reports (Erickson et al., 1978) but at the same time increases the trustworthiness of journalists and scientists (Jensen, 2008). Additionally, the interpretation of hedges is very context dependent (Bonnefon and Villejoubert, 2006; Durik et al., 2008; Ferson et al., 2015) and shows high individual variation (Rubin, 2007; Ferson et al., 2015). We investigate how neural network predictions about human judgments change when we manipulate the presence of those hedges in crime stories.

2.3 Linguistic basis in neural networks

Neural networks have been shown to achieve high performances on various NLP tasks, such as sentiment analysis (Socher et al., 2013; Devlin et al., 2019) and translation (Bahdanau et al., 2015). A growing amount of literature has tried to gain linguistic insight from those neural networks (Belinkov and Glass, 2019). For example, neural networks seem to learn some basic syntactic structures. Futrell et al. find evidence for basic incremental syntactic state representations by treating neural network outputs like human subjects (Linzen et al., 2016). A vast amount of recent literature finds that that contextual word embedding models like BERT (Devlin et al., 2019) have structural information encoded (Jawahar et al., 2019; Tenney et al., 2019; Clark et al., 2019). Clark et al. find indications that particular attention heads in BERT specialize to specific aspects of syntax. Generally attention weights have been used for the analysis of neural language models, for example, in tasks such as NLI (Rocktäschel et al., 2015; Yin et al., 2016), sentiment analysis and (Wang et al., 2016; Liu and Zhang, 2017) summarization (Rush et al., 2015; Allamanis et al., 2016). In these studies, attention weights seem to align with natural language intuitions and are used as windows into what guides predictions. In this work we visualize the attention weights and probe the model outputs to gain insight into what the model learns for guilt prediction.

3 Dataset

For training and testing the model, we use the Annotated Iterated Narration Corpus (AINC) collected by (Kreiss et al., 2019). [cp: I think we have to make this corpus public and include the URL to download it, else this will be seen as an indirect violation of anonymity.][ek: I see, that makes sense. Chris/Judith: What is a good way to make the corpus public? Github?] 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. The stories are identical up to the last phrase, which then either raises doubts about the arrest or emphasizes its validity. For example, if a suspect was arrested on the basis 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 was 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 chains over 5 generations, resulting in 250 reproductions, and therefore 260 stories overall.

After the corpus collection, Kreiss et al. annotated the corpus with human judgments. The

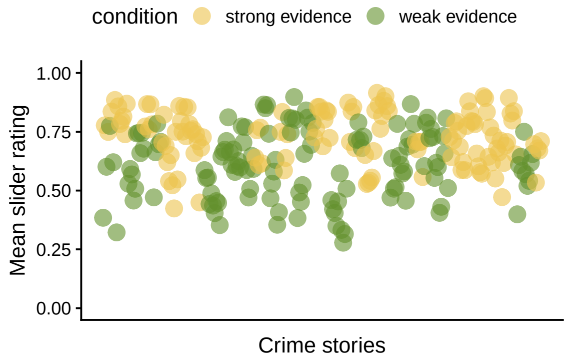

Figure 1: A point represents the mean subject guilt rating for a story in the corpus. They are color-coded with respect to their condition.

questions were primarily related to different aspects of guilt perception 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 received approximately 20 ratings, ranging from 0 to 1. For the purposes of this work, we consider the mean rating for each story as its guilt judgment label. The labels of each story are shown in Figure 1. In contrast to the raw ratings, the means only range from 0.27 to 0.92.

In summary, the AINC is a corpus of reproduced news stories, annotated by human guilt judgments. Since the stories originated from only 5 unique stories (each in 2 conditions), a lot of information is shared between single data points. Despite this similarity, the range of guilt judgments is still high, alluding to subtle differences that trigger this variance. We now turn to the question of whether a deep learning model can learn to predict the assessments of guilt the corpus provides.

4 Model

Our model is based on the one proposed by Lin et al., which is fundamentally a bidirectional LSTM with attention mechanisms applied to its outputs. We chose this model primarily because its attention mechanisms seem especially promising for introspection, as they essentially provide a weight for each word in the input. The overall architecture is summarized in Figure 2.

