RESCIENCE C

Replication / ML Reproducibility Challenge 2020, Natural Language Processing

[Re] Reproducibility study - Does enforcing diversity in hidden states of LSTM-Attention models improve transparency?

Pieter Bouwman[1], Yun Li[1], Rogier van der Weerd[1], and Frank Verhoef[1]
[1]University of Amsterdam, Amsterdam, The Netherlands

Edited by
(Editor)

Received
–

Published
–

DOI
–

Reproducibility Summary

It has been shown [1] that the weights in attention mechanisms do not necessarily offer a faithful explanation of the model's predictions. In the paper Towards Transparent and Explainable Attention Models Mohankumar et al.[2] propose two methods to enhance faithfulness and plausibility of the explanations provided by an LSTM model combined with a basic attention mechanism.

Scope of Reproducibility — For this reproducibility study, we focus on the main claims made in this paper:

- The attention weights in standard LSTM attention models do not provide faithful and plausible explanations for its predictions. This is potentially because the conicity of the LSTM hidden vectors is high.
- Two methods can be applied to reduce conicity: Orthogonalization and Diversity Driven Training. When applying these methods, the resulting attention weights offer more faithful and plausible explanations of the model's predictions, without sacrificing model performance.

Methodology — The paper includes a link to a repository with the code used to generate its results. We follow four investigative routes: (i) Replication: we rerun experiments on datasets from the paper in order to replicate the results, and add the results that are missing in the paper; (ii) Code review: we scrutinize the code to validate its correctness; (iii) Evaluation methodology: we extend the set of evaluation metrics used in the paper with the LIME method, in an attempt to resolve inconclusive results; (iv) Generalization to other architectures: we test whether the authors' claims apply to variations of the base model (more complex forms of attention and a BiLSTM encoder).

Results — We confirm that the Orthogonal and Diversity LSTM achieve similar accuracies as the Vanilla LSTM, while lowering conicity. However, we cannot reproduce the results of several of the experiments in the paper that underlie their claim of better transparency. In addition, a close inspection of the code base reveals some potentially problematic inconsistencies. Despite this, under certain conditions, we do confirm that the Orthogonal

Correspondence should be addressed to Pieter Bouwman (pieter.bouwman@student.uva.nl)
The authors have declared that no competing interests exist.
Code is available at https://anonymous.4open.science/r/FACT_AI_project/..
Open peer review is available at https://openreview.net/forum?id=lE0wqKGROKa.

and Diversity LSTM can be useful methods to increase transparency. How to formulate these conditions more generally remains unclear and deserves further research. The single input sequence tasks appear to benefit most from the methods. For these tasks, the attention mechanism does not play a critical role for achieving performance.

What was easy/difficult — The codebase of the authors is accessible and can be run easily, with good facilities to prepare datasets and define configurations. The Orthogonalization and Diversity Training methods are well explained in the paper and mostly cleanly implemented. The larger datasets (Amazon and CNN) are difficult to run due to memory requirements and compute times. The codebase can be hard to navigate, a consequence of the choice to accommodate a large variation of models and datasets in one framework.

Communication with original authors — We reached out to the authors on a fundamental but unexplained choice in the model architecture but unfortunately did not hear back before the deadline of our assignment.

1 Introduction

The popularity of attention models has sparked many studies on the interpretability of the attention distributions, with often conflicting claims [1, 3, 4]. Mohankumar et al.[2] argue that the reason why attention weights do not always provide a faithful explanation of the model's predictions is that the learned hidden states of the LSTM based encoder are very similar across time steps, which is expressed by high conicity of these vectors. As a result, random permutation of the attention weights leads to a similar final context vector, which implies the weights do not provide a faithful explanation. The authors propose two methods that force the hidden states of the LSTM to be more diverse. Orthogonal LSTM ensures low conicity by orthogonalizing the hidden state at time t with respect to the mean of the previous hidden states. In Diversity LSTM the model is trained to jointly maximize the log-likelihood of the training data and to minimize the conicity of the hidden states.

2 Scope of reproducibility

In this reproducibility study we focus on the authors' main claim that the Diversity LSTM and Orthogonal LSTM lead to more faithful and plausible explanations, while maintaining accuracy of the predictions. The authors support their claim by evaluating a series of metrics that are assumed to be indicative of levels of faithfulness and plausibility. We follow four investigative routes:

- Replication: The main part of our study is focused on reproducing the results on the metrics in Mohankumar et al.[2], and to validate whether we can confirm their observations and conclusions. Furthermore, as the original paper only presents the results of a selection of models and datasets, we complement the results where possible. Most notably, we add results on the Orthogonal LSTM that were not in the original paper. Models, code and datasets are described in Section 3. Our replication results are presented in Section 4;
- Code review: As the authors' code[1] is publicly available, we use their code for the reproduction. In Section 5 we investigate whether the implementation is consistent with the description of the algorithms in the paper;
- Evaluation methodology: In Section 6 we report on our attempt to resolve inconclusive results we found on the attribution methods by extending the set of evaluation metrics used in the paper with the LIME method;
- Generalization to other architectures: In Section 7 we test whether the authors' claims apply to variations of the base model (more complex forms of attention and a BiLSTM encoder).

We conclude this paper in Section 8 with a discussion on the conditions under which the proposed methods are most likely to be effective, and a reflection on our replication study.

3 Methodology

Code — The code accompanying the paper is an extension based on the code[2] first developed by Jain and Wallace[1]. The entry point of the code is clear and well documented and allows a user to define specific jobs using command line arguments for hyperparameters. Preprocessing routines for the most datasets are included.

[1] https://github.com/akashkm99/Interpretable-Attention
[2] https://github.com/successar/AttentionExplanation

Datasets — We reran the experiments on 11 of the 14 datasets used in the paper. The nature and size of the datasets covers a wide range, from relatively simple binary sentiment classification tasks with single input sequence (abbreviated: SS) (e.g. SST with average input sentence length of 20 words), to complex question answering tasks with dual input sequences (abbreviated: DS)[3] (e.g. CNN with average document size of 760 words and an average of 26 answer categories). Some illustrations of data points can be found in Appendix D. The code repository includes links to the datasets, as well as the pre-processing routines used by the authors. We excluded the Amenia and Diabetes datasets because they were not accessible in time. The Amazon dataset caused memory issues when running the experiments. Despite these issues we were able to get the accuracies and conicity values for this dataset.

Model descriptions — The baseline model (Vanilla LSTM) used in the paper is shown in Figure 1. For DS tasks, it consists of two uni-directional LSTM encoders that act on a P-path (for document input phrases) and a Q-path (for question input phrases). When applied on SS tasks in the paper, only the P-path is used. An attention decoder is applied to the hidden states of the P-path LSTM to form the context vector $\mathbf{c}_\alpha$ on which the model calculates its output. The last hidden state of the Q-path is used as the query term for DS tasks.
The Diversity and Orthogonal LSTM that Mohankumar et al.[2] propose are variants of the baseline model. The Orthogonal LSTM applies an orthogonalization procedure to the LSTM hidden state vectors during training: the hidden state in timestep t is set to the component that is orthogonal to the mean of previous hidden states. This enforces low conicity of the hidden state vectors $\mathbf{h}_t^p$. The Diversity LSTM uses a standard LSTM cell with no explicit orthogonalization, but minimizes conicity jointly with the standard loss.

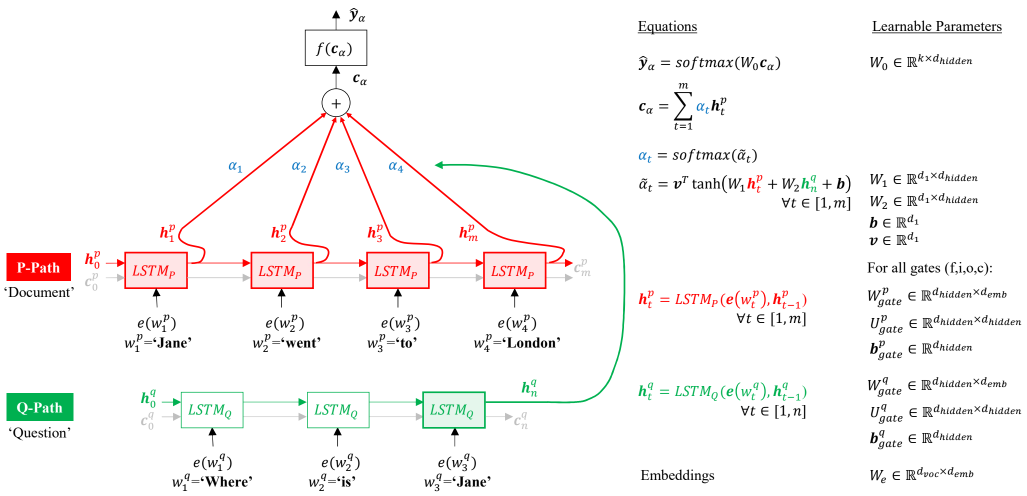

$\hat{y}_\alpha$
$f(\mathbf{c}_\alpha)$
$\mathbf{c}_\alpha$
+
α_1
α_2
α_3
α_4
h_1^p
h_2^p
h_3^p
h_m^p
P-Path
'Document'
h_0^p
c_0^p
$LSTM_P$
$LSTM_P$
$LSTM_P$
$LSTM_P$
c_m^p
$e(w_1^p)$
w_1^p='Jane'
$e(w_2^p)$
w_2^p='went'
$e(w_3^p)$
w_3^p='to'
$e(w_4^p)$
w_4^p='London'
Q-Path
'Question'
h_0^q
c_0^q
$LSTM_Q$
$LSTM_Q$
$LSTM_Q$
h_n^q
c_n^q
$e(w_1^q)$
w_1^q='Where'
$e(w_2^q)$
w_2^q='is'
$e(w_3^q)$
w_3^q='Jane'
Equations
Learnable Parameters
$\hat{\mathbf{y}}_\alpha = softmax(W_0 \mathbf{c}_\alpha)$
$W_0 \in \mathbb{R}^{k \times d_{hidden}}$
$\mathbf{c}_\alpha = \sum_{t=1}^{m} \alpha_t \mathbf{h}_t^p$
$\alpha_t = softmax(\tilde{\alpha}_t)$
$\tilde{\alpha}_t = \mathbf{v}^T \tanh(W_1 \mathbf{h}_t^p + W_2 \mathbf{h}_n^q + \mathbf{b}) \quad \forall t \in [1, m]$
$W_1 \in \mathbb{R}^{d_1 \times d_{hidden}}$
$W_2 \in \mathbb{R}^{d_1 \times d_{hidden}}$
$\mathbf{b} \in \mathbb{R}^{d_1}$
$\mathbf{v} \in \mathbb{R}^{d_1}$
For all gates (f,i,o,c):
$\mathbf{h}_t^p = LSTM_P(\mathbf{e}(w_t^p), \mathbf{h}_{t-1}^p) \quad \forall t \in [1, m]$
$W_{gate}^p \in \mathbb{R}^{d_{hidden} \times d_{emb}}$
$U_{gate}^p \in \mathbb{R}^{d_{hidden} \times d_{hidden}}$
$\mathbf{b}_{gate}^p \in \mathbb{R}^{d_{hidden}}$
$\mathbf{h}_t^q = LSTM_Q(\mathbf{e}(w_t^q), \mathbf{h}_{t-1}^q) \quad \forall t \in [1, n]$
$W_{gate}^q \in \mathbb{R}^{d_{hidden} \times d_{emb}}$
$U_{gate}^q \in \mathbb{R}^{d_{hidden} \times d_{hidden}}$
$\mathbf{b}_{gate}^q \in \mathbb{R}^{d_{hidden}}$
Embeddings
$W_e \in \mathbb{R}^{d_{voc} \times d_{emb}}$

Figure 1. The LSTM+attention model as defined in the paper

Hyperparameters — Given the wide variety of tasks and datasets, there is an elaborate set of model- and optimization hyperparameters. Not all parameter values are indicated in the original paper, some were retrieved by inspecting the code (an overview is presented in Appendix A). For all parameters, we used the defaults provided in the original code. We do not engage in further hyperparameter optimization to stay close to the original paper's approach. Note that we are interested in transparency and explainability of the models, not their optimal performance.

[3] These distinctions are differently named in the code: SS is referred to as BC and DS as QA

Experimental setup and computational requirements — We strictly follow the code environment as dictated by the requirements file that accompanies the code. All our experiments with this code are conducted on GPU nodes of the Lisa Cluster at SURFsara[4]. We had access to two Nvdia GTX1080Ti GPUs (11Gb VRAM).
Train and evaluation times varied between datasets and model variations, from ca. 5 minutes (SST dataset) to more than 40 hours (CNN dataset). We ran multiple seeds only on a selection of critical datasets to verify that differences we observed w.r.t. the results in the original paper were significant. Due to resource constraints, all other comparisons are based on single seeding, as was done in the original paper. This means that our observations are indicative, not conclusive.

4 Replication of the paper's results

4.1 Core replication results

Our reproduction study reveals numerous differences in results reported by Mohankumar et al.[2], for all datasets where we ran the experiments. Despite the differences, we support the observation that Diversity and Orthogonal LSTM reach similar accuracies as Vanilla LSTM, and lower conicity values, with the same exception reported in the paper (CNN). However, we find the claim that Diversity LSTM leads to more transparent attention distributions is not consistently supported. For Orthogonal LSTM, some results were omitted in the original paper, and we find conflicting results about the effect on faithfulness and plausibility. We present an overview of the comparisons by metric, and the impact our findings have on the main claims of the authors.

