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

Anonymous Author(s) Affiliation Address email

Reproducibility Summary

2 It has been shown (Jain and Wallace, 2019) that the weights in attention mechanisms do not necessarily offer a

³ faithful explanation of the model's predictions. In the paper 'Towards Transparent and Explainable Attention Models'

4 (Mohankumar et al., 2020) the authors propose two methods to enhance faithfulness and plausibility of the explanations

5 provided by an LSTM model combined with a basic attention mechanism.

1

6 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.

12 **Methodology** The paper includes a link to a repository with the code used to generate its results. All our exper-13 iments with this code are conducted on GPU nodes of the Lisa Cluster at SURFsara¹. We follow four investiga-14 tive routes: (i) Replication: we rerun experiments on datasets from the paper in order to replicate the results, and 15 add the results that are missing in the paper; (ii) Code review: we scrutinize the code to validate its correctness; 16 (iii) Evaluation methodology: we extend the set of evaluation metrics used in the paper with the LIME method, in an 17 attempt to resolve inconclusive results; (iv) Generalization to other architectures: we test whether the authors' claims 18 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 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.

¹We had access to two Nvdia GTX1080Ti/11Gb VRAM GPUs. https://userinfo.surfsara.nl/systems/lisa/description.

33 1 Introduction

The popularity of attention models has sparked many studies on the interpretability of the attention distributions, with 34 often conflicting claims (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Serrano and Smith, 2019). Mohankumar 35 et al. (2020) argue that the reason why attention weights do not always provide a faithful explanation of the model's 36 predictions is that the learned hidden states of the LSTM based encoder are very similar across time steps, which is 37 expressed by high conicity of these vectors. As a result, random permutation of the attention weights leads to a similar 38 final context vector, which implies the weights do not provide a faithful explanation. The authors propose two methods 39 that force the hidden states of the LSTM to be more diverse. Orthogonal LSTM ensures low conicity by orthogonalizing 40 the hidden state at time t with respect to the mean of the previous hidden states. In Diversity LSTM the model is trained 41 to jointly maximize the log-likelihood of the training data and to minimize the conicity of the hidden states. 42

43 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 (Mohankumar et al., 2020) 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. (2020), 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' code² 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 method³;
- 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.

61 3 Methodology

Code The code accompanying the paper is an extension based on the code first developed by Jain and Wallace (2019)⁴.
 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 included⁵.

Datasets We reran the experiments on 11 of the 14 datasets used in the paper. The nature and size of the datasets 65 covers a wide range, from relatively simple binary sentiment classification tasks with single input sequence (abbreviated: 66 SS) (e.g. SST with average input sentence length of 20 words), to complex question answering tasks with dual input 67 sequences (abbreviated: DS)⁶ (e.g. CNN with average document size of 760 words and an average of 26 answer 68 categories). Some illustrations of data points can be found in Appendix D. The code repository includes links to the 69 datasets, as well as the pre-processing routines used by the authors. We excluded the Amenia and Diabetes datasets 70 because they were not accessible. The Amazon dataset caused memory issues when running the experiments. Despite 71 these issues we were able to get the accuracies and conicity values for this dataset. 72

73 Model descriptions The baseline model (Vanilla LSTM) used in the paper is shown in Figure 1. For DS tasks, it 74 consists of two uni-directional LSTM encoders that act on a P-path (for document input phrases) and a Q-path (for

²https://github.com/akashkm99/Interpretable-Attention

³The code with all extensions we made for this review can be accessed at https://anonymous.4open.science/r/FACT_AI_project/

⁴https://github.com/successar/AttentionExplanation

⁵Diabetes and Anemia are not included. These datasets are not publicly available (ethics screening required before use)

⁶These distinctions are differently named in the code: SS is referred to as BC and DS as QA

rs question input phrases). When applied on SS tasks in the paper, only the P-path is used. An attention decoder is applied

- ⁷⁶ to the hidden states of the P-path LSTM to form the context vector \mathbf{c}_{α} on which the model calculates its output. The
- ⁷⁷ 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. (2020) propose are variants of the baseline model. The

- 79 Orthogonal LSTM applies an orthogonalization procedure to the LSTM hidden state vectors during training: the
- 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
- ⁸² orthogonalization, but minimizes conicity jointly with the standard loss.