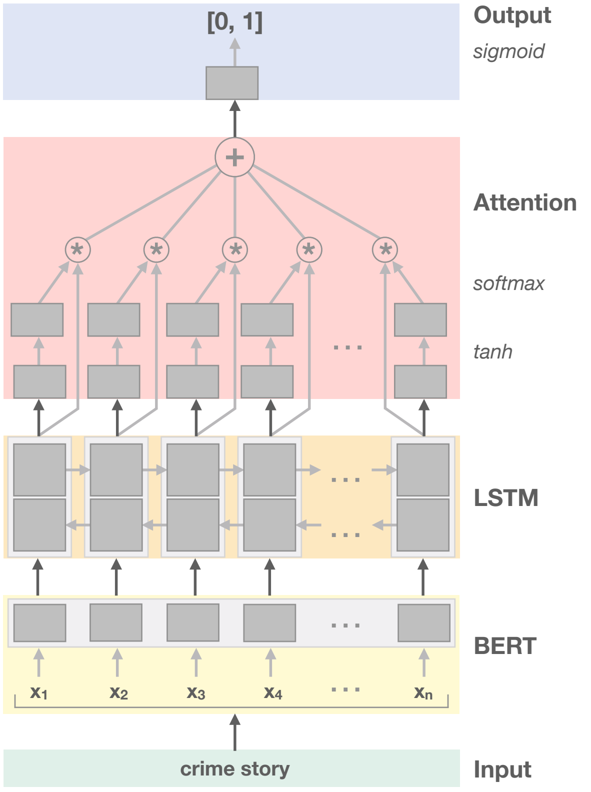

Figure 2: Model architecture. [cp: I would not draw the output of the BERT layer as a single gray rectangle. The output is a sequence of token representations, and these are fed into the LSTM.] [ek: Is this better? The operations in the BERT layer are context sensitive which is why I didn’t simply want to put boxes next to each other without any connection between them.]

The only major adjustment we made to the model is that we replaced the GloVe word embeddings (Pennington et al., 2014) by BERT representations (Devlin et al., 2019). Whereas GloVe provides a single embedding for each word, with no sensitivity to the context in which it occurs, BERT representations vary by syntactic context. 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, and we saw comparable improvements when we switched from GloVe to BERT. To access pretrained BERT parameters, we used the Hugging Face toolkit.[1]

We allow BERT to tokenize the input string according to its internal tokenization method (Wu et al., 2016), to make maximal use of its own pretrained embedding. These token representations are processed by BERT using its pretrained Trans-

[1] https://github.com/huggingface/pytorch-transformers

former parameters (Vaswani et al., 2017), yielding contextual representations for them. These token lists are padded or clipped as needed to ensure that they always have length n. [cp: Elisa, could you verify this and provide any necessary details that are missing? What is the value of n too?] [ek: n is variable and the BERT tokenizer doesn't clip in any way. The input size to the network is variable.] These are fed into a bidirectional LSTM layer in which each's cell's output has dimension 200. The two representations at each step are concatenated to form the LSTM layer output for each token.

For the attention layer, we follow the design of Lin et al.: the LSTM outputs are organized into a matrix H of dimension $n \times 400$, and we apply a dense layer with parameters W (dimension 50×400) and a $\tanh$ activation to create a matrix A of 50-dimensional representations for each token:

$$A = \tanh(WH^{\top}) \tag{1}$$

The matrix A is further compressed to the attention weight vector $\mathbf{a_w}$ of size 1×50, with a softmax applied so that the weights sum to 1:

$$\mathbf{a_w} = \mathrm{softmax}(WA) \tag{2}$$

Lin et al. actually generalize this attention operation to return a matrix with r attention weights for each token. We set $r = 1$ to keep the number of parameters low and increase the interpretability of the network.

Finally, we take the dot product of the LSTM hidden state matrix H and the just obtained vector a_w to obtain the attention output vector $\mathbf{a}_{\text{out}}$ of size 1×400:

$$\mathbf{a}_{\text{out}} = \mathbf{a_w} H \tag{3}$$

This is the output of the attention module. Intuitively, the attention weights capture the relative importance of each token with regard to the final prediction, and $\mathbf{a}_{\text{out}}$ synthesizes all of these weighted contributions into a single vector.

Finally, the logistic regression module makes the prediction of guilt. For this step, $\mathbf{a}_{\text{out}}$ is linearly transformed into a single number $p = \mathrm{sigmoid}(\mathbf{a}_{\text{out}})$. The sigmoid function ensures that the prediction lies between 0 and 1, just like the restriction on the guilt judgments it is trained on.