Accuracy and conicity — Of all accuracy and conicity values reported by Mohankumar et al.[2], we are able to reproduce 86% within a 3%-point margin. Models and datasets that produced the most notable differences are highlighted in Table 1. Despite the different values, the observation that Diversity and Orthogonal LSTM reach similar accuracies as Vanilla LSTM still holds, except for the CNN dataset. Also, we can confirm that conicity values are much lower in Diversity and Orthogonal LSTM, except for CNN in the Diversity LSTM. The largest difference in accuracy we observe for bAbI3, but the output files reveal that the model was not done training after the default 200 epochs.

Importance of hidden representation — Mohankumar et al.[2] analyse the importance of hidden representations using intermediate representation erasure [4] and also by examining the effect of permuting the attentions weights [1].
A visual comparison of the box plots about representation erasure in the paper with box plots in our reruns shows similar results in 25 of the 30 boxes. Despite the fact that our rerun shows lower medians for the box plots for the LSTM in IMDB and 20News dataset, the observation still holds that Diversity LSTM and Orthogonal LSTM reach a quicker decision flip for SS tasks. We concur with the authors' observations on the paraphrase detection (QQP) and Q&A task (bAbI1). In our rerun we see that the quick decision flip that is shown in bAbI1 also occurs in bAbI2 and bAbI3. Mohankumar et al.[2] do not report on SNLI and CNN, where our rerun shows no improvement of the Diversity LSTM and Orthogonal LSTM models over Vanilla LSTM.
The impact of permuting attention weights is difficult to compare with our results as Mohankumar et al.[2] only report a graphical representation (violin plots) of median output difference. After visual comparison we judge that the overall results are similar for IMDB, 20News and Yelp. We also evaluate the median output difference for datasets not reported by Mohankumar et al.[2]. We observe that the results for SST and Tweets show a similar 'shift to the right' as reported for other binary classification tasks. For DS tasks we

[4] https://userinfo.surfsara.nl/systems/lisa/description

Accuracy%

	Vanilla LSTM		Diversity LSTM		Orthogonal LSTM	
Dataset	original	rerun	original	rerun	original	rerun
SST	81.79	80.3	79.95	80.0	80.05	77.6
IMDB	89.49	89.3	88.54	87.8	88.71	88.3
Yelp	95.60	94.5	95.40	93.8	96.00	94.5
20News	93.55	90.8	91.03	90.8	92.15	91.9
Tweets	87.02	83.3	87.04	85.4	83.20	83.9
SNLI	78.23	77.3	76.96	74.0	76.46	76.6
QQP	78.74	78.4	78.40	78.2	78.61	78.6
bAbI1	99.10	100.0	100.00	100.0	99.90	99.9
bAbI2	40.10	54.4	40.20	54.6	56.10	59.0
bAbI3	47.70	21.1	50.90	56.3	51.20	57.7
CNN	63.07	59.5	58.19	46.3	54.30	53.6

Conicity

	Vanilla LSTM		Diversity LSTM		Orthogonal LSTM	
Dataset	original	rerun	original	rerun	original	rerun
SST	0.68	0.71	0.20	0.19	0.28	0.28
IMDB	0.69	0.60	0.08	0.09	0.18	0.16
Yelp	0.53	0.54	0.06	0.35	0.18	0.19
20News	0.77	0.76	0.15	0.14	0.23	0.24
Tweets	0.77	0.78	0.24	0.23	0.27	0.26
SNLI	0.56	0.59	0.12	0.04	0.27	0.31
QQP	0.59	0.58	0.04	0.03	0.33	0.32
bAbI1	0.56	0.77	0.07	0.07	0.22	0.23
bAbI2	0.48	0.43	0.05	0.13	0.21	0.17
bAbI3	0.43	0.93	0.10	0.11	0.12	0.13
CNN	0.45	0.40	0.06	0.38	0.07	0.10

Table 1. Comparison of reported accuracy and conicity values (differences > 0.03 are highlighted).

observe that Vanilla LSTM already has relatively high median output difference, and the Diversity LSTM and Orthogonal LSTM provide less improvement.
We conclude that in our experiments, the Diversity and Orthogonal LSTM do result in quicker decision flips and higher output difference for SS tasks, but not consistently for the other tasks.

Comparison with rationales — Our rerun of rationale length and rationale attention shows very different results as reported by Mohankumar et al.[2], see Table 2. Although we can confirm that Diversity LSTM results in shorter rationales, we cannot support the claim that Diversity LSTM provides much higher attention to the rationale than Vanilla LSTM. In our rerun this only holds for 20News.
The data for the Orthogonal LSTM, which were not reported by Mohankumar et al.[2], show much shorter rationale length, consistent with the paper's claim. However, impact on the share of attention on the rationale is mixed: it is higher for Yelp and 20 News, similar for IMDB and Tweets, but lower for SST.
For DS tasks, the rationale comparison is not implemented by the authors, we suspect because of the high computational costs involved for calculating rationales in tasks with multiple output categories.

Comparison with attribution methods — The rerun of the correlation metrics shows numerous differences in both Pearson correlation and JS Divergence. After studying Pearson correlation, we support the authors' claim that compared with Vanilla LSTM, Diversity LSTM produces attention weights that better correlate with gradients and integrated gradients, although in our results the relative increase of correlation with gradients is smaller: 13%[5] instead of the 65% reported by Mohankumar et al.[2]. However, we do not see the claimed reduction in JS Divergence. In fact, for all datasets the Diversity LSTM produces similar or even higher JS Divergence values than Vanilla LSTM, except JS Divergence with Integrated Gradients for 20News, see Table 3. The Orthogonal LSTM, for

[5] This percentage represents the average of the increases over all datasets.

Rationale attention

	Vanilla LSTM		Diversity LSTM		Orthogonal LSTM	
Dataset	original	rerun	original	rerun	original	rerun
SST	0.348	0.74	0.624	0.55	-	0.35
IMBD	0.472	0.97	0.761	0.91	-	0.92
Yelp	0.438	0.43	0.574	0.27	-	0.55
20News	0.627	0.62	0.884	0.94	-	0.86
Tweets	0.284	0.82	0.764	0.59	-	0.79

Rationale length

	Vanilla LSTM		Diversity LSTM		Orthogonal LSTM	
Dataset	original	rerun	original	rerun	original	rerun
SST	0.240	0.72	0.175	0.18	-	0.10
IMBD	0.217	0.92	0.169	0.22	-	0.27
Yelp	0.173	0.38	0.160	0.19	-	0.11
20News	0.215	0.59	0.173	0.27	-	0.24
Tweets	0.225	0.81	0.306	0.32	-	0.39

Table 2. Comparison of reported rationales (differences > 0.05 are highlighted)

which no correlation data is reported in the paper, is in line with the Diversity LSTM in this respect.

JS Divergence Gradients

	Vanilla LSTM		Diversity LSTM		Orthogonal LSTM	
	original	rerun	original	rerun	original	rerun
SST	0.10	0.09	0.08	0.09	-	0.14
IMDB	0.09	0.08	0.09	0.11	-	0.13
Yelp	0.15	0.12	0.13	0.17	-	0.16
20News	0.15	0.18	0.06	0.17	-	0.17
Tweets	0.08	0.07	0.12	0.09	-	0.18
SNLI	0.11	0.11	0.10	0.11	-	0.12
QQP	0.15	0.10	0.10	0.11	-	0.12
bAbI1	0.33	0.12	0.21	0.23	-	0.21
bAbI2	0.53	0.39	0.23	0.40	-	0.38
bAbI3	0.46	0.26	0.37	0.36	-	0.43
CNN	0.22	0.16	0.17	0.34	-	0.39

JS Divergence Integrated Gradients

	Vanilla LSTM		Diversity LSTM		Orthogonal LSTM	
Dataset	original	rerun	original	rerun	original	rerun
SST	0.12	0.10	0.09	0.10	-	0.15
IMDB	0.13	0.11	0.13	0.15	-	0.18
Yelp	0.19	0.18	0.19	0.19	-	0.17
20News	0.21	0.22	0.07	0.13	-	0.15
Tweets	0.08	0.08	0.15	0.10	-	0.19
SNLI	0.16	0.14	0.13	0.14	-	0.15
QQP	0.19	0.15	0.15	0.14	-	0.14
bAbI1	0.43	0.25	0.24	0.22	-	0.28
bAbI2	0.58	0.51	0.19	0.58	-	0.54
bAbI3	0.64	0.35	0.41	0.64	-	0.64
CNN	0.30	0.23	0.21	0.51	-	0.44

Table 3. Comparison of correlation metrics. Differences > 0.03 are highlighted

Analysis by POS tags — A comparison of the importance that is attributed to various POS tags shows similar importance and ranking for the SST, 20News and Tweets datasets. For Yelp and QQP we get different outcomes. Most notably, with vanilla LSTM model for Yelp we see no attention given to punctuations (PUNC), for which Mohankumar et al.[2] reports highest attention. For QQP, Mohankumar et al.[2] reports 23% on PUNC, while we find only 9%. Our results indicate the improvements shown in POS tags are less clear than reported by Mohankumar et al.[2].

Human evaluation — We could not reproduce the human evaluation within the four-week time frame of our research. Mohankumar et al.[2] reports convincing results, and we also believe human interpretation should play a key role in judging whether their methods improve transparency. We include some examples in Appendix D for this purpose.

4.2 Conclusion regarding reproducibility

Our findings are summarized in Table 4. We conclude that it is not immediately clear that Diversity LSTM and Orthogonal LSTM provide better transparency for all the studied datasets.

Metric	Claim (with reference to paragraph number in Mohankumar et al.[2])	Supported after rerun	Notes
Accuracy and conicity	Diversity LSTM and Orthogonal LSTM achieve similar accuracies as Vanilla LSTM, but much lower conicity (§5.2)	Yes	Except CNN in Diversity LSTM
Fraction of hidden representation required for decision flip (box plots)	Diversity LSTM and Orthogonal LSTM reach quicker decision flip (§5.3)	Yes (for models in paper)	Especially for BC tasks, somewhat for QQP; not for SNLI and QA tasks
Median Output difference on randomly permuting attention weights (violin charts)	Diversity LSTM and Orthogonal LSTM are more sensitive to random permutation of weights than Vanilla LSTM (§5.3)	Yes (for models in paper)	Clear difference for BC tasks, mixed picture for the dual-sequence tasks.
Rationale attention	Diversity LSTM provides much higher attention to rationales than Vanilla LSTM across the 8 Text classification datasets (§5.4)	No	Only true for 20News; No results reported on QA tasks
Rationale length	Diversity LSTM often provides shorter rationales than Vanilla LSTM (§5.4)	Yes	No results reported on QA tasks
Pearson correlation and JS divergence between distribution of attention and (integrated) gradients	Attention weights in Diversity LSTM better agree with gradients and integrated gradients than Vanilla LSTM (§5.5)	Mixed	Diversity LSTM has higher Pearson correlation, but similar or higher JS Divergence
Attention given to POS tags	Attention given to punctuation marks is significantly reduced on the Yelp, Amazon and QQP datasets (§5.6)	No	Not for Yelp, less clear for QQP
	Diversity LSTM gives much more attention to adjectives than Vanilla LSTM in the four sentiment analysis tasks (SST, IMDB, Yelp, Amazon) (§5.6)	Yes	True for SST and IMDB, but not for Yelp
Human evaluation of plausibility	Human evaluators prefer attention distribution of Diversity LSTM over Vanilla LSTM for Yelp, SNLI, QQP and bAbI1 (§5.7)	Not reproduced	Evaluation by only 15 people

Table 4. Evidence for authors' claims after rerun

- The Orthogonal LSTM clearly leads to lower conicity than Vanilla LSTM, but Mohankumar et al.[2] show little evidence with other metrics that indicate higher faithfulness: of the 14 datasets, only 6 boxplots and 4 violin charts are included. The results observed in our rerun are mixed. For example, Orthogonal LSTM works well for 20News, but for SNLI there is hardly any effect on the box plot, and also correlation/JSD with (integrated) gradients is worse.
- For Diversity LSTM, Mohankumar et al.[2] show convincing evidence with substantial data. We observe similar trends in conicity, and the impact of diversity training is clear in the box plots and violin charts for the binary classification tasks. However, for the tasks that require two input sequences like SNLI, bAbI2, CNN our rerun shows that Diversity LSTM does not contribute much to faithfulness and can lead to lower correlation with (integrated) gradients and higher JS Divergence.

5 Code Review

As part of the reproduction study, we familiarized ourselves with the code to understand how the model and the experiments had been implemented. We also scrutinized the code to check whether we could find a cause for the differences we found in the reported metrics. The code's class architecture can accommodate a wide range of tasks, datasets and model configurations. While convenient, this also makes the codebase complex and susceptible to errors. The code review revealed several debatable choices, of which the main ones are described below.

Orthogonalization of Q-path in dual input sequence tasks — For DS tasks, we expect the orthogonalization procedure to only be activated in the P-path (the path of the input document) of the model, as this is the path on which the attention mechanism applies its weights α_t. However, in the code, orthogonalization is also applied to the Q-path (the path of the question phrase in the Q&A tasks, or the second input phrase in SNLI and QQP). In our view, this introduces a potentially problematic effect. The attention mechanism uses only the last hidden state vector $\mathbf{h}_t^q$ as the query term. This representation for the last word in the sequence will only retain the vector component orthogonal to the mean of the previous word representations, as a result of orthogonalization. We argue that the direction of $\mathbf{h}_t^q$ in the hidden space will represent the exclusive 'change of meaning' that the last word adds to the sequence. This is not a problem in the bAbI tasks, where the prompt word in the question phrase is always the last word (e.g., 'Where is Jane'). But for longer questions where the prompt words appear earlier in the question, this may impede the attention mechanism from finding the right prompt words.
In order to test this sensitivity, we conduct an experiment for the simpler SS tasks. We apply orthogonalization during training and compare model performance when i) attention weights are left unconstrained vs. ii) all attention weights are set to zero, except for the last hidden state. The result is shown in Table 5.