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

83 Hyperparameters Given the wide variety of tasks and datasets, there is an elaborate set of model- and optimization

⁸⁴ hyperparameters. Not all parameter values are indicated in the original paper, some were retrieved by inspecting the

⁸⁵ code (an overview is presented in Appendix A). For all parameters, we used the defaults provided in the original code.

⁸⁶ 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.

Experimental setup and computational requirements We strictly follow the code environment as dictated by the requirements file that accompanies the code. All models are run on 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

94 conclusive.

95 **4** Replication of the paper's results

96 4.1 Core replication results

Our reproduction study reveals numerous differences in results reported by Mohankumar et al. (2020), 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. (2020), 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.

Dataset			Accura	acy%					Coni	city		
	Vanilla l	LSTM	Diversity	LSTM	Orthogona	al LSTM	Vanilla I	LSTM	Diversity	LSTM	Orthogona	al LSTM
	original	rerun	original	rerun	original	rerun	original	rerun	original	rerun	original	rerun
SST	81.79	80.3	79.95	80.0	80.05	77.6	0.68	0.71	0.20	0.19	0.28	0.28
IMDB	89.49	89.3	88.54	87.8	88.71	88.3	0.69	0.60	0.08	0.09	0.18	0.16
Yelp	95.60	94.5	95.40	93.8	96.00	94.5	0.53	0.54	0.06	0.35	0.18	0.19
20News	93.55	90.8	91.03	90.8	92.15	91.9	0.77	0.76	0.15	0.14	0.23	0.24
Tweets	87.02	83.3	87.04	85.4	83.20	83.9	0.77	0.78	0.24	0.23	0.27	0.26
SNLI	78.23	77.3	76.96	74.0	76.46	76.6	0.56	0.59	0.12	0.04	0.27	0.31
QQP	78.74	78.4	78.40	78.2	78.61	78.6	0.59	0.58	0.04	0.03	0.33	0.32
bAbI1	99.10	100.0	100.00	100.0	99.90	99.9	0.56	0.77	0.07	0.07	0.22	0.23
bAbI2	40.10	54.4	40.20	54.6	56.10	59.0	0.48	0.43	0.05	0.13	0.21	0.17
bAbI3	47.70	21.1	50.90	56.3	51.20	57.7	0.43	0.93	0.10	0.11	0.12	0.13
CNN	63.07	59.5	58.19	46.3	54.30	53.6	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).

Importance of hidden representation Mohankumar et al. (2020) analyse the importance of hidden representations using intermediate representation erasure (Serrano and Smith, 2019) and also by examining the effect of permuting the attentions using the using the using the use (Lein and Wollage, 2010).

112 attentions weights (Jain and Wallace, 2019).

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

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. (2020) do not report on SNLI and CNN, where our rerun shows no improvement of the

Diversity LSTM and Orthogonal LSTM models over Vanilla LSTM.

¹²⁰ The impact of permuting attention weights is difficult to compare with our results as Mohankumar et al. (2020) 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. (2020). 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 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. (2020), 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. (2020), 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.

The second secon

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% instead of the 65%

reported by Mohankumar et al. $(2020)^7$. However, we do not see the claimed reduction in JS Divergence. In fact, for

⁷This percentage represents the average of the increases over all datasets.

Dataset			Rationale attention						Rationale	e length		
	Vanilla I	LSTM	Diversity	LSTM	Orthogona	ıl LSTM	Vanilla	LSTM	Diversity	LSTM	Orthogona	al LSTM
	original	rerun	original	rerun	original	rerun	original	rerun	original	rerun	original	rerun
SST	0.348	0.74	0.624	0.55	-	0.35	0.240	0.72	0.175	0.18	-	0.10
IMBD	0.472	0.97	0.761	0.91	-	0.92	0.217	0.92	0.169	0.22	-	0.27
Yelp	0.438	0.43	0.574	0.27	-	0.55	0.173	0.38	0.160	0.19	-	0.11
20News	0.627	0.62	0.884	0.94	-	0.86	0.215	0.59	0.173	0.27	-	0.24
Tweets	0.284	0.82	0.764	0.59	-	0.79	0.225	0.81	0.306	0.32	-	0.39

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

- all datasets the Diversity LSTM produces similar or even higher JS Divergence values than Vanilla LSTM, except JS
- 143 Divergence with Integrated Gradients for 20News, see Table 3. The Orthogonal LSTM, for which no correlation data is
- reported in the paper, is in line with the Diversity LSTM in this respect.