Overall, the model has $111,054,742$ trainable parameters. This sounds mismatched with our very small dataset, but only 0.02% of these parameters are in the attention and output modules and 1.40% are in the LSTM module. The rest of the parameters (98.59%) are the pretrained weights in the BERT word embedding module.

In summary, the model receives a crime story as an input. BERT word embeddings for the story feed into a bidirectional LSTM. The attention module computes the weighted sum over the LSTM's hidden states. The resulting attention vector is fed through a linear layer and a Sigmoid function to ensure that the prediction lies between 0 and 1.

5 Experiment

Our initial goal is to assess the extent to which our bidirectional LSTM with self-attention (as described in Section 4) can predict human guilt judgments from news stories. Assuming the model succeeds, we can then probe its internal representations for linguistic insights.

5.1 Optimization

To begin with, we held out 26 randomly selected stories (from the 260 stories in total) as the final test set. The remaining 234 stories were then for used for model training and validation, which was done using 10-fold cross validation. The cross-validation results inform us about the model variation observed across various training/validation splits; it is quite likely that, given the small size of the dataset, this variation could be quite high.

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. As noted above, this guilt rating is the mean participant guilt rating obtained from the previously described data collection and annotation.

Both model performances will be evaluated on the mean-squared error (MSE) of their prediction to the target label on the cross-validation test set.

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.

5.2 Results

Figure 3 shows the mean squared error (MSE) loss for each training epoch. We show the dev-set loss for our model (orange) as well as the train-set loss

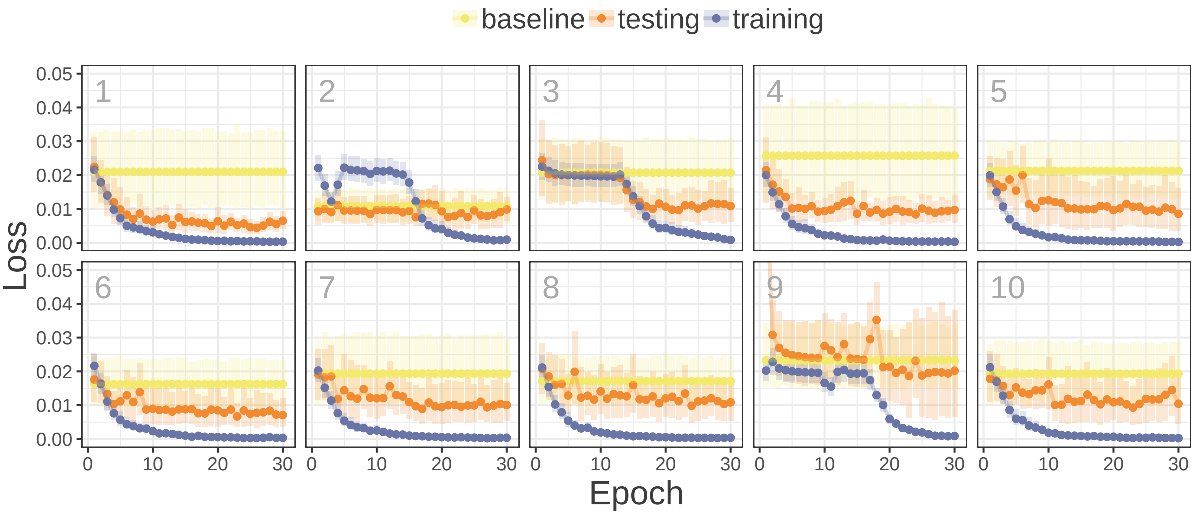

Figure 3: Loss (mean squared error) over epochs (x axis), faceted over cross-validation configurations. The performance of the model on the training set (in blue) approaches zero. The performance of the model on the dev-test set (in orange) generally outperforms the baseline (in yellow).

(blue). In addition, to provide context for the results, we include the loss for a dummy regressor that simply predict the mean of the training data labels for all cases.

The training loss approaches 0 toward the end of the training in all cross-validation configurations indicating model convergence. Crucially, dev-set performance is always subtantially btter than the baseline model, indicating that the model is indeed learning from the dataset. That said, there is a high amount of variation between the different cross-validation steps, with the baseline actually proving competitive in some folds. This seems an inevitable consequence of our small dataset, but the model clearly has gotten traction on the problem overall.