	Vanilla LSTM		Orthogonal LSTM	
Dataset	Base attention	last_only attention	Base attention	last_only attention
SST: accuracy	0.803	0.810	0.776	0.583
(conicity)	(0.713)	(0.763)	(0.283)	(0.265)
IMDB: accuracy	0.893	0.876	0.883	0.784
(conicity)	(0.602)	(0.885)	(0.163)	(.141)
20News: accuracy	0.908	0.857	0.919	0.583
(conicity)	(0.761)	(0.831)	(0.235)	(0.395)
Tweets: accuracy	0.833	0.782	0.839	0.712
(conicity)	(0.776)	(0.798)	(0.260)	(0.330)

Table 5. Demonstration of adverse effect of orthogonalization on the information content of the last hidden state vector (results reflect our experiments, not the original paper)

What is striking is the performance remains on par (marked in green) for Vanilla LSTM when only attending to the last hidden state, indicating the model performs well without the attention mechanism. However, we observe a performance drop of 10%-34% (absolute) when attention is constrained for the Orthogonal LSTM (marked in red). Indeed, it appears part of the information required for inference is lost. How this effect impacts the results requires further study. It may explain the accuracy drop from 63% (Vanilla LSTM) to 58%/54%(Diversity/Orthogonal LSTM) for CNN as reported in Table 2 by Mohankumar et al.[2]. We have contacted the authors to verify their intentions, but did not receive a response prior to submission of this reproduction study.

Disparate calculation of final prediction — For DS tasks, in the code the final prediction layer is implemented as $\hat{y} = \mathrm{softmax}(\mathrm{W}_r(\tanh(\mathrm{W}_p\mathbf{c}_\alpha + \mathbf{b}_p + \mathrm{W}_q\mathbf{h}_n^q + \mathbf{b}_q)) + \mathbf{b}_r)$. This deviates from the prediction function $\hat{y} = \mathrm{softmax}(\mathrm{W}_0\mathbf{c}_\alpha)$ described in Section 2.1 by Mohankumar et al.[2][6]. However, this does not affect the core architecture, namely LSTM and attention, so we did not modify the code or conduct further experiments.

Fine-tuning of embeddings — The models use pre-trained embeddings except for the bAbI datasets. Words outside of the pre-trained embeddings' vocabulary are initialized with zero-vectors. All embeddings are fine-tuned (i.e. trainable), independently for the P- and Q-paths for DS tasks. This is not mentioned in the original paper and this choice is

[6] https://github.com/akashkm99/Interpretable-Attention/blob/master/model/modules/Decoder.py#L101-L107

questionable as it leads to an excessive number of trainable parameters (e.g., >40M for the CNN dataset, see Appendix A) and training time, while it is unlikely to be critical for the tasks.

Definition of dev set for bAbI datasets — While pre-processing bAbI datasets, 15% of the train set is randomly selected to be used as dev set, resulting in much higher similarity between these two splits compared to the test set. As a result, the trained model is overfit on the train set, and we observe a large gap between dev and test accuracy.

6 Extension of the evaluation methods

As discussed in Section 4.1.4, our rerun of Pearson's correlation and JS Divergence between attention weights and gradients/integrated gradients points towards a less convincing conclusion. We therefore also used the LIME framework [5] as a third metric for comparing how transparent the attention weights are as explanations, as well as how much improvements are brought about by the Diversity and Orthogonal LSTM.

	Pearson's Correlation			JS Divergence		
Dataset	Vanilla	Ortho.	Div.	Vanilla	Ortho.	Div.
IMDB	0.42	0.33	0.42	0.26	0.44	0.42
20News	0.30	0.70	0.71	0.22	0.42	0.45
Tweets	0.13	0.38	0.43	0.07	0.33	0.18
SNLI	0.24	0.22	0.23	0.15	0.15	0.12
bAbI1	0.69	0.67	0.58	0.42	0.38	0.46

Numbers that agree with expectations (higher correlation, lower JS Divergence) are highlighted in green, numbers opposite to expectations are highlighted in red.

Table 6. Correlation and JS Divergence between attention weights and LIME scores

We use LIME to generate a score for the predicted class on each word-position in the sentence, which can then be compared with the attention weights. For calculating JS divergence we also rescaled the lime score so that the scores range from 0 to 1, and sums to 1 per sentence (i.e. similar to attention scores). The results are shown in Table 6, where we experimented with only a representative selection of datasets due to time and resource constraints.

Similar to our comparison of attention weights with gradient-based methods, Table 6 indicates Diversity and Orthogonal LSTM fail to produce explanations consistent with LIME. It is also not clear which statistical measure is best for comparing whether two explanation methods agree with each other. In several instances (e.g. 20News and Tweets), we observe an increase in Pearson's correlation and an increase in JS Divergence at the same time when going from Vanilla LSTM to Orthogonal/Diversity LSTM models.

7 Generalization to other model architectures

Despite the differences we found between our observations and the observations reported by [2], we still see the potential value of the methods they propose. This is because we did observe sparser attention weights when using Diversity and Orthogonal LSTM, and because of the strong preference expressed for the Diversity LSTM in the human evaluations conducted by Mohankumar et al.[2].

We therefore investigate how well these methods work in alternative settings. So far the Orthogonalization and Diversity Training methods are only tested on one-layer unidirectional LSTM models with attention. However, in many recent studies, BiLSTM-based attention models or Transformer models are used [6, 7, 8]. Similarly, more complex attention mechanisms such as self-attention and multi-head attention [9] gained popularity due to their superior performance. For this reason, we investigate whether the proposed methods can be extended to more complex models and whether the authors' two main claims still apply.

Extending to other attention mechanisms — The application of more advanced attention mechanisms (such as multi-head attention) poses a challenge because they produce more than one attention weight per word. It is not straightforward to generate explanations and word importance based on these weights. As a consequence, several of the evaluation metrics used by the authors cannot be applied in their current form. This would require making non-trivial design choices on how to combine multiple distributions of the attention weights. Further research is required to investigate this and whether existing methods such as Attention Flow and Attention Rollout [10] can provide a resolution.

Extending to other architectures: BiLSTM Experiments — We replace the uni-directional LSTM in the model (Figure 1) with a bi-directional LSTM. We choose the BiLSTM architecture, and not a Transformer based architecture, as the latter requires dealing with the more advanced attention mechanisms discussed above.
In order to maintain the decoder's complexity (the attention mechanism), we preserve the output dimension of the LSTM. This requires halving the dimension of the hidden states, which also ensures that the number of trainable weights of the BiLSTM is comparable to that of the unidirectional LSTM. For the Diversity BiLSTM, the same diversity weights are used as in Mohankumar et al.[2]. The conicity term present in the loss function of the Diversity BiLSTM is calculated based on the concatenated forward and backward hidden representations. Orthogonalization for the Ortho BiLSTM is applied before concatenation of the forward and backward hidden states.
Results show that the application of the two methods proposed by Mohankumar et al.[2] do not result in performance loss and do lower conicity. However, on other metrics and across datasets, the picture is mixed like we saw in our reproducibility results for the unidirectional LSTM, indicating the methods do not unconditionally improve explanations. We will not discuss these results in detail, but conclude that it is indeed possible to extend the proposed methods to BiLSTM attention models. Full results are included in Appendix C for completeness.

8 Discussion

Our reproduction shows that enforcing low conicity between the hidden states of an LSTM encoder does not guarantee improved transparency in the studied datasets, at least not on the metrics used by Mohankumar et al.[2]. We find the authors' claim about improved transparency not generally applicable and under certain conditions their methods even hurt accuracy. Still, the Diversity LSTM and Orthogonal LSTM do lead to improved metrics on some datasets, and the human evaluation Mohankumar et al.[2] conducted shows strong preference for the Diversity LSTM over Vanilla LSTM. This raises the question under what conditions these methods should be applied.

Conditions underlying effectiveness — One pattern that seems to emerge is that the benefits of orthogonalizing or diversity training are most apparent for the relatively simpler SS tasks. The potential to improve faithfulness of the weights might be high in those cases as it not a given that attention weights carry any meaning for these task.
For some tasks, the LSTM does not strictly need the attention mechanism to perform well, as is shown in Table 7 when the attention mechanism is constrained to be either uniform or attending to the last word only. In contrast, the more difficult DS tasks do require the attention mechanism in order to reach higher accuracies. This pattern is similar to that described by Wiegreffe and Pinter[3].

	Base attention		Constrained attention	
Dataset	Reported	Repr.	uniform	last_only
SST	.818	.803	.800	.810
IMDB	.895	.893	.883	.876
Yelp	.956	.949	.950	.949
20News	.936	.908	.898	.857
Tweets	.870	.833	.833	.782
SNLI	.782	.773	.755	.759
QQP	.787	.784	.789	.792
bAbI1	.991	1.00	.485	.729
bAbI2	.401	.544	.315	.441
bAbI3	.477	.211*	-	-
CNN	.631	.595	.424	.367

* Reproduction failed, comparisons not applicable

Table 7. Impact on performance of the Vanilla LSTM when forcing uniform, first- and last only attention

We suspect that there is a relation between a) how crucial the attention mechanism is for performance in a given task, b) how much improvement Orthogonal/Diversity LSTM can offer w.r.t. plausibility of the attention weights for explaining the model's outputs. This relationship, and the conditions under which orthogonalization and diversity training offer the best results, deserves additional investigation.

Reflection on our replication study — A key insight we have gained is that even with access to the original code, exact reproduction of the results is not guaranteed. We have not been able to find the cause of several differences in results. The available time and hardware limited our possibilities to repeat these experiments with multiple seeds to find an estimate of the variance of outcomes.
Another insight we gained is that the metrics concerning faithfulness and plausibility can be hard to interpret, as it is deeply entangled with the nature of the dataset as well as the model implementation. To enable scalable development of transparent AI models, reliable quantitative metrics are needed that can accurately approximate real humans' judgement. We believe further development of transparency metrics is an important area for further research to help build more transparent models.

Acknowledgement

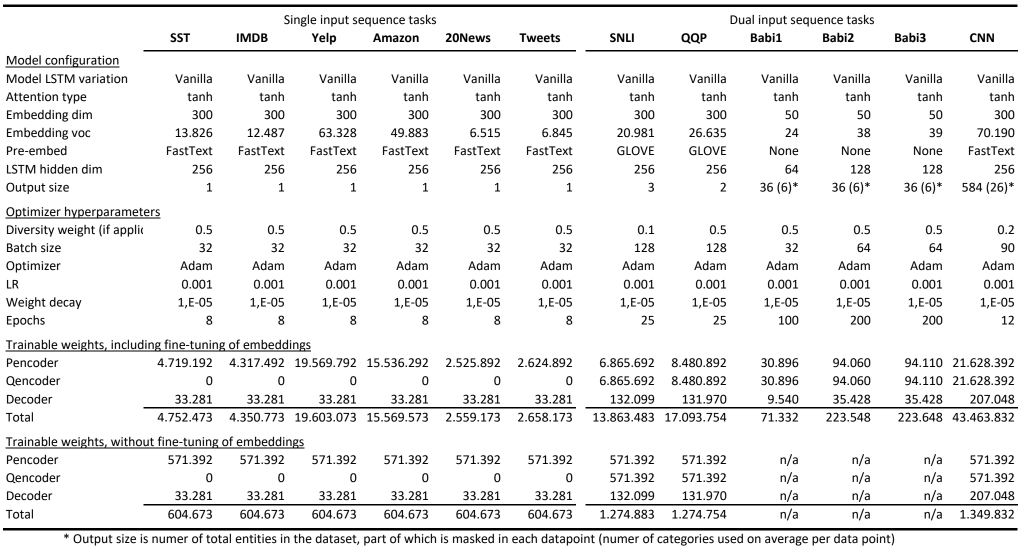
We'd like to thank Stefan Schouten for his guidance and insightful discussions.

References

1. S. Jain and B. C. Wallace. "Attention is not Explanation." In: **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3543–3556. DOI: 10.18653/v1/N19-1357. URL: https://www.aclweb.org/anthology/N19-1357.
2. A. K. Mohankumar, P. Nema, S. Narasimhan, M. M. Khapra, B. V. Srinivasan, and B. Ravindran. "Towards Transparent and Explainable Attention Models." In: **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**. Online: Association for Computational Linguistics, July 2020, pp. 4206–4216. DOI: 10.18653/v1/2020.acl-main.387. URL: https://www.aclweb.org/anthology/2020.acl-main.387.
3. S. Wiegreffe and Y. Pinter. "Attention is not not Explanation." In: **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 11–20. DOI: 10.18653/v1/D19-1002. URL: https://www.aclweb.org/anthology/D19-1002.
4. S. Serrano and N. A. Smith. "Is Attention Interpretable?" In: **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 2931–2951. DOI: 10.18653/v1/P19-1282. URL: https://www.aclweb.org/anthology/P19-1282.

5. M. T. Ribeiro, S. Singh, and C. Guestrin. "" Why should I trust you?" Explaining the predictions of any classifier." In: **Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining**. 2016, pp. 1135–1144.
6. Q. Zhou and H. Wu. "NLP at IEST 2018: BiLSTM-attention and LSTM-attention via soft voting in emotion classification." In: **Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis**. 2018, pp. 189–194.
7. L.-H. Lee, Y. Lu, P.-H. Chen, P.-L. Lee, and K.-K. Shyu. "NCUEE at MEDIQA 2019: medical text inference using ensemble BERT-BiLSTM-Attention model." In: **Proceedings of the 18th BioNLP Workshop and Shared Task**. 2019, pp. 528–532.
8. A. Aziz Sharfuddin, M. Nafis Tihami, and M. Saiful Islam. "A Deep Recurrent Neural Network with BiLSTM model for Sentiment Classification." In: **2018 International Conference on Bangla Speech and Language Processing (ICBSLP)**. 2018, pp. 1–4. DOI: 10.1109/ICBSLP.2018.8554396.
9. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need." In: **Proceedings of the 31st International Conference on Neural Information Processing Systems**. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 6000–6010.
10. S. Abnar and W. Zuidema. "Quantifying Attention Flow in Transformers." In: **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**. Online: Association for Computational Linguistics, July 2020, pp. 4190–4197. DOI: 10.18653/v1/2020.acl-main.385. URL: https://www.aclweb.org/anthology/2020.acl-main.385.