Dataset			JS Divergenc	e Gradients				JS Di	vergence Inte	grated Grad	lients	
	Vanilla I	LSTM	Diversity	LSTM	Orthogona	al LSTM	Vanilla I	LSTM	Diversity	LSTM	Orthogona	l LSTM
	original	rerun	original	rerun	original	rerun	original	rerun	original	rerun	original	rerun
SST	0.10	0.09	0.08	0.09	-	0.14	0.12	0.10	0.09	0.10	-	0.15
IMDB	0.09	0.08	0.09	0.11	-	0.13	0.13	0.11	0.13	0.15	-	0.18
Yelp	0.15	0.12	0.13	0.17	-	0.16	0.19	0.18	0.19	0.19	-	0.17
20News	0.15	0.18	0.06	0.17	-	0.17	0.21	0.22	0.07	0.13	-	0.15
Tweets	0.08	0.07	0.12	0.09	-	0.18	0.08	0.08	0.15	0.10	-	0.19
SNLI	0.11	0.11	0.10	0.11	-	0.12	0.16	0.14	0.13	0.14	-	0.15
QQP	0.15	0.10	0.10	0.11	-	0.12	0.19	0.15	0.15	0.14	-	0.14
bAbI1	0.33	0.12	0.21	0.23	-	0.21	0.43	0.25	0.24	0.22	-	0.28
bAbI2	0.53	0.39	0.23	0.40	-	0.38	0.58	0.51	0.19	0.58	-	0.54
bAbI3	0.46	0.26	0.37	0.36	-	0.43	0.64	0.35	0.41	0.64	-	0.64
CNN	0.22	0.16	0.17	0.34	-	0.39	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,

147 with vanilla LSTM model for Yelp we see no attention given to punctuations (PUNC), for which Mohankumar et al.

(2020) reports highest attention. For QQP, Mohankumar et al. (2020) 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. (2020).

150 **Human evaluation** We could not reproduce the human evaluation within the four-week time frame of our research.

Mohankumar et al. (2020) 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.

153 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.

The Orthogonal LSTM clearly leads to lower conicity than Vanilla LSTM, but Mohankumar et al. (2020) 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. (2020) 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.

166 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,

Metric	Claim (with reference to paragraph number in Mohankumar et al. (2020))	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 representa-	Diversity LSTM and Orthogonal LSTM reach quicker decision flip	Yes (for mod-	Especially for BC tasks, some-
tion required for decision flip	(§5.3)	els in paper)	what for QQP; not for SNLI
(box plots)			and QA tasks
Median Output difference on	Diversity LSTM and Orthogonal LSTM are more sensitive to ran-	Yes (for mod-	Clear difference for BC tasks,
randomly permuting attention	dom permutation of weights than Vanilla LSTM (§5.3)	els in paper)	mixed picture for the dual-
weights (violin charts)			sequence tasks.
Rationale attention	Diversity LSTM provides much higher attention to rationales than	No	Only true for 20News; No re-
	Vanilla LSTM across the 8 Text classification datasets (§5.4)		sults reported on QA tasks
Rationale length	Diversity LSTM often provides shorter rationales than Vanilla	Yes	No results reported on QA
	LSTM (§5.4)		tasks
Pearson correlation and JS di-	Attention weights in Diversity LSTM better agree with gradients	Mixed	Diversity LSTM has higher
vergence between distribution	and integrated gradients than Vanilla LSTM (§5.5)		Pearson correlation, but simi-
of attention and (integrated)			lar or higher JS Divergence
gradients			
Attention given to POS tags	Attention given to punctuation marks is significantly reduced on	No	Not for Yelp, less clear for
	the Yelp, Amazon and QQP datasets (§5.6)		QQP
	Diversity LSTM gives much more attention to adjectives than	Yes	True for SST and IMDB, but
	Vanilla LSTM in the four sentiment analysis tasks (SST, IMDB,		not for Yelp
	Yelp, Amazon) (§5.6)		-
Human evaluation of plausibil-	Human evaluators prefer attention distribution of Diversity LSTM	Not repro-	Evaluation by only 15 people
ity	over Vanilla LSTM for Yelp, SNLI, QQP and bAbI1 (§5.7)	duced	

Table 4: Evidence for authors' claims after rerun

datasets and model configurations. While convenient, this also makes the codebase complex and susceptible to errors.