The MSE loss alone is not sufficient to assess how well the model actually learns to predict the underlying distribution. Figure 4 shows the correlation between the actual target labels (on the x axis) and the model predictions (on the y axis) for one of the cross-validation folds. Before training (left), the models are undifferentiated. After training (right), the model predictions (blue) are higly correlated with the true labels ($r = 0.85$), and the mean squared error is small (0.007).

Qualitatively, these plots appear very similar throughout all cross-validation configurations and can be compared in Figure 8 and 9 in the Appendix. Quantitatively the correlation on the dev-test set does show variation, driven mainly by outliers. However, overall, the mean correlation be-

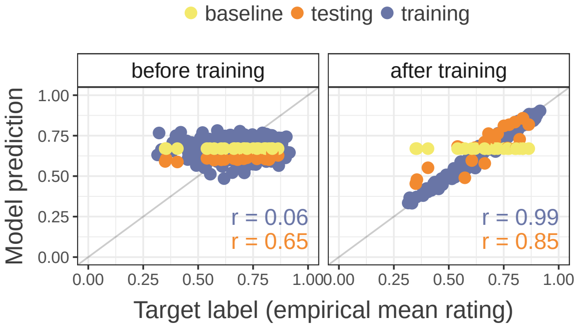

Figure 4: Correlation between target label (x axis) vs. model prediction (y axis) before and after training.

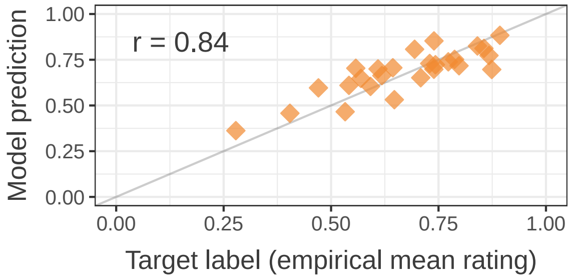

Figure 5: Testing label (x axis) vs. model prediction (y axis) after training on the held out test set (26 data points) using the parameter settings obtained after the 30th epoch from the first cross-validation. Pearson correlation of 84%.

tween the model prediction and the human judgment on the testing set across all cross-validation steps is 0.68. When we collapse over all cross-validation folds and examine the loss after training, the difference in loss between the model predictions and baseline is significant ($p < 0.0001$).

As a final performance evaluation, we examine the target–prediction correlation on the held-out test set (Figure 5). For predictions on the test set, we used the final model weights obtained by the first cross-validation configuration. This setup was chosen because of the low dev-set loss and the high correlation of the dev-set with the target labels. This was the only evaluation that was performed on this held-out test set. The Pearson correlation on the held-out test set is still high (0.84) and almost identical with performance on the dev-set. This high correlation, and the fact that the high correlation reproduces with the held-out testing data, indicates that the model was able to learn to generalize.

6 Model analysis

We have established that the proposed model can predict human guilt judgments when given a crime story. This result invites us to ask the question of whether there are patterns underlying these predictions that can provide higher-level insights into how langauge is construed in these criminal contexts.

6.1 Visualization

We begin by focusing attention on the learned attention weight vector $\mathbf{a_w}$ in equation (2). Since the softmax forces the sum of the weights to be 1, we cannot interpret the weights on their own for each word or across stories. Instead, the relevance lies in the differences between words and phrases within each story, and patterns of similarities between stories.

To investigate what might affect model predictions, we ran the model again on the final test data. Figure 6 displays three of these stories with their attention weight distribution. We see that the model seems to focus on phrases that explicitly describe uncertainty about the evidence (see Figure 6C). If present, they usually outweigh the rest of the story. This suggests that, if there is an explicit claim that affects the evidence of the suspect's guilt, it is considered as the most important source to inform guilt judgment.

Additionally, peaks occur on words and phrases which communicate turning points in a story, such as “however”, “but”, “even though”, and “it turns out”. This can be seen in all three stories in Figure 6. Those phrases could be relevant because they usually indicate a contrast to a prior narrative. Since those markers mostly follow reports of arrests, they might be strongly correlated with objections to those arrests which would in turn influence guilt perceptions.