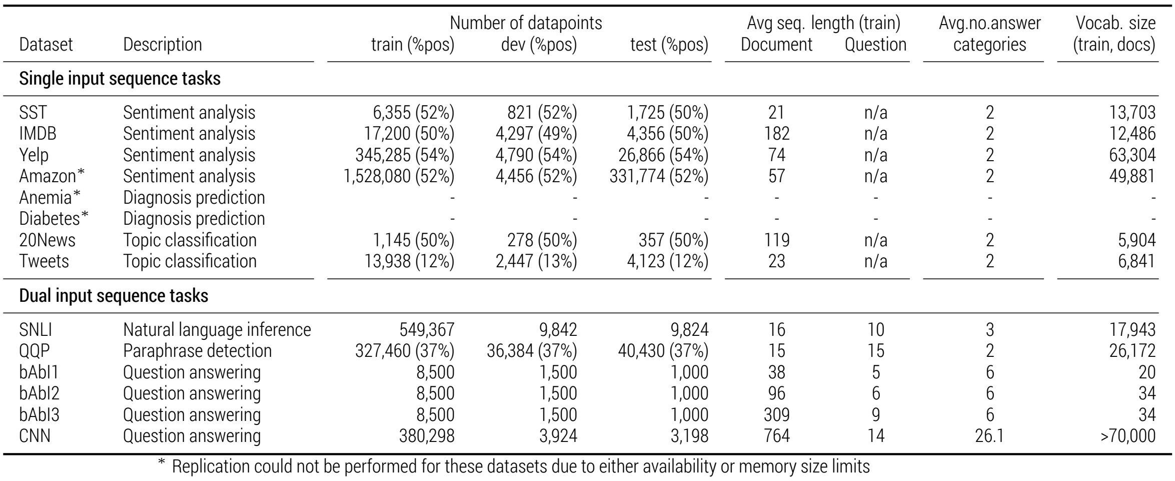
Appendix A: Details of models and datasets



	Single input sequence tasks						Dual input sequence tasks					
	SST	**IMDB**	**Yelp**	**Amazon**	**20News**	**Tweets**	**SNLI**	**QQP**	**Babi1**	**Babi2**	**Babi3**	**CNN**
Model configuration												
Model LSTM variation	Vanilla	Vanilla	Vanilla	Vanilla	Vanilla	Vanilla	Vanilla	Vanilla	Vanilla	Vanilla	Vanilla	Vanilla
Attention type	tanh	tanh	tanh	tanh	tanh	tanh	tanh	tanh	tanh	tanh	tanh	tanh
Embedding dim	300	300	300	300	300	300	300	300	50	50	50	300
Embedding voc	13.826	12.487	63.328	49.883	6.515	6.845	20.981	26.635	24	38	39	70.190
Pre-embed	FastText	FastText	FastText	FastText	FastText	FastText	GLOVE	GLOVE	None	None	None	FastText
LSTM hidden dim	256	256	256	256	256	256	256	256	64	128	128	256
Output size	1	1	1	1	1	1	3	2	36 (6)*	36 (6)*	36 (6)*	584 (26)*
Optimizer hyperparameters												
Diversity weight (if applic	0.5	0.5	0.5	0.5	0.5	0.5	0.1	0.5	0.5	0.5	0.5	0.2
Batch size	32	32	32	32	32	32	128	128	32	64	64	90
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam	Adam	Adam	Adam	Adam	Adam
LR	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Weight decay	1,E-05	1,E-05	1,E-05	1,E-05	1,E-05	1,E-05	1,E-05	1,E-05	1,E-05	1,E-05	1,E-05	1,E-05
Epochs	8	8	8	8	8	8	25	25	100	200	200	12
Trainable weights, including fine-tuning of embeddings												
Pencoder	4.719.192	4.317.492	19.569.792	15.536.292	2.525.892	2.624.892	6.865.692	8.480.892	30.896	94.060	94.110	21.628.392
Qencoder	0	0	0	0	0	0	6.865.692	8.480.892	30.896	94.060	94.110	21.628.392
Decoder	33.281	33.281	33.281	33.281	33.281	33.281	132.099	131.970	9.540	35.428	35.428	207.048
Total	4.752.473	4.350.773	19.603.073	15.569.573	2.559.173	2.658.173	13.863.483	17.093.754	71.332	223.548	223.648	43.463.832
Trainable weights, without fine-tuning of embeddings												
Pencoder	571.392	571.392	571.392	571.392	571.392	571.392	571.392	571.392	n/a	n/a	n/a	571.392
Qencoder	0	0	0	0	0	0	571.392	571.392	n/a	n/a	n/a	571.392
Decoder	33.281	33.281	33.281	33.281	33.281	33.281	132.099	131.970	n/a	n/a	n/a	207.048
Total	604.673	604.673	604.673	604.673	604.673	604.673	1.274.883	1.274.754	n/a	n/a	n/a	1.349.832

* Output size is numer of total entities in the dataset, part of which is masked in each datapoint (numer of categories used on average per data point)

Table 8. Model- and hyperparameters for standard configurations per dataset

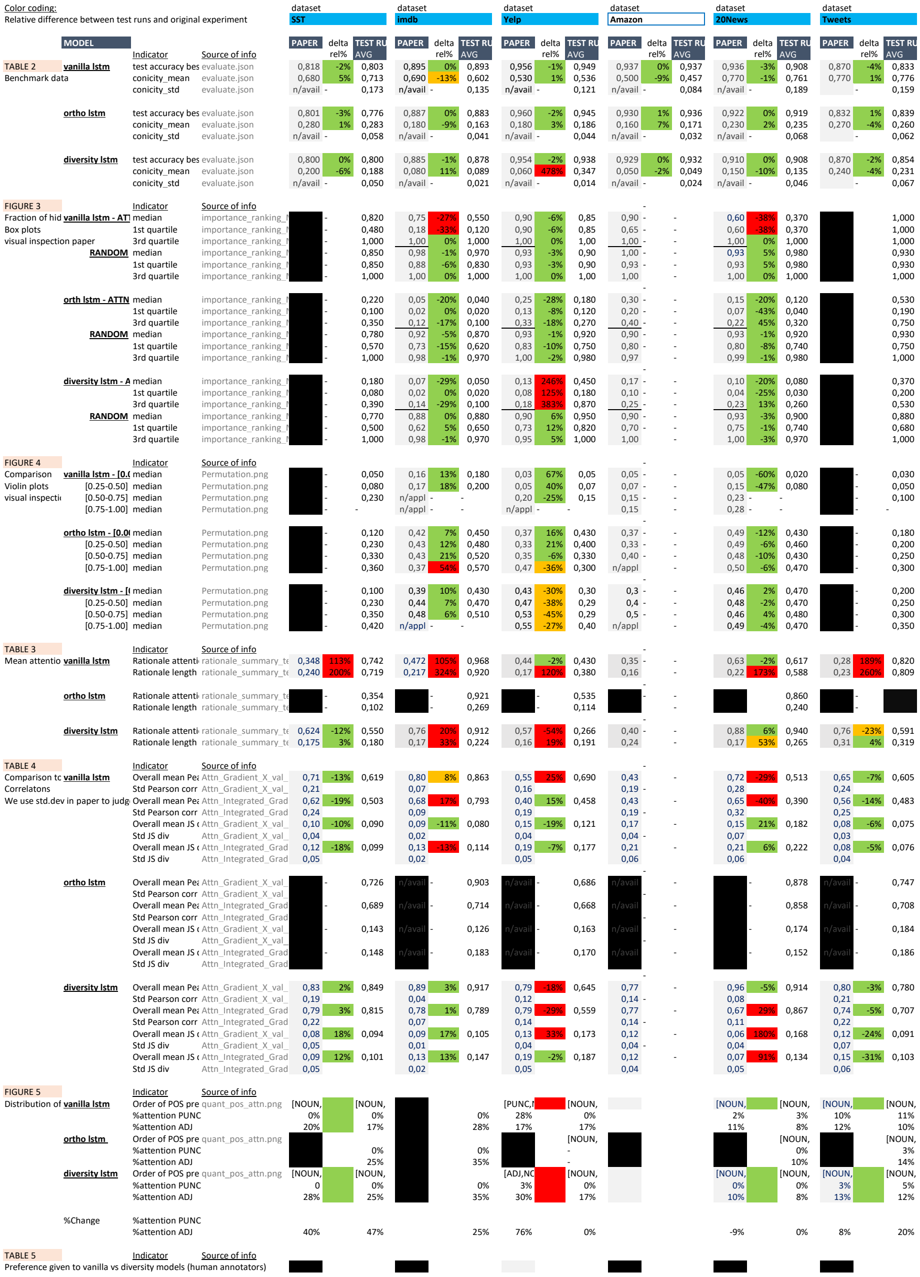


Dataset	Description	Number of datapoints train (%pos)	dev (%pos)	test (%pos)	Avg seq. length (train) Document	Question	Avg.no.answer categories	Vocab. size (train, docs)
Single input sequence tasks								
SST	Sentiment analysis	6,355 (52%)	821 (52%)	1,725 (50%)	21	n/a	2	13,703
IMDB	Sentiment analysis	17,200 (50%)	4,297 (49%)	4,356 (50%)	182	n/a	2	12,486
Yelp	Sentiment analysis	345,285 (54%)	4,790 (54%)	26,866 (54%)	74	n/a	2	63,304
Amazon*	Sentiment analysis	1,528,080 (52%)	4,456 (52%)	331,774 (52%)	57	n/a	2	49,881
Anemia*	Diagnosis prediction	-	-	-	-	-	-	-
Diabetes*	Diagnosis prediction	-	-	-	-	-	-	-
20News	Topic classification	1,145 (50%)	278 (50%)	357 (50%)	119	n/a	2	5,904
Tweets	Topic classification	13,938 (12%)	2,447 (13%)	4,123 (12%)	23	n/a	2	6,841
Dual input sequence tasks								
SNLI	Natural language inference	549,367	9,842	9,824	16	10	3	17,943
QQP	Paraphrase detection	327,460 (37%)	36,384 (37%)	40,430 (37%)	15	15	2	26,172
bAbI1	Question answering	8,500	1,500	1,000	38	5	6	20
bAbI2	Question answering	8,500	1,500	1,000	96	6	6	34
bAbI3	Question answering	8,500	1,500	1,000	309	9	6	34
CNN	Question answering	380,298	3,924	3,198	764	14	26.1	>70,000

* Replication could not be performed for these datasets due to either availability or memory size limits