171 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 <u>last</u> 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

182 mechanism from finding the right prompt words.

In order to test this sensitivity, we conduct an experiment 183 for the simpler SS tasks. We apply orthogonalization during 184 training and compare model performance when i) attention 185 weights are left unconstrained vs. ii) all attention weights are 186 set to zero, except for the last hidden state. The result is shown 187 in Table 5. What is striking is the performance remains on 188 par (marked in green) for Vanilla LSTM when only attending 189 to the last hidden state, indicating the model performs well 190 without the attention mechanism. However, we observe a 191 performance drop of 10%-34% (absolute) when attention is 192 constrained for the Orthogonal LSTM (marked in red). Indeed, 193 it appears part of the information required for inference is lost. 194

How this effect impacts the results requires further study. It

may explain the accuracy drop from 63% (Vanilla LSTM) to

197 58%/54% (Diversity/Orthogonal LSTM) for CNN as reported

¹⁹⁸ in Table 2 by Mohankumar et al. (2020). We have contacted

Vanilla LSTM Orthogonal LSTM Base last only Base last only Dataset attention attention attention 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 (0.602)(0.885)(0.163) (.141) (conicity) 20News: accuracy 0.908 0.919 0.583 (0.761)(0.831) (0.235)(0.395) (conicity) Tweets: accuracy 0.833 0.839 0.712 0.78 (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)

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 $\hat{y} =$ softmax $(\mathbf{W}_r(\tanh(\mathbf{W}_p\mathbf{c}_{\alpha}+\mathbf{b}_p+\mathbf{W}_q\mathbf{h}_n^q+\mathbf{b}_q))+\mathbf{b}_r)$. This deviates from the prediction function $\hat{y} = \operatorname{softmax}(\mathbf{W}_0\mathbf{c}_{\alpha})$ described in Section 2.1 by Mohankumar et al. $(2020)^8$. 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 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.

212 6 Extension of the evaluation methods

As discussed in Section 4.1, 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 (Ribeiro et al., 2016) 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.

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

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

224 constraints.

225 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

²³¹ Divergence at the same time when going from Vanilla LSTM to Orthogonal/Diversity LSTM models.

232 7 Generalization to other model architectures

²³³ Despite the differences we found between our observations and the observations reported by (Mohankumar et al., 2020),

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. (2020).

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 uni-directional LSTM models with attention. However, in many recent studies, BiLSTM-based attention models or Transformer models are used (Zhou and Wu, 2018; Lee et al., 2019; Aziz Sharfuddin et al., 2018). Similarly, more complex attention mechanisms such as self-attention and multi-head attention (Vaswani et al., 2017) 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.

²⁴² proposed methods can be extended to more complex models and whether the authors two main claims stin appr

Extending to other attention mechanisms The application of more advanced attention mechanisms (such as multihead 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 (Abnar and Zuidema, 2020) can provide a resolution.

	Pearso	n's Correla	tion	JS	Divergence	e
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

⁸https://github.com/akashkm99/Interpretable-Attention/blob/master/model/modules/Decoder.py#L101-L107

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. (2020). 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.

259

Results show that the application of the two methods proposed by Mohankumar et al. (2020) 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.

265 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. (2020). 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. (2020) 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 attraction weights correct on the sector.

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 (2019).

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

²⁸⁸ best results, deserves additional investigation.

	Base atte	ention	Constraine	ed 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, firstand last only attention

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.

298 Acknowledgement

²⁹⁹ We'd like to thank [name to be added] for his guidance and insightful discussions.