Figure 6B shows a case where the model seems to find a simple declarative (“two boys destroyed”) to be relevant for the final prediction. This is especially interesting, since declaratives on their own do not generally communicate guilt-related information. However, they are very important for guilt judgments because they do not allow any uncertainty about the association between crime and suspect.

In summary, the visualization of the attention weights contribute further evidence that the model learns meaningful patterns in the data.

6.2 Qualitative analysis

The model visualization provides evidence for the claim that the model picks up on meaningful patterns in the corpus. But does the model make reasonable predictions on newly constructed data points? The original corpus started out with five different crime stories. Each of these stories occurred in two conditions – one suggesting that the evidence that led to the arrest was weak, and the other that the evidence provided a strong case. Additionally, the stories were filled with uncertainty markers such as “allegedly” or “(un)likely”, which we expect to have an impact on guilt judgments. Howver, the corpus cannot inform us about this relationship.

We can, though, use the model to predict a guilt judgment for each of the original stories without these uncertainty markers/hedges. Thus, we rewrote those 10 original stories into versions without any uncertainty markers. They remained as close to the original as possible, while still remaining grammatical. Figure 7 shows the results of this analysis. The results suggest that uncertainty markers have different effects on guilt prediction in the two evidence conditions. When the evidence is strong, removing all uncertainty markers does not affect the guilt judgments. However when the evidence is weak, removing those hedges