Table 9. Characteristics of the datasets

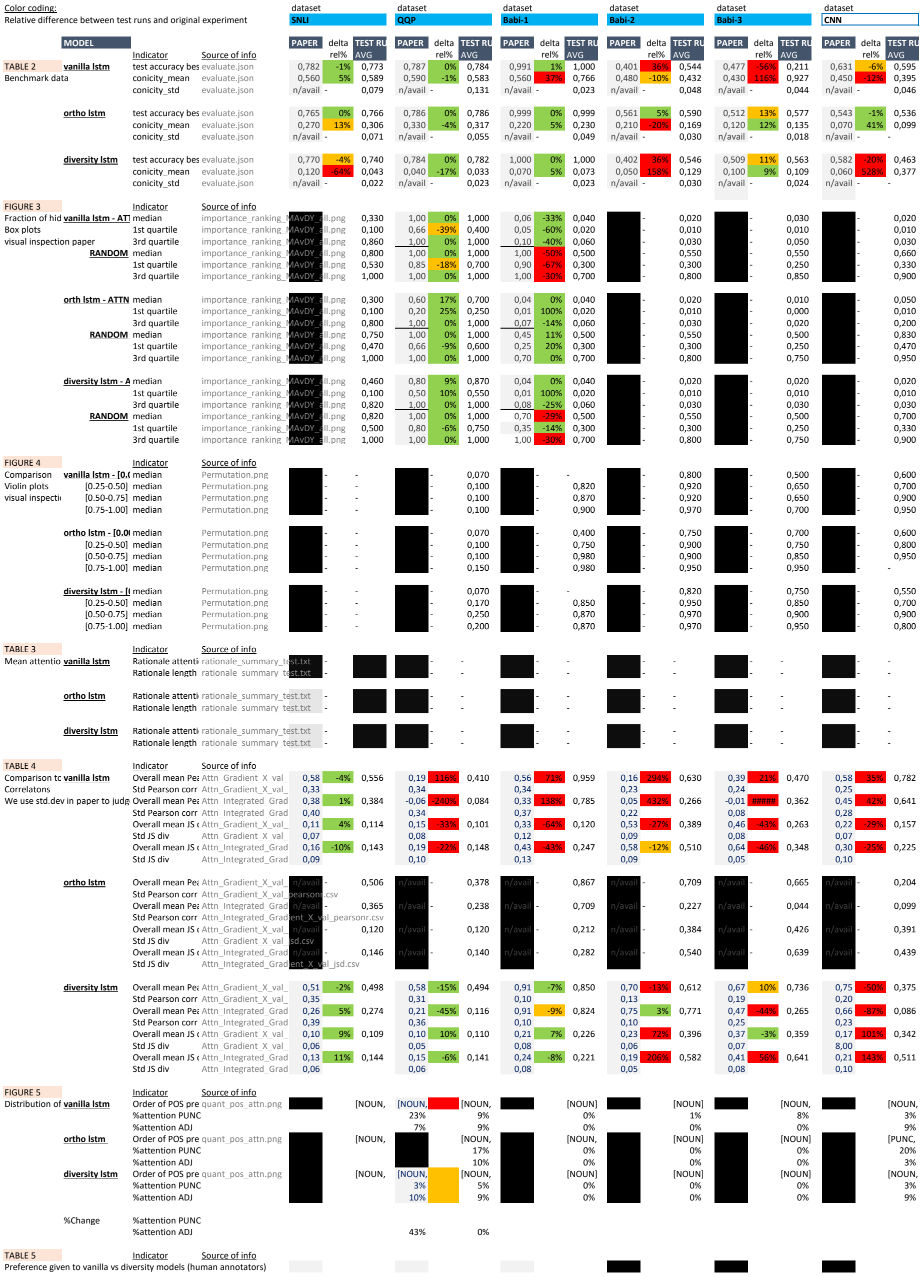
Appendix B: Full replication results



Color coding:
Relative difference between test runs and original experiment

					dataset SST			dataset imdb			dataset Yelp			dataset Amazon			dataset 20News			dataset Tweets		
	MODEL	Indicator	Source of info		PAPER	delta rel%	TEST RU AVG	PAPER	delta rel%	TEST RU AVG	PAPER	delta rel%	TEST RU AVG	PAPER	delta rel%	TEST RU AVG	PAPER	delta rel%	TEST RU AVG	PAPER	delta rel%	TEST RU AVG
TABLE 2	vanilla lstm	test accuracy bes	evaluate.json		0,818	-2%	0,803	0,895	0%	0,893	0,956	-1%	0,949	0,937	0%	0,937	0,936	-3%	0,908	0,870	-4%	0,833
Benchmark data		conicity_mean	evaluate.json		0,680	5%	0,713	0,690	-13%	0,602	0,530	1%	0,536	0,500	-9%	0,457	0,770	-1%	0,761	0,770	1%	0,776
		conicity_std	evaluate.json		n/avail	-	0,173	n/avail	-	0,135	n/avail	-	0,121	n/avail	-	0,084	n/avail	-	0,189		-	0,159
	ortho lstm	test accuracy bes	evaluate.json		0,801	-3%	0,776	0,887	0%	0,883	0,960	-2%	0,945	0,930	1%	0,936	0,922	0%	0,919	0,832	1%	0,839
		conicity_mean	evaluate.json		0,280	1%	0,283	0,180	-9%	0,163	0,180	3%	0,186	0,160	7%	0,171	0,230	2%	0,235	0,270	-4%	0,260
		conicity_std	evaluate.json		n/avail	-	0,058	n/avail	-	0,041	n/avail	-	0,044	n/avail	-	0,032	n/avail	-	0,068		-	0,062
	diversity lstm	test accuracy bes	evaluate.json		0,800	0%	0,800	0,885	-1%	0,878	0,954	-2%	0,938	0,929	0%	0,932	0,910	0%	0,908	0,870	-2%	0,854
		conicity_mean	evaluate.json		0,200	-6%	0,188	0,080	11%	0,089	0,060	478%	0,347	0,050	-2%	0,049	0,150	-10%	0,135	0,240	-4%	0,231
		conicity_std	evaluate.json		n/avail	-	0,050	n/avail	-	0,021	n/avail	-	0,014	n/avail	-	0,024	n/avail	-	0,046		-	0,067
FIGURE 3		Indicator	Source of info																			
Fraction of hid	vanilla lstm - ATT	median	importance_ranking_		■	-	0,820	0,75	-27%	0,550	0,90	-6%	0,85	0,90	-	-	0,60	-38%	0,370	■		1,000
Box plots		1st quartile	importance_ranking_		■	-	0,480	0,18	-33%	0,120	0,90	-6%	0,85	0,65	-	-	0,60	-38%	0,370	■		1,000
visual inspection paper		3rd quartile	importance_ranking_		■	-	1,000	1,00	0%	1,000	1,00	0%	1,00	1,00	-	-	1,00	0%	1,000	■		1,000
	RANDOM	median	importance_ranking_		■	-	0,850	0,98	-1%	0,970	0,93	-3%	0,90	1,00	-	-	0,93	5%	0,980	■		0,930
		1st quartile	importance_ranking_		■	-	0,850	0,88	-6%	0,830	0,93	-3%	0,90	0,93	-	-	0,93	5%	0,980	■		0,930
		3rd quartile	importance_ranking_		■	-	1,000	1,00	0%	1,000	1,00	0%	1,00	1,00		-	1,00	0%	1,000	■		1,000
	orth lstm - ATTN	median	importance_ranking_		■	-	0,220	0,05	-20%	0,040	0,25	-28%	0,180	0,30	-	-	0,15	-20%	0,120	■		0,530
		1st quartile	importance_ranking_		■	-	0,100	0,02	0%	0,020	0,13	-8%	0,120	0,20	-	-	0,07	-43%	0,040	■		0,190
		3rd quartile	importance_ranking_		■	-	0,350	0,12	-17%	0,100	0,33	-18%	0,270	0,40	-	-	0,22	45%	0,320	■		0,750
	RANDOM	median	importance_ranking_		■	-	0,780	0,92	-5%	0,870	0,93	-1%	0,920	0,90	-	-	0,93	-1%	0,920	■		0,930
		1st quartile	importance_ranking_		■	-	0,570	0,73	-15%	0,620	0,83	-10%	0,750	0,80	-	-	0,80	-8%	0,740	■		0,750
		3rd quartile	importance_ranking_		■	-	1,000	0,98	-1%	0,970	1,00	-2%	0,980	0,97		-	0,99	-1%	0,980	■		1,000
	diversity lstm - A	median	importance_ranking_		■	-	0,180	0,07	-29%	0,050	0,13	246%	0,450	0,17	-	-	0,10	-20%	0,080	■		0,370
		1st quartile	importance_ranking_		■	-	0,080	0,02	0%	0,020	0,08	125%	0,180	0,10	-	-	0,04	-25%	0,030	■		0,200
		3rd quartile	importance_ranking_		■	-	0,390	0,14	-29%	0,100	0,18	383%	0,870	0,25	-	-	0,23	13%	0,260	■		0,530
	RANDOM	median	importance_ranking_		■	-	0,770	0,88	0%	0,880	0,90	6%	0,950	0,90	-	-	0,93	-3%	0,900	■		0,880
		1st quartile	importance_ranking_		■	-	0,500	0,62	5%	0,650	0,73	12%	0,820	0,70	-	-	0,75	-1%	0,740	■		0,680
		3rd quartile	importance_ranking_		■	-	1,000	0,98	-1%	0,970	0,95	5%	1,000	1,00		-	1,00	-3%	0,970	■		1,000
FIGURE 4		Indicator	Source of info																			
Comparison	vanilla lstm - [0.0	median	Permutation.png		■	-	0,050	0,16	13%	0,180	0,03	67%	0,05	0,05	-	-	0,05	-60%	0,020	■	-	0,030
Violin plots	[0.25-0.50]	median	Permutation.png		■	-	0,080	0,17	18%	0,200	0,05	40%	0,07	0,07	-	-	0,15	-47%	0,080	■	-	0,050
visual inspecti	[0.50-0.75]	median	Permutation.png		■	-	0,230	n/appl	-	-	0,20	-25%	0,15	0,15	-	-	0,23	-	-	■	-	0,100
	[0.75-1.00]	median	Permutation.png		■	-	-	n/appl	-	-	n/appl	-	-	0,15	-	-	0,28	-	-	■	-	-
	ortho lstm - [0.0	median	Permutation.png		■	-	0,120	0,42	7%	0,450	0,37	16%	0,430	0,37	-	-	0,49	-12%	0,430	■	-	0,180
	[0.25-0.50]	median	Permutation.png		■	-	0,230	0,43	12%	0,480	0,33	21%	0,400	0,33	-	-	0,49	-6%	0,460	■	-	0,200
	[0.50-0.75]	median	Permutation.png		■	-	0,330	0,43	21%	0,520	0,35	-6%	0,330	0,40	-	-	0,48	-10%	0,430	■	-	0,250
	[0.75-1.00]	median	Permutation.png		■	-	0,360	0,37	54%	0,570	0,47	-36%	0,300	n/appl		-	0,50	-6%	0,470	■	-	0,300
	diversity lstm - [(	median	Permutation.png		■	-	0,100	0,39	10%	0,430	0,43	-30%	0,30	0,3	-	-	0,46	2%	0,470	■	-	0,200
	[0.25-0.50]	median	Permutation.png		■	-	0,230	0,44	7%	0,470	0,47	-38%	0,29	0,4	-	-	0,48	-2%	0,470	■	-	0,250
	[0.50-0.75]	median	Permutation.png		■	-	0,350	0,48	6%	0,510	0,53	-45%	0,29	0,5	-	-	0,46	4%	0,480	■	-	0,300
	[0.75-1.00]	median	Permutation.png		■	-	0,420	n/appl	-	-	0,55	-27%	0,40	n/appl		-	0,49	-4%	0,470	■	-	0,350
TABLE 3		Indicator	Source of info																			
Mean attentio	vanilla lstm	Rationale attenti	rationale_summary_te		0,348	113%	0,742	0,472	105%	0,968	0,44	-2%	0,430	0,35	-	-	0,63	-2%	0,617	0,28	189%	0,820
		Rationale length	rationale_summary_te		0,240	200%	0,719	0,217	324%	0,920	0,17	120%	0,380	0,16		-	0,22	173%	0,588	0,23	260%	0,809
	ortho lstm	Rationale attenti	rationale_summary_te		■	-	0,354	■	-	0,921	■	-	0,535	■	-	-	■		0,860	■	-	■
		Rationale length	rationale_summary_te		■	-	0,102	■	-	0,269	■	-	0,114	■		-	■		0,240	■	-	■
	diversity lstm	Rationale attenti	rationale_summary_te		0,624	-12%	0,550	0,76	20%	0,912	0,57	-54%	0,266	0,40	-	-	0,88	6%	0,940	0,76	-23%	0,591
		Rationale length	rationale_summary_te		0,175	3%	0,180	0,17	33%	0,224	0,16	19%	0,191	0,24		-	0,17	53%	0,265	0,31	4%	0,319
TABLE 4		Indicator	Source of info																			
Comparison tc	vanilla lstm	Overall mean Pea	Attn_Gradient_X_val_		0,71	-13%	0,619	0,80	8%	0,863	0,55	25%	0,690	0,43		-	0,72	-29%	0,513	0,65	-7%	0,605
Correlatons		Std Pearson corr	Attn_Gradient_X_val_		0,21			0,07			0,16			0,19	-		0,28			0,24		
We use std.dev in paper to judg		Overall mean Pea	Attn_Integrated_Grad		0,62	-19%	0,503	0,68	17%	0,793	0,40	15%	0,458	0,43		-	0,65	-40%	0,390	0,56	-14%	0,483
		Std Pearson corr	Attn_Integrated_Grad		0,24			0,09			0,19			0,19	-		0,32			0,25		
		Overall mean JS (	Attn_Gradient_X_val_		0,10	-10%	0,090	0,09	-11%	0,080	0,15	-19%	0,121	0,17		-	0,15	21%	0,182	0,08	-6%	0,075
		Std JS div	Attn_Gradient_X_val_		0,04			0,02			0,04			0,04	-		0,07			0,03		
		Overall mean JS (	Attn_Integrated_Grad		0,12	-18%	0,099	0,13	-13%	0,114	0,19	-7%	0,177	0,21		-	0,21	6%	0,222	0,08	-5%	0,076
		Std JS div	Attn_Integrated_Grad		0,05			0,02			0,05			0,06			0,06			0,04		
	ortho lstm	Overall mean Pea	Attn_Gradient_X_val_		■	-	0,726	n/avail	-	0,903	n/avail	-	0,686	n/avail		-	■	-	0,878	n/avail	-	0,747
		Std Pearson corr	Attn_Gradient_X_val_		■			■			■			■	-		■			■		
		Overall mean Pea	Attn_Integrated_Grad		■	-	0,689	n/avail	-	0,714	n/avail	-	0,668	n/avail		-	■	-	0,858	n/avail	-	0,708
		Std Pearson corr	Attn_Integrated_Grad		■			■			■			■	-		■			■		
		Overall mean JS (	Attn_Gradient_X_val_		■	-	0,143	n/avail	-	0,126	n/avail	-	0,163	n/avail		-	■	-	0,174	n/avail	-	0,184
		Std JS div	Attn_Gradient_X_val_		■			■			■			■	-		■			■		
		Overall mean JS (	Attn_Integrated_Grad		■	-	0,148	n/avail	-	0,183	n/avail	-	0,170	n/avail		-	■	-	0,152	n/avail	-	0,186
		Std JS div	Attn_Integrated_Grad		■			■			■			■	-		■			■		
	diversity lstm	Overall mean Pea	Attn_Gradient_X_val_		0,83	2%	0,849	0,89	3%	0,917	0,79	-18%	0,645	0,77		-	0,96	-5%	0,914	0,80	-3%	0,780
		Std Pearson corr	Attn_Gradient_X_val_		0,19			0,04			0,12			0,14	-		0,08			0,21		
		Overall mean Pea	Attn_Integrated_Grad		0,79	3%	0,815	0,78	1%	0,789	0,79	-29%	0,559	0,77		-	0,67	29%	0,867	0,74	-5%	0,707
		Std Pearson corr	Attn_Integrated_Grad		0,22			0,07			0,14			0,14	-		0,11			0,22		
		Overall mean JS (	Attn_Gradient_X_val_		0,08	18%	0,094	0,09	17%	0,105	0,13	33%	0,173	0,12		-	0,06	180%	0,168	0,12	-24%	0,091
		Std JS div	Attn_Gradient_X_val_		0,05			0,01			0,04			0,04	-		0,04			0,07		
		Overall mean JS (	Attn_Integrated_Grad		0,09	12%	0,101	0,13	13%	0,147	0,19	-2%	0,187	0,12		-	0,07	91%	0,134	0,15	-31%	0,103
		Std JS div	Attn_Integrated_Grad		0,05			0,02			0,05			0,04			0,05			0,06		
FIGURE 5		Indicator	Source of info																			
Distribution of	vanilla lstm	Order of POS pre	quant_pos_attn.png		[NOUN,		[NOUN,	■			[PUNC,I		[NOUN,				[NOUN,		[NOUN,	[NOUN,		[NOUN,
		%attention PUNC			0%		0%	■		0%	28%		0%				2%		3%	10%		11%
		%attention ADJ			20%		17%	■		28%	17%		17%				11%		8%	12%		10%
	ortho lstm	Order of POS pre	quant_pos_attn.png		■			■			■		[NOUN,	■			■		[NOUN,	■		[NOUN,
		%attention PUNC			■		0%	■		0%	■		-	■			■		0%	■		3%
		%attention ADJ			■		25%	■		35%	■		-	■			■		10%	■		14%
	diversity lstm	Order of POS pre	quant_pos_attn.png		[NOUN,		[NOUN,	■			[ADJ,NC		[NOUN,				[NOUN,		[NOUN,	[NOUN,		[NOUN,
		%attention PUNC			0		0%	■		0%	3%		0%				0%		0%	3%		5%
		%attention ADJ			28%		25%	■		35%	30%		17%				10%		8%	13%		12%
	%Change	%attention PUNC																				
		%attention ADJ			40%		47%			25%	76%		0%				-9%		0%	8%		20%
TABLE 5		Indicator	Source of info																			
Preference given to vanilla vs diversity models (human annotators)					■			■						■			■			■		