300 **References**

- Samira Abnar and Willem Zuidema. 2020. Quantifying Attention Flow in Transformers. In *Proceedings of the* 58th Annual Meeting of the Association for Computational Linguistics, pages 4190–4197, Online. Association for
- 303 Computational Linguistics.
- A. Aziz Sharfuddin, M. Nafis Tihami, and M. Saiful Islam. 2018. A deep recurrent neural network with bilstm model
 for sentiment classification. In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP),
 pages 1–4.
- ³⁰⁷ Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. *CoRR*, abs/1902.10186.
- Lung-Hao Lee, Yi Lu, Po-Han Chen, Po-Lei Lee, and Kuo-Kai Shyu. 2019. Ncuee at mediqa 2019: medical text
 inference using ensemble bert-bilstm-attention model. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 528–532.
- Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasan Srinivasan, and
 Balaraman Ravindran. 2020. Towards Transparent and Explainable Attention Models. In *Proceedings of the* 58th Annual Meeting of the Association for Computational Linguistics, pages 4206–4216, Online. Association for
 Computational Linguistics.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "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*, pages 1135–1144.
- 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, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and
 Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

- Sarah Wiegreffe and Yuval Pinter. 2019. 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)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.
- Qimin Zhou and Hao Wu. 2018. 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*
- 329 Social Media Analysis, pages 189–194.

Appendix A: Details of models and datasets 330

		9	Single input se	equence tasks				D	ual input seq	uence 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 hyperparamete	rs											
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, includi	ng fine-tunin	g of embedd	ings									
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, withou	t fine-tuning	of embeddi	ngs									
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

		Nu	mber of datapoint	s	Avg seq. le	ngth (train)	Avg.no.answer	Vocab. size
Dataset	Description	train (%pos)	dev (%pos)	test (%pos)	Document	Question	categories	(train, docs)
Single inpu	t 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