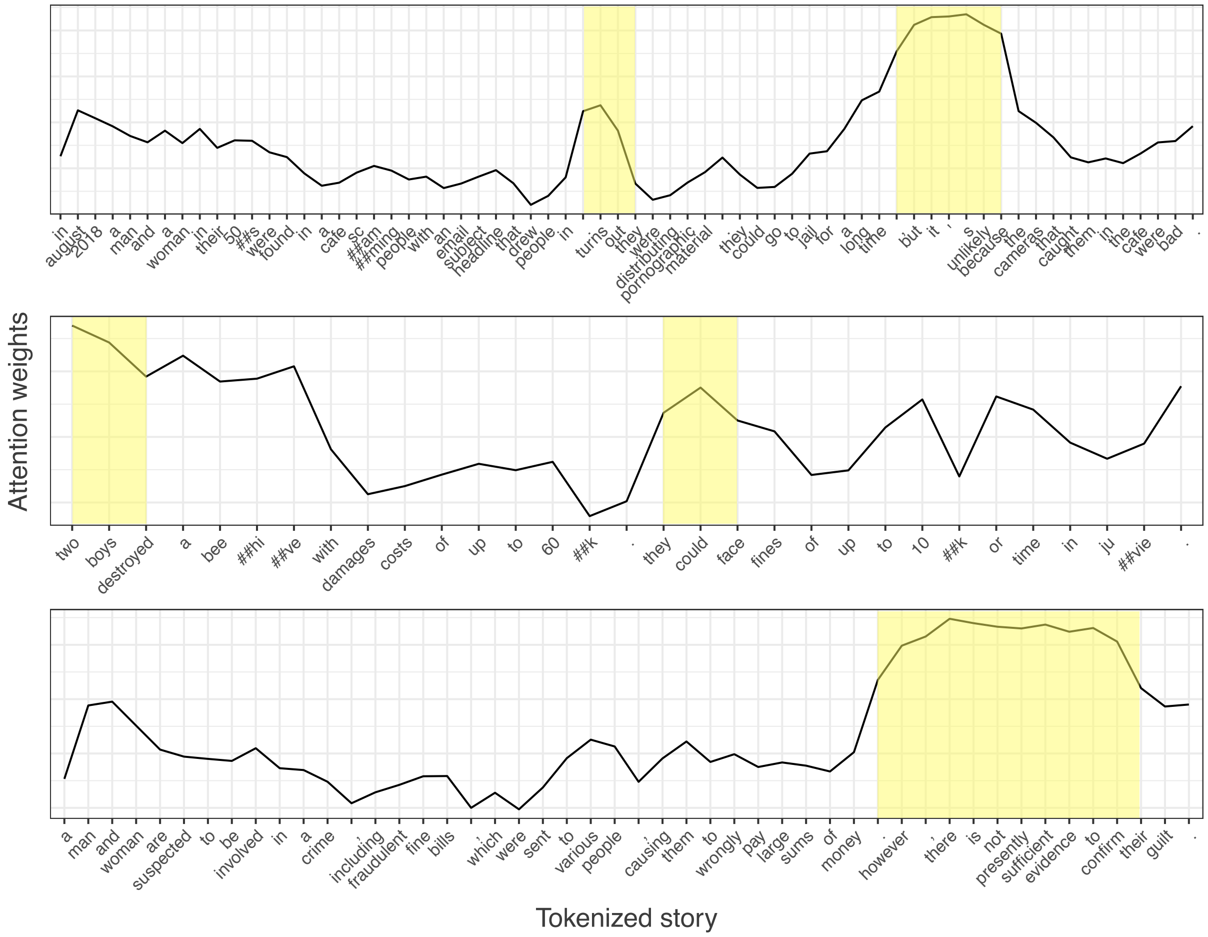

Figure 6: Visualization of the attention weights (y axis) for a tokenized story (x axis) from the test set. Because we took the softmax over the attention weights, the scale of the y axis is irrelevant. Areas of high attention weight are marked in yellow and correlate with markers of (un)certainty as well as markers of contrast.

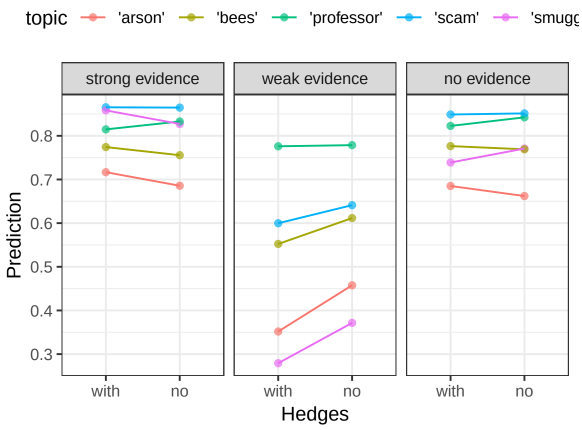

Figure 7: Model predictions on the relationship between uncertainty markers and evidence statements. [ek: fix caption and make figure more readable]

increases the guilt judgment. This has an intuitive interpretation: when evidence already overwhelmingly speaks for the suspect's guilt, it outweighs the hedges; when the evidence is questionable, other sources of uncertainty are considered to inform a final judgment.

Notably, if the evidence manipulation is excluded, the model predicts guilt judgments in the range of the strong evidence condition. In other words, the model assumes that additional justification for an arrest does not change the model's prediction of a guilty judgment. However, if reasons to question the arrest are given explicitly, this does affect the model's prediction. When we remove the hedges again, we cannot see a common structure to the change in ratings (possibly more similar to the strong evidence ratings though).[cp: I am not sure I understand this last sentence. Could it be expanded?]

7 Discussion

[ek: past tense!] The way a crime and arrest is presented in a news article affects how readers perceive the suspect's guilt. In this paper we showed that a recurrent neural network with self-attention can predict readers' guilt judgments. By visualizing the attention weights, we find that explicit evidence descriptions and phrases suggesting a turn of events influence predictions.

To gain linguistic insights into the neural network, we investigated its change in predictions when we exclude the hedges and information about the evidence from the stories. First of all the model makes similar predictions in the case where the evidence leading to an arrest is strong/convincing and the case where the evidence is not specified. [ek: One possible interpretation of this result is that the model learns that there is an implicature of strong evidence if the evidence is underspecified?] The model predictions are affected the most when the weakness of the evidence is mentioned explicitly.

Furthermore, we found that an exclusion of hedges in the stories only seems to affect predictions of stories in the weak evidence condition, where suspects are rated more guilty when there are no hedges. However, if the evidence is strong or underspecified, the model predictions do not seem to be affected. Possibly, in case of strongly perceived evidence, the model has learned that the hedges become irrelevant.

These results point to promising avenues for future investigations of hedging, particularly in the field of guilt perception.

[ek: how we perceive stories affects how we reproduce them; relevance for iterated chains]

[ek: Note on professor seed as outlier: Story about sexual harrassment allegations against male professor; only story that has clear gender split between victim(s) and suspect; Erickson 1978 show that powerless style (which includes hedges) affects credibility ratings dependent on whether witness is same or other sex]

[ek: Conclusion: propose a network that can predict guilt judgments and might be used to inform hypotheses for insight on linguistic features that determine guilt perception...]