Figure 2. Replication results for single sequence tasks



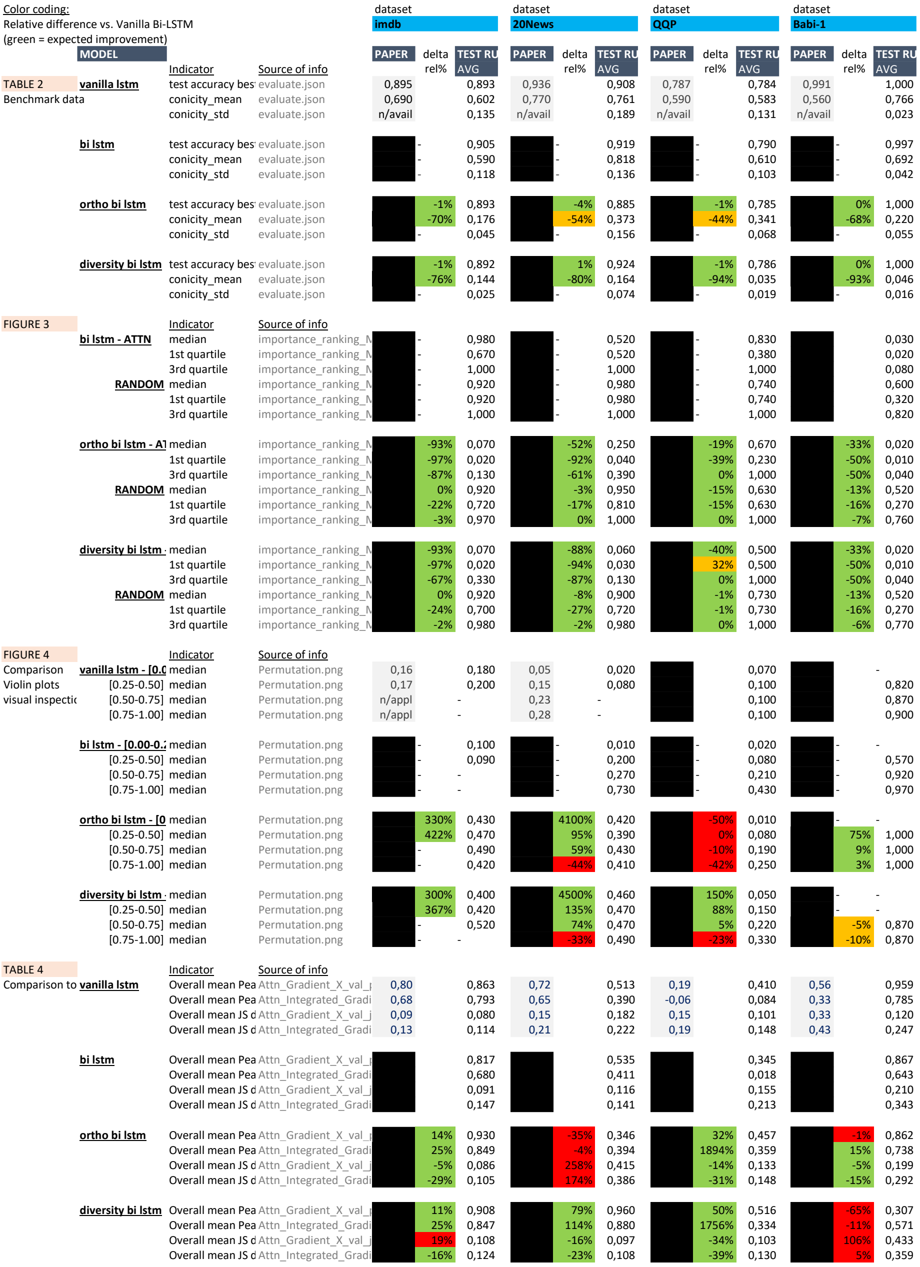
Color coding:

Relative difference between test runs and original experiment

				dataset			dataset			dataset			dataset			dataset			dataset		
				SNLI			QQP			Babi-1			Babi-2			Babi-3			CNN		
	MODEL			PAPER	delta	TEST RU	PAPER	delta	TEST RU	PAPER	delta	TEST RU	PAPER	delta	TEST RU	PAPER	delta	TEST RU	PAPER	delta	TEST RU
		Indicator	Source of info		rel%	AVG		rel%	AVG		rel%	AVG		rel%	AVG		rel%	AVG		rel%	AVG
TABLE 2	vanilla lstm	test accuracy bes	evaluate.json	0,782	-1%	0,773	0,787	0%	0,784	0,991	1%	1,000	0,401	36%	0,544	0,477	-56%	0,211	0,631	-6%	0,595
Benchmark data		conicity_mean	evaluate.json	0,560	5%	0,589	0,590	-1%	0,583	0,560	37%	0,766	0,480	-10%	0,432	0,430	116%	0,927	0,450	-12%	0,395
		conicity_std	evaluate.json	n/avail	-	0,079	n/avail	-	0,131	n/avail	-	0,023	n/avail	-	0,048	n/avail	-	0,044	n/avail	-	0,046
	ortho lstm	test accuracy bes	evaluate.json	0,765	0%	0,766	0,786	0%	0,786	0,999	0%	0,999	0,561	5%	0,590	0,512	13%	0,577	0,543	-1%	0,536
		conicity_mean	evaluate.json	0,270	13%	0,306	0,330	-4%	0,317	0,220	5%	0,230	0,210	-20%	0,169	0,120	12%	0,135	0,070	41%	0,099
		conicity_std	evaluate.json	n/avail	-	0,071	n/avail	-	0,055	n/avail	-	0,049	n/avail	-	0,030	n/avail	-	0,018	n/avail	-	-
	diversity lstm	test accuracy bes	evaluate.json	0,770	-4%	0,740	0,784	0%	0,782	1,000	0%	1,000	0,402	36%	0,546	0,509	11%	0,563	0,582	-20%	0,463
		conicity_mean	evaluate.json	0,120	-64%	0,043	0,040	-17%	0,033	0,070	5%	0,073	0,050	158%	0,129	0,100	9%	0,109	0,060	528%	0,377
		conicity_std	evaluate.json	n/avail	-	0,022	n/avail	-	0,023	n/avail	-	0,023	n/avail	-	0,030	n/avail	-	0,024	n/avail	-	-
FIGURE 3		Indicator	Source of info																		
Fraction of hid	vanilla lstm - ATT	median	importance_ranking_MAvDY_all.png			0,330	1,00	0%	1,000	0,06	-33%	0,040		-	0,020		-	0,030		-	0,020
Box plots		1st quartile	importance_ranking_MAvDY_all.png			0,100	0,66	-39%	0,400	0,05	-60%	0,020		-	0,010		-	0,010		-	0,010
visual inspection paper		3rd quartile	importance_ranking_MAvDY_all.png			0,860	1,00	0%	1,000	0,10	-40%	0,060		-	0,030		-	0,050		-	0,030
	RANDOM	median	importance_ranking_MAvDY_all.png			0,800	1,00	0%	1,000	1,00	-50%	0,500		-	0,550		-	0,550		-	0,660
		1st quartile	importance_ranking_MAvDY_all.png			0,530	0,85	-18%	0,700	0,90	-67%	0,300		-	0,300		-	0,250		-	0,330
		3rd quartile	importance_ranking_MAvDY_all.png			1,000	1,00	0%	1,000	1,00	-30%	0,700		-	0,800		-	0,850		-	0,900
	orth lstm - ATTN	median	importance_ranking_MAvDY_all.png			0,300	0,60	17%	0,700	0,04	0%	0,040		-	0,020		-	0,010		-	0,050
		1st quartile	importance_ranking_MAvDY_all.png			0,100	0,20	25%	0,250	0,01	100%	0,020		-	0,010		-	0,000		-	0,010
		3rd quartile	importance_ranking_MAvDY_all.png			0,800	1,00	0%	1,000	0,07	-14%	0,060		-	0,030		-	0,020		-	0,200
	RANDOM	median	importance_ranking_MAvDY_all.png			0,750	1,00	0%	1,000	0,45	11%	0,500		-	0,550		-	0,500		-	0,830
		1st quartile	importance_ranking_MAvDY_all.png			0,470	0,66	-9%	0,600	0,25	20%	0,300		-	0,300		-	0,250		-	0,470
		3rd quartile	importance_ranking_MAvDY_all.png			1,000	1,00	0%	1,000	0,70	0%	0,700		-	0,800		-	0,750		-	0,950
	diversity lstm - A	median	importance_ranking_MAvDY_all.png			0,460	0,80	9%	0,870	0,04	0%	0,040		-	0,020		-	0,020		-	0,020
		1st quartile	importance_ranking_MAvDY_all.png			0,100	0,50	10%	0,550	0,01	100%	0,020		-	0,010		-	0,010		-	0,010
		3rd quartile	importance_ranking_MAvDY_all.png			0,820	1,00	0%	1,000	0,08	-25%	0,060		-	0,030		-	0,030		-	0,030
	RANDOM	median	importance_ranking_MAvDY_all.png			0,820	1,00	0%	1,000	0,70	-29%	0,500		-	0,550		-	0,500		-	0,700
		1st quartile	importance_ranking_MAvDY_all.png			0,500	0,80	-6%	0,750	0,35	-14%	0,300		-	0,300		-	0,250		-	0,330
		3rd quartile	importance_ranking_MAvDY_all.png			1,000	1,00	0%	1,000	1,00	-30%	0,700		-	0,800		-	0,750		-	0,900
FIGURE 4		Indicator	Source of info																		
Comparison	vanilla lstm - [0.(	median	Permutation.png		-	-		-	0,070		-	-		-	0,800		-	0,500		-	0,600
Violin plots	[0.25-0.50]	median	Permutation.png		-	-		-	0,100		-	0,820		-	0,920		-	0,650		-	0,700
visual inspecti	[0.50-0.75]	median	Permutation.png		-	-		-	0,100		-	0,870		-	0,920		-	0,650		-	0,900
	[0.75-1.00]	median	Permutation.png		-	-		-	0,100		-	0,900		-	0,970		-	0,700		-	0,950
	ortho lstm - [0.0(	median	Permutation.png		-	-		-	0,070		-	0,400		-	0,750		-	0,700		-	0,600
	[0.25-0.50]	median	Permutation.png		-	-		-	0,100		-	0,750		-	0,900		-	0,750		-	0,800
	[0.50-0.75]	median	Permutation.png		-	-		-	0,100		-	0,980		-	0,900		-	0,850		-	0,950
	[0.75-1.00]	median	Permutation.png		-	-		-	0,150		-	0,980		-	0,950		-	0,950		-	-
	diversity lstm - [(	median	Permutation.png		-	-		-	0,070		-	-		-	0,820		-	0,750		-	0,550
	[0.25-0.50]	median	Permutation.png		-	-		-	0,170		-	0,850		-	0,950		-	0,850		-	0,700
	[0.50-0.75]	median	Permutation.png		-	-		-	0,250		-	0,870		-	0,970		-	0,900		-	0,900
	[0.75-1.00]	median	Permutation.png		-	-		-	0,200		-	0,870		-	0,970		-	0,950		-	0,800
TABLE 3		Indicator	Source of info																		
Mean attentio	vanilla lstm	Rationale attenti	rationale_summary_test.txt		-			-	-		-	-		-	-		-	-		-	-
		Rationale length	rationale_summary_test.txt		-			-	-		-	-		-	-		-	-		-	-
	ortho lstm	Rationale attenti	rationale_summary_test.txt		-			-	-		-	-		-	-		-	-		-	-
		Rationale length	rationale_summary_test.txt		-			-	-		-	-		-	-		-	-		-	-
	diversity lstm	Rationale attenti	rationale_summary_test.txt		-			-	-		-	-		-	-		-	-		-	-
		Rationale length	rationale_summary_test.txt		-			-	-		-	-		-	-		-	-		-	-
TABLE 4		Indicator	Source of info																		
Comparison tc	vanilla lstm	Overall mean Pea	Attn_Gradient_X_val_	0,58	-4%	0,556	0,19	116%	0,410	0,56	71%	0,959	0,16	294%	0,630	0,39	21%	0,470	0,58	35%	0,782
Correlatons		Std Pearson corr	Attn_Gradient_X_val_	0,33			0,34			0,34			0,23			0,24			0,25		
We use std.dev in paper to judg		Overall mean Pea	Attn_Integrated_Grad	0,38	1%	0,384	-0,06	-240%	0,084	0,33	138%	0,785	0,05	432%	0,266	-0,01	#####	0,362	0,45	42%	0,641
		Std Pearson corr	Attn_Integrated_Grad	0,40			0,34			0,37			0,22			0,08			0,28		
		Overall mean JS (	Attn_Gradient_X_val_	0,11	4%	0,114	0,15	-33%	0,101	0,33	-64%	0,120	0,53	-27%	0,389	0,46	-43%	0,263	0,22	-29%	0,157
		Std JS div	Attn_Gradient_X_val_	0,07			0,08			0,12			0,09			0,08			0,07		
		Overall mean JS (	Attn_Integrated_Grad	0,16	-10%	0,143	0,19	-22%	0,148	0,43	-43%	0,247	0,58	-12%	0,510	0,64	-46%	0,348	0,30	-25%	0,225
		Std JS div	Attn_Integrated_Grad	0,09			0,10			0,13			0,09			0,05			0,10		
	ortho lstm	Overall mean Pea	Attn_Gradient_X_val_	n/avail	-	0,506	n/avail	-	0,378	n/avail	-	0,867	n/avail	-	0,709	n/avail	-	0,665	n/avail	-	0,204
		Std Pearson corr	Attn_Gradient_X_val_pearsonr.csv																		
		Overall mean Pea	Attn_Integrated_Grad	n/avail	-	0,365	n/avail	-	0,238	n/avail	-	0,709	n/avail	-	0,227	n/avail	-	0,044	n/avail	-	0,099
		Std Pearson corr	Attn_Integrated_Gradient_X_val_pearsonr.csv																		
		Overall mean JS (	Attn_Gradient_X_val_	n/avail	-	0,120	n/avail	-	0,120	n/avail	-	0,212	n/avail	-	0,384	n/avail	-	0,426	n/avail	-	0,391
		Std JS div	Attn_Gradient_X_val_jsd.csv																		
		Overall mean JS (	Attn_Integrated_Grad	n/avail	-	0,146	n/avail	-	0,140	n/avail	-	0,282	n/avail	-	0,540	n/avail	-	0,639	n/avail	-	0,439
		Std JS div	Attn_Integrated_Gradient_X_val_jsd.csv																		
	diversity lstm	Overall mean Pea	Attn_Gradient_X_val_	0,51	-2%	0,498	0,58	-15%	0,494	0,91	-7%	0,850	0,70	-13%	0,612	0,67	10%	0,736	0,75	-50%	0,375
		Std Pearson corr	Attn_Gradient_X_val_	0,35			0,31			0,10			0,13			0,19			0,20		
		Overall mean Pea	Attn_Integrated_Grad	0,26	5%	0,274	0,21	-45%	0,116	0,91	-9%	0,824	0,75	3%	0,771	0,47	-44%	0,265	0,66	-87%	0,086
		Std Pearson corr	Attn_Integrated_Grad	0,39			0,36			0,10			0,10			0,25			0,23		
		Overall mean JS (	Attn_Gradient_X_val_	0,10	9%	0,109	0,10	10%	0,110	0,21	7%	0,226	0,23	72%	0,396	0,37	-3%	0,359	0,17	101%	0,342
		Std JS div	Attn_Gradient_X_val_	0,06			0,05			0,08			0,06			0,07			8,00		
		Overall mean JS (	Attn_Integrated_Grad	0,13	11%	0,144	0,15	-6%	0,141	0,24	-8%	0,221	0,19	206%	0,582	0,41	56%	0,641	0,21	143%	0,511
		Std JS div	Attn_Integrated_Grad	0,06			0,06			0,08			0,05			0,08			0,10		
FIGURE 5		Indicator	Source of info																		
Distribution of	vanilla lstm	Order of POS pre	quant_pos_attn.png			[NOUN,	[NOUN,		[NOUN,			[NOUN]			[NOUN]			[NOUN,			[NOUN,
		%attention PUNC					23%		9%			0%			1%			8%			3%
		%attention ADJ					7%		9%			0%			0%			0%			9%
	ortho lstm	Order of POS pre	quant_pos_attn.png			[NOUN,			[NOUN,			[NOUN,			[NOUN]			[NOUN]			[PUNC,
		%attention PUNC							17%			0%			0%			0%			20%
		%attention ADJ							10%			0%			0%			0%			3%
	diversity lstm	Order of POS pre	quant_pos_attn.png			[NOUN,	[NOUN,		[NOUN,			[NOUN]			[NOUN]			[NOUN]			[NOUN,
		%attention PUNC					3%		5%			0%			0%			0%			3%
		%attention ADJ					10%		9%			0%			0%			0%			9%
	%Change	%attention PUNC																			
		%attention ADJ					43%		0%												
TABLE 5		Indicator	Source of info																		
Preference given to vanilla vs diversity models (human annotators)																					

Figure 3. Replication results for dual sequence tasks

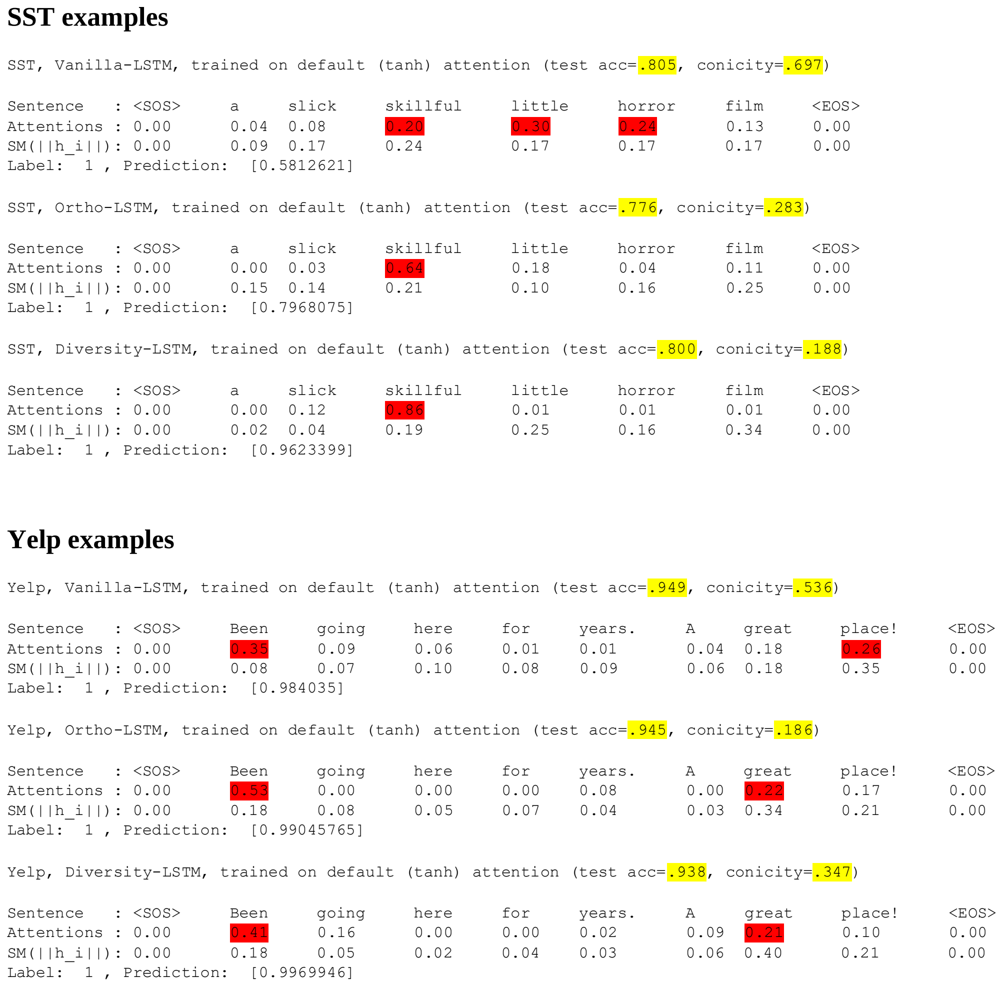
Appendix C: Results of BiLSTM extension



Color coding:				dataset			dataset			dataset			dataset		
Relative difference vs. Vanilla Bi-LSTM				imdb			20News			QQP			Babi-1		
(green = expected improvement)															
	MODEL			PAPER	delta	TEST RU	PAPER	delta	TEST RU	PAPER	delta	TEST RU	PAPER	delta	TEST RU
		Indicator	Source of info		rel%	AVG		rel%	AVG		rel%	AVG		rel%	AVG
TABLE 2	vanilla lstm	test accuracy bes	evaluate.json	0,895		0,893	0,936		0,908	0,787		0,784	0,991		1,000
Benchmark data		conicity_mean	evaluate.json	0,690		0,602	0,770		0,761	0,590		0,583	0,560		0,766
		conicity_std	evaluate.json	n/avail		0,135	n/avail		0,189	n/avail		0,131	n/avail		0,023
	bi lstm	test accuracy bes	evaluate.json		-	0,905		-	0,919		-	0,790		-	0,997
		conicity_mean	evaluate.json		-	0,590		-	0,818		-	0,610		-	0,692
		conicity_std	evaluate.json		-	0,118		-	0,136		-	0,103		-	0,042
	ortho bi lstm	test accuracy bes	evaluate.json		-1%	0,893		-4%	0,885		-1%	0,785		0%	1,000
		conicity_mean	evaluate.json		-70%	0,176		-54%	0,373		-44%	0,341		-68%	0,220
		conicity_std	evaluate.json		-	0,045		-	0,156		-	0,068		-	0,055
	diversity bi lstm	test accuracy bes	evaluate.json		-1%	0,892		1%	0,924		-1%	0,786		0%	1,000
		conicity_mean	evaluate.json		-76%	0,144		-80%	0,164		-94%	0,035		-93%	0,046
		conicity_std	evaluate.json		-	0,025		-	0,074		-	0,019		-	0,016
FIGURE 3		Indicator	Source of info												
	bi lstm - ATTN	median	importance_ranking_M		-	0,980		-	0,520		-	0,830			0,030
		1st quartile	importance_ranking_M		-	0,670		-	0,520		-	0,380			0,020
		3rd quartile	importance_ranking_M		-	1,000		-	1,000		-	1,000			0,080
	RANDOM	median	importance_ranking_M		-	0,920		-	0,980		-	0,740			0,600
		1st quartile	importance_ranking_M		-	0,920		-	0,980		-	0,740			0,320
		3rd quartile	importance_ranking_M		-	1,000		-	1,000		-	1,000			0,820
	ortho bi lstm - AT	median	importance_ranking_M		-93%	0,070		-52%	0,250		-19%	0,670		-33%	0,020
		1st quartile	importance_ranking_M		-97%	0,020		-92%	0,040		-39%	0,230		-50%	0,010
		3rd quartile	importance_ranking_M		-87%	0,130		-61%	0,390		0%	1,000		-50%	0,040
	RANDOM	median	importance_ranking_M		0%	0,920		-3%	0,950		-15%	0,630		-13%	0,520
		1st quartile	importance_ranking_M		-22%	0,720		-17%	0,810		-15%	0,630		-16%	0,270
		3rd quartile	importance_ranking_M		-3%	0,970		0%	1,000		0%	1,000		-7%	0,760
	diversity bi lstm -	median	importance_ranking_M		-93%	0,070		-88%	0,060		-40%	0,500		-33%	0,020
		1st quartile	importance_ranking_M		-97%	0,020		-94%	0,030		32%	0,500		-50%	0,010
		3rd quartile	importance_ranking_M		-67%	0,330		-87%	0,130		0%	1,000		-50%	0,040
	RANDOM	median	importance_ranking_M		0%	0,920		-8%	0,900		-1%	0,730		-13%	0,520
		1st quartile	importance_ranking_M		-24%	0,700		-27%	0,720		-1%	0,730		-16%	0,270
		3rd quartile	importance_ranking_M		-2%	0,980		-2%	0,980		0%	1,000		-6%	0,770
FIGURE 4		Indicator	Source of info												
Comparison	vanilla lstm - [0.0	median	Permutation.png	0,16		0,180	0,05		0,020			0,070			-
Violin plots	[0.25-0.50]	median	Permutation.png	0,17		0,200	0,15		0,080			0,100			0,820
visual inspecti	[0.50-0.75]	median	Permutation.png	n/appl		-	0,23		-			0,100			0,870
	[0.75-1.00]	median	Permutation.png	n/appl		-	0,28		-			0,100			0,900
	bi lstm - [0.00-0.2	median	Permutation.png		-	0,100		-	0,010		-	0,020		-	-
	[0.25-0.50]	median	Permutation.png		-	0,090		-	0,200		-	0,080		-	0,570
	[0.50-0.75]	median	Permutation.png		-	-		-	0,270		-	0,210		-	0,920
	[0.75-1.00]	median	Permutation.png		-	-		-	0,730		-	0,430		-	0,970
	ortho bi lstm - [0	median	Permutation.png		330%	0,430		4100%	0,420		-50%	0,010		-	-
	[0.25-0.50]	median	Permutation.png		422%	0,470		95%	0,390		0%	0,080		75%	1,000
	[0.50-0.75]	median	Permutation.png		-	0,490		59%	0,430		-10%	0,190		9%	1,000
	[0.75-1.00]	median	Permutation.png		-	0,420		-44%	0,410		-42%	0,250		3%	1,000
	diversity bi lstm -	median	Permutation.png		300%	0,400		4500%	0,460		150%	0,050		-	-
	[0.25-0.50]	median	Permutation.png		367%	0,420		135%	0,470		88%	0,150		-	-
	[0.50-0.75]	median	Permutation.png		-	0,520		74%	0,470		5%	0,220		-5%	0,870
	[0.75-1.00]	median	Permutation.png		-	-		-33%	0,490		-23%	0,330		-10%	0,870
TABLE 4		Indicator	Source of info												
Comparison to	vanilla lstm	Overall mean Pea	Attn_Gradient_X_val_p	0,80		0,863	0,72		0,513	0,19		0,410	0,56		0,959
		Overall mean Pea	Attn_Integrated_Gradi	0,68		0,793	0,65		0,390	-0,06		0,084	0,33		0,785
		Overall mean JS d	Attn_Gradient_X_val_j	0,09		0,080	0,15		0,182	0,15		0,101	0,33		0,120
		Overall mean JS d	Attn_Integrated_Gradi	0,13		0,114	0,21		0,222	0,19		0,148	0,43		0,247
	bi lstm	Overall mean Pea	Attn_Gradient_X_val_p			0,817			0,535			0,345			0,867
		Overall mean Pea	Attn_Integrated_Gradi			0,680			0,411			0,018			0,643
		Overall mean JS d	Attn_Gradient_X_val_j			0,091			0,116			0,155			0,210
		Overall mean JS d	Attn_Integrated_Gradi			0,147			0,141			0,213			0,343
	ortho bi lstm	Overall mean Pea	Attn_Gradient_X_val_p		14%	0,930		-35%	0,346		32%	0,457		-1%	0,862
		Overall mean Pea	Attn_Integrated_Gradi		25%	0,849		-4%	0,394		1894%	0,359		15%	0,738
		Overall mean JS d	Attn_Gradient_X_val_j		-5%	0,086		258%	0,415		-14%	0,133		-5%	0,199
		Overall mean JS d	Attn_Integrated_Gradi		-29%	0,105		174%	0,386		-31%	0,148		-15%	0,292
	diversity bi lstm	Overall mean Pea	Attn_Gradient_X_val_p		11%	0,908		79%	0,960		50%	0,516		-65%	0,307
		Overall mean Pea	Attn_Integrated_Gradi		25%	0,847		114%	0,880		1756%	0,334		-11%	0,571
		Overall mean JS d	Attn_Gradient_X_val_j		19%	0,108		-16%	0,097		-34%	0,103		106%	0,433
		Overall mean JS d	Attn_Integrated_Gradi		-16%	0,124		-23%	0,108		-39%	0,130		5%	0,359