References

Afra Alishahi, Grzegorz Chrupała, and Tal Linzen. 2019. Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop. *Journal of Natural Language Engineering*, 25(4):543–557.

Miltiadis Allamanis, Hao Peng, and Charles Sutton. 2016. A convolutional attention network for extreme summarization of source code. In *International Conference on Machine Learning*, pages 2091–2100.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations 2015*.

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

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Jean-François Bonnefon and Gaëlle Villejoubert. 2006. Tactful or doubtful? expectations of politeness explain the severity bias in the interpretation of probability phrases. *Psychological Science*, 17(9):747–751.

Penelope Brown, Stephen C Levinson, and Stephen C Levinson. 1987. *Politeness: Some universals in language usage*, volume 4. Cambridge university press.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert look at? an analysis of bert's attention. In *BlackBoxNLP@ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Association of Computational Linguistics*, Stroudsburg, PA. Association for Computational Linguistics.

Amanda M Durik, M Anne Britt, Rebecca Reynolds, and Jennifer Storey. 2008. The effects of hedges in persuasive arguments: A nuanced analysis of language. *Journal of Language and Social Psychology*, 27(3):217–234.

Bonnie Erickson, E Allan Lind, Bruce C Johnson, and William M O'Barr. 1978. Speech style and impression formation in a court setting: The effects of "powerful" and "powerless" speech. *Journal of Experimental Social Psychology*, 14(3):266–279.

Caitlin M Fausey and Lera Boroditsky. 2010. Subtle linguistic cues influence perceived blame and financial liability. *Psychonomic Bulletin & Review*, 17(5):644–650.

Scott Ferson, Jason O'Rawe, Andrei Antonenko, Jack Siegrist, James Mickley, Christian C Luhmann, Kari Sentz, and Adam M Finkel. 2015. Natural language of uncertainty: numeric hedge words. *International Journal of Approximate Reasoning*, 57:19–39.

Bruce Fraser. 2010. Pragmatic competence: The case of hedging. *New approaches to hedging*, 1(1):15–34.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 18th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Jakob D Jensen. 2008. Scientific uncertainty in news coverage of cancer research: Effects of hedging on scientists' and journalists' credibility. *Human communication research*, 34(3):347–369.

Kankawin Kowsrihawat, Peerapon Vateekul, and Prachya Boonkwan. 2018. Predicting judicial decisions of criminal cases from thai supreme court using bi-directional gru with attention mechanism. In *2018 5th Asian Conference on Defense Technology (ACDT)*, pages 50–55. IEEE.

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.

George Lakoff. 1972. Hedges: A study in meaning criteria and the logic of fuzzy concepts. In *Papers from the Eighth Regional Meeting of the Chicago Linguistic Society*, pages 183–228. Reprinted in Journal of Philosophical Logic, 1973, 2: 4, 458–508, and in D. Hockney et al. (eds.). Contemporary research in philosophical logic and linguistic semantics. Dodrecht: Fortis, 221–271.

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*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Jiangming Liu and Yue Zhang. 2017. Attention modeling for targeted sentiment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 572–577.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

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.

Ellen F Prince, Joel Frader, Charles Bosk, et al. 1982. On hedging in physician-physician discourse. *Linguistics and the Professions*, 8(1):83–97.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Victoria L. Rubin. 2007. Stating with certainty or stating with doubt: Intercoder reliability results for manual annotation of epistemically modalized statements. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 141–144, Rochester, New York. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Association for Computational Linguistics*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Michael Benedict L Virtucio, Jeffrey A Aborot, John Kevin C Abonita, Roxanne S Aviñante, Rother Jay B Copino, Michelle P Neverida, Vanesa O Osiana, Elmer C Peramo, Joanna G Syjuco, and Glenn Brian A Tan. 2018. Predicting decisions of the philippine supreme court using natural language processing and machine learning. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 130–135. IEEE.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. ArXiv:1609.08144.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272.

8 Appendices

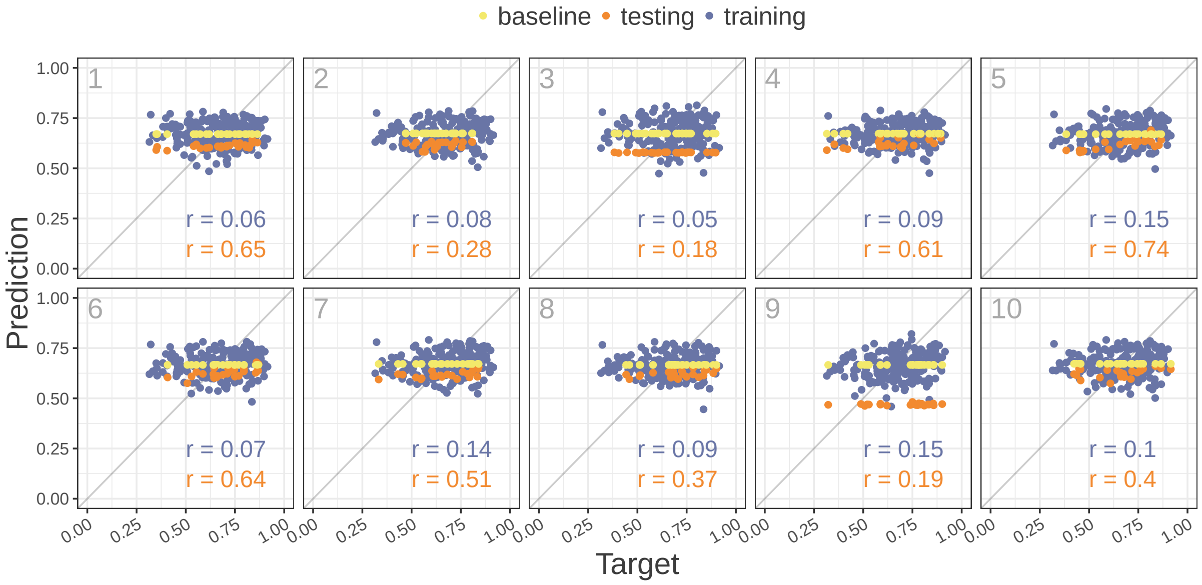

Figure 8: Testing label (x axis) vs. model prediction (y axis) before training; faceted over cross-validation configurations.

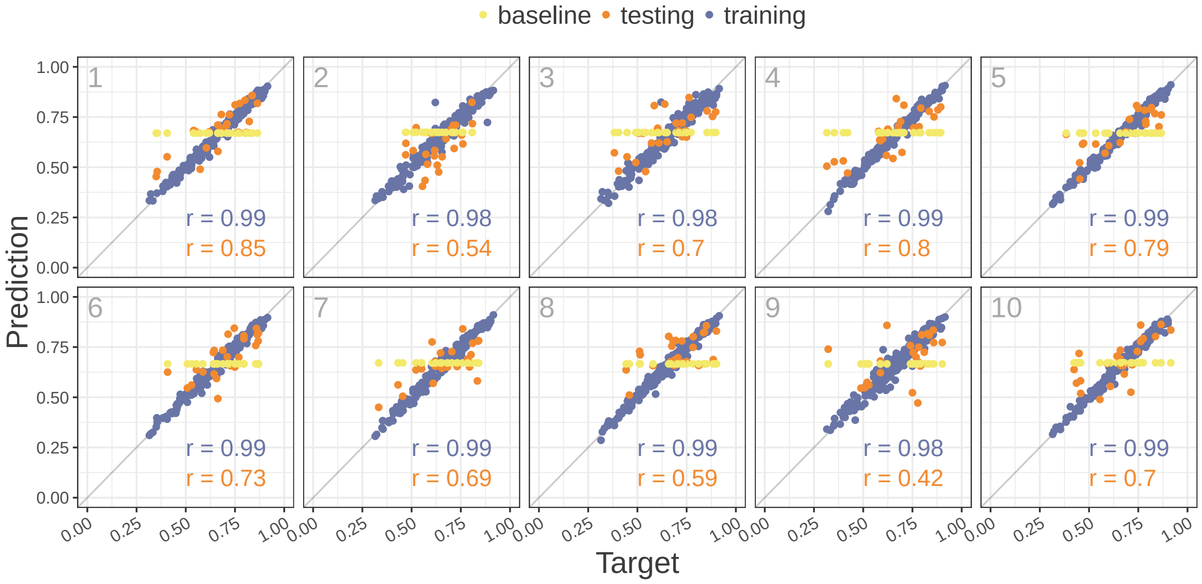

Figure 9: Testing label (x axis) vs. model prediction (y axis) after training; faceted over cross-validation configurations.