Figure 4. Performance of BiLSTM on evaluation metrics

Appendix D: Selected data examples and illustration of model behavior



SST examples

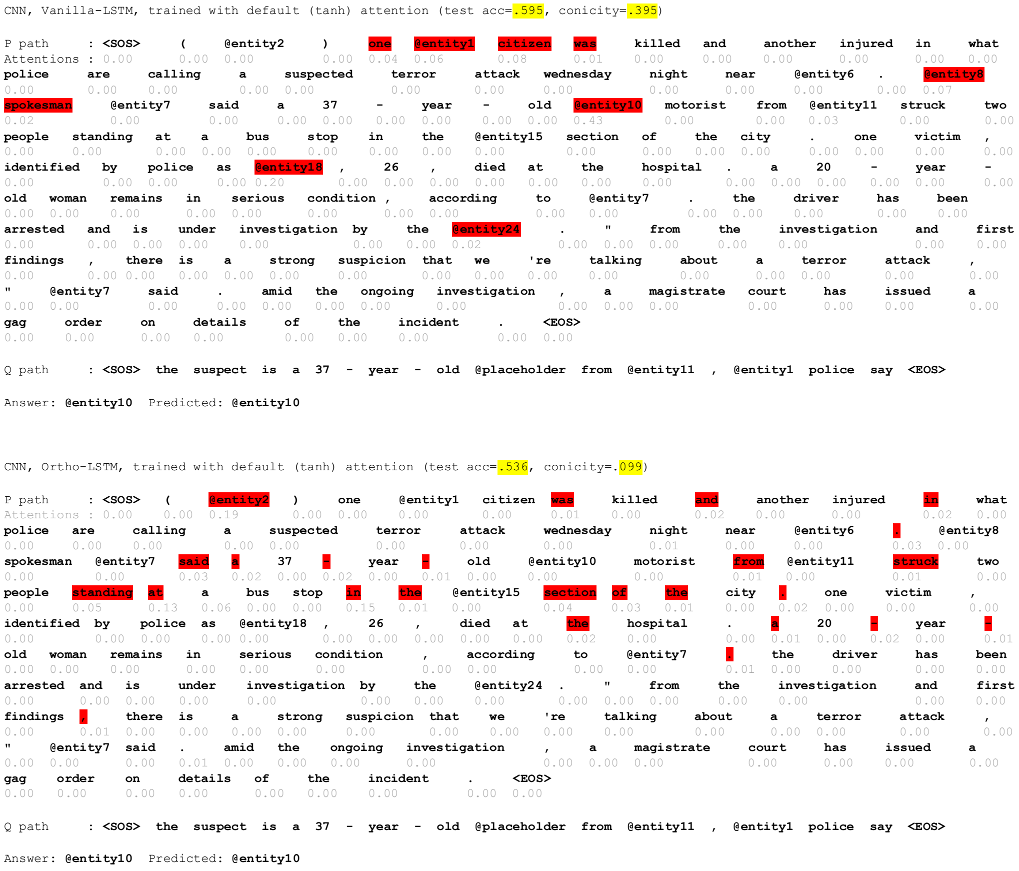
```
SST, Vanilla-LSTM, trained on default (tanh) attention (test acc=.805, conicity=.697)

Sentence   : <SOS>     a     slick     skillful     little     horror     film     <EOS>
Attentions : 0.00      0.04  0.08      0.20         0.30       0.24       0.13     0.00
SM(||h_i||): 0.00      0.09  0.17      0.24         0.17       0.17       0.17     0.00
Label:  1 , Prediction:  [0.5812621]

SST, Ortho-LSTM, trained on default (tanh) attention (test acc=.776, conicity=.283)

Sentence   : <SOS>     a     slick     skillful     little     horror     film     <EOS>
Attentions : 0.00      0.00  0.03      0.64         0.18       0.04       0.11     0.00
SM(||h_i||): 0.00      0.15  0.14      0.21         0.10       0.16       0.25     0.00
Label:  1 , Prediction:  [0.7968075]

SST, Diversity-LSTM, trained on default (tanh) attention (test acc=.800, conicity=.188)

Sentence   : <SOS>     a     slick     skillful     little     horror     film     <EOS>
Attentions : 0.00      0.00  0.12      0.86         0.01       0.01       0.01     0.00
SM(||h_i||): 0.00      0.02  0.04      0.19         0.25       0.16       0.34     0.00
Label:  1 , Prediction:  [0.9623399]
```

Yelp examples

```
Yelp, Vanilla-LSTM, trained on default (tanh) attention (test acc=.949, conicity=.536)

Sentence   : <SOS>     Been     going     here     for     years.     A     great     place!     <EOS>
Attentions : 0.00      0.35     0.09      0.06     0.01    0.01       0.04  0.18      0.26       0.00
SM(||h_i||): 0.00      0.08     0.07      0.10     0.08    0.09       0.06  0.18      0.35       0.00
Label:  1 , Prediction:  [0.984035]

Yelp, Ortho-LSTM, trained on default (tanh) attention (test acc=.945, conicity=.186)

Sentence   : <SOS>     Been     going     here     for     years.     A     great     place!     <EOS>
Attentions : 0.00      0.53     0.00      0.00     0.00    0.08       0.00  0.22      0.17       0.00
SM(||h_i||): 0.00      0.18     0.08      0.05     0.07    0.04       0.03  0.34      0.21       0.00
Label:  1 , Prediction:  [0.99045765]

Yelp, Diversity-LSTM, trained on default (tanh) attention (test acc=.938, conicity=.347)

Sentence   : <SOS>     Been     going     here     for     years.     A     great     place!     <EOS>
Attentions : 0.00      0.41     0.16      0.00     0.00    0.02       0.09  0.21      0.10       0.00
SM(||h_i||): 0.00      0.18     0.05      0.02     0.04    0.03       0.06  0.40      0.21       0.00
Label:  1 , Prediction:  [0.9969946]
```

Figure 5. Examples of single input sequence tasks

CNN examples



```
CNN, Vanilla-LSTM, trained with default (tanh) attention (test acc=.595, conicity=.395)

P path     : <SOS>     (     @entity2     )     one   @entity1   citizen   was     killed   and     another   injured   in     what
Attentions : 0.00      0.00  0.00         0.00  0.04  0.06       0.08      0.01    0.00     0.00    0.00      0.00      0.00   0.00
police     are     calling     a     suspected     terror     attack   wednesday     night     near     @entity6   .     @entity8
0.00       0.00    0.00        0.00  0.00          0.00       0.00     0.00          0.00      0.00     0.00       0.00  0.07
spokesman     @entity7     said     a     37     -     year    -     old   @entity10   motorist     from  @entity11   struck      two
0.02          0.00         0.00     0.00  0.00   0.00  0.00    0.00  0.00  0.43        0.00         0.00  0.03        0.00        0.00
people   standing   at    a     bus     stop     in    the     @entity15   section   of     the   city     .     one     victim    ,
0.00     0.00       0.00  0.00  0.00    0.00    0.00   0.00   0.00        0.00      0.00   0.00  0.00     0.00  0.00    0.00      0.00
identified   by    police   as   @entity18  ,     26    ,     died   at     the     hospital   .     a     20     -     year      -
0.00         0.00  0.00     0.00 0.20       0.00  0.00  0.00  0.00   0.00   0.00    0.00       0.00  0.00  0.00   0.00  0.00      0.00
old   woman   remains   in    serious   condition ,     according     to     @entity7     .     the     driver     has     been
0.00  0.00    0.00      0.00  0.00      0.00      0.00  0.00          0.00   0.00         0.00  0.00    0.00       0.00    0.00
arrested   and   is    under   investigation  by     the   @entity24     .     "     from     the     investigation     and      first
0.00       0.00  0.00  0.00    0.00           0.00   0.00  0.02          0.00  0.00  0.00     0.00    0.00              0.00     0.00
findings   ,    there  is    a     strong   suspicion  that   we     're     talking     about     a     terror     attack     ,
0.00       0.00 0.00   0.00  0.00  0.00     0.00       0.00   0.00   0.00    0.00        0.00      0.00  0.00       0.00       0.00
"     @entity7     said     .     amid   the   ongoing   investigation   ,     a     magistrate   court     has     issued      a
0.00  0.00         0.00     0.00  0.00   0.00  0.00      0.00            0.00  0.00  0.00         0.00      0.00    0.00        0.00
gag     order     on     details     of     the     incident     .     <EOS>
0.00    0.00      0.00   0.00        0.00   0.00    0.00         0.00  0.00

Q path     : <SOS>  the  suspect  is  a  37  -  year  -  old  @placeholder  from  @entity11  ,  @entity1  police   say  <EOS>

Answer: @entity10  Predicted: @entity10


CNN, Ortho-LSTM, trained with default (tanh) attention (test acc=.536, conicity=.099)

P path     : <SOS>   (     @entity2   )     one     @entity1   citizen  was     killed     and     another   injured     in     what
Attentions : 0.00    0.00  0.19       0.00  0.00    0.00       0.00     0.01    0.00       0.02    0.00      0.00        0.02   0.00
police   are     calling     a     suspected     terror     attack     wednesday     night     near     @entity6     .      @entity8
0.00     0.00    0.00        0.00  0.00          0.00       0.00       0.00          0.01      0.00     0.00         0.03   0.00
spokesman   @entity7   said   a     37    -     year   -     old     @entity10     motorist     from   @entity11     struck      two
0.00        0.00       0.03   0.02  0.00  0.02  0.00   0.01  0.00    0.00          0.00         0.01   0.00          0.01        0.00
people   standing  at     a     bus   stop   in     the    @entity15   section  of     the     city   .     one     victim      ,
0.00     0.05      0.13   0.06  0.00  0.00  0.15   0.01   0.00        0.04     0.03   0.01    0.00   0.02  0.00    0.00        0.00
identified  by    police  as   @entity18  ,     26    ,     died   at     the     hospital     .     a     20     -     year      -
0.00        0.00  0.00    0.00 0.00       0.00  0.00  0.00  0.00   0.00   0.02    0.00         0.00  0.01  0.00   0.02  0.00      0.01
old   woman   remains   in     serious   condition     ,     according     to     @entity7     .     the     driver     has     been
0.00  0.00    0.00      0.00   0.00      0.00          0.00  0.00          0.00   0.00         0.01  0.00    0.00       0.00    0.00
arrested  and   is     under    investigation  by     the     @entity24  .     "     from     the     investigation     and      first
0.00      0.00  0.00   0.00     0.00           0.00   0.00    0.00       0.00  0.00  0.00     0.00    0.00              0.00     0.00
findings  ,     there  is     a     strong   suspicion  that    we     're     talking     about     a     terror     attack      ,
0.00      0.01  0.00   0.00   0.00  0.00     0.00       0.00    0.00   0.00    0.00        0.00      0.00  0.00       0.00        0.00
"     @entity7  said   .     amid   the    ongoing   investigation     ,     a     magistrate     court     has     issued      a
0.00  0.00      0.00   0.01  0.00   0.00   0.00      0.00              0.00  0.00  0.00           0.00      0.00    0.00        0.00
gag    order    on     details   of     the     incident     .     <EOS>
0.00   0.00     0.00   0.00      0.00   0.00    0.00         0.00  0.00

Q path     : <SOS>  the  suspect  is  a  37  -  year  -  old  @placeholder  from  @entity11  ,  @entity1  police   say  <EOS>

Answer: @entity10  Predicted: @entity10
```

Figure 6. Examples of dual input sequence tasks