331 Appendix B: Full replication results

	t runs and origina	l experiment	SST		IMdb		Yelp		Amazon		20News		Tweets	_
MODEL	Indicator	Source of info		TEST RU AVG	PAPER delta rel%			elta TEST RU el% AVG	PAPER delta rel%		PAPER delt rel9		PAPER del rel	lta TE
BLE 2 vanilla lstm hchmark data	test accuracy bes		0,818 -2% 0,680 5% n/avail -	0,803 0,713 0,173	0,895 09 0,690 -139 n/avail -		0,956 0,530 n/avail -	-1% 0,949 1% 0,536 0,121	0,937 09 0,500 -99 n/avail -		0,936 -3 0,770 -1 n/avail -			4% (1% (
ortho lstm		evaluate.json	0,801 -3% 0,280 1%	0,776 0,283	0,887 09 0,180 -99	6 0,163	0,180	-2% 0,945 3% 0,186	0,930 19 0,160 79	6 0,171	0,922 0 0,230 2	% 0,235		1% (4% (
diversity lstm	conicity_std test accuracy bes conicity mean	evaluate.json evaluate.json evaluate.json	n/avail - 0,800 0% 0,200 -6%	0,058 0,800 0,188	n/avail - 0,885 -19 0,080 119		n/avail - 0,954 0,060 4	0,044 -2% 0,938 78% 0,347	n/avail - 0,929 09 0,050 -29		n/avail - 0,910 0 0,150 -10			2% (4% (
	conicity_std	evaluate.json	n/avail -	0,050	n/avail -	0,021	n/avail -	0,014	n/avail -	0,024	n/avail -	0,046	-	(
URE 3 tion of hid vanilla lstm - A1		Source of info importance_ranking_	-	0,820	0,75 -279	0,550		-6% 0,85	0,90 -	-	0,60 -38	% 0,370		:
plots al inspection paper	1st quartile 3rd quartile	importance_ranking_ importance_ranking_	-	0,480 1,000	0,18 -339 1,00 09	0,120 1,000	0,90 1,00	-6% 0,85 0% 1,00	0,65 - 1,00 -		0,60 -38	% 0,370 % 1,000		
RANDON	1 median 1st quartile 3rd quartile	importance_ranking_ importance_ranking_ importance_ranking	 	0,850 0,850 1,000	0,98 -19 0,88 -69 1,00 09	6 0,830	0,93	-3% 0,90 -3% 0,90 0% 1,00	1,00 - 0,93 - 1,00	-	0,93 5 0,93 5 1,00 0	% 0,980		
orth lstm - ATT	median 1st quartile	importance_ranking_ importance_ranking		0,220 0,100	0,05 -209 0,02 09	6 0,040	0,25	28% 0,180 -8% 0,120	- 0,30 - 0,20 -	-	0,15 -20 0,07 -43	% 0,120		
	3rd quartile	importance_ranking_	-	0,350	0,12 -179	6 0,100	0,33 -:	18% 0,270	0,40 -		0,22 45	% 0,320		
RANDON	1st quartile 3rd quartile	<pre>importance_ranking_ importance_ranking_ importance_ranking_</pre>	-	0,780 0,570 1,000	0,92 -59 0,73 -159 0,98 -19	6 0,620	0,83 -	-1% 0,920 10% 0,750 -2% 0,980	0,90 - 0,80 - 0,97	-	0,93 -1 0,80 -8 0,99 -1	% 0,740		
diversity lstm -	nedian 1st quartile	importance_ranking_ importance_ranking_	-	0,180 0,080	0,07 -299 0,02 09		0,13 2	46% 0,450 25% 0,180	0,17 - 0,10 -	-	0,10 -20 0,04 -25			
RANDON	3rd quartile median	importance_ranking_ importance ranking	-	0,390 0,770	0,14 -299 0,88 09		0,18 3	83% 0,870 6% 0,950	0,25 - 0,90 -		0,23 13 0,93 -3			
	1st quartile 3rd quartile	importance_ranking_ importance_ranking_	-	0,500 1,000	0,62 59 0,98 -19	6 0,650		12% 0,820 5% 1,000	0,70 - 1,00	-	0,75 -1 1,00 -3	% 0,740		
JRE 4 Iparison vanilla Istm - [0		Source of info Permutation.png	-	0,050	0,16 139			67% 0,05	0,05 -		0,05 -60		-	
n plots [0.25-0.50 al inspecti [0.50-0.75 [0.75-1.00	median	Permutation.png Permutation.png Permutation.png	-	0,080 0,230 -	0,17 189 n/appl - n/appl -	0,200 - -		40% 0,07 25% 0,15 -	0,07 - 0,15 - 0,15	-	0,15 -47 0,23 - 0,28 -	• 0,080 - -	-	
ortho lstm - [0.0		Permutation.png	·	0,120	0,42 79			16% 0,430	0,37 -		0,49 -12		-	
[0.25-0.50 [0.50-0.75 [0.75-1.00	median	Permutation.png Permutation.png Permutation.png	-	0,230 0,330 0,360	0,43 129 0,43 219 0,37 549		0,35	21% 0,400 -6% 0,330 36% 0,300	0,33 - 0,40 - n/appl	-	0,49 -6 0,48 -10 0,50 -6	% 0,430	-	
diversity lstm - (0.25-0.50		Permutation.png Permutation.png		0,100 0.230	0,39 109			30% 0,30 38% 0,29	- 0,3 - 0,4 -	-	0,46 2			
[0.50-0.75 [0.75-1.00	median	Permutation.png Permutation.png	-	0,350 0,420	0,48 69 n/appl -		0,53 -	45% 0,29 27% 0,40	0,5 - n/appl	-	0,46 4 0,49 -4	% 0,480	-	
LE 3 In attentio <u>vanilla Istm</u>		Source of info rationale_summary_t		0,742	0,472 1059	0,968		- 2% 0,430	0,35 -		0,63 -2		0,28 18	9%
ortho lstm		rationale_summary_t	e 0,240 200%	0,719 0,354	0,217 3249	0,920	0,17 1	0,380 0,535	0,16		0,22 173	0,588	0,23 26	0%
diversity lstm		rationale_summary_t	• 0,624 -12%	0,102 0,550	0,76 20%	0,269	0,57	0,114	- 0,40 -		0,88 6	0,240	- 0,76 -2	3%
.E 4	Rationale length	rationale_summary_t	e 0,175 3%	0,180	0,17 339	0,224	0,16	1 <mark>9%</mark> 0,191	0,24		0,17 53			4%
parison to vanilla Istm elatons	Overall mean Pea	Attn_Gradient_X_val_ Attn_Gradient_X_val_		0,619	0,80 89	6 0,863	0,55	25% 0,690	0,43 0,19 -		0,72 -29	% 0,513	0,65 -	7%
use std.dev in paper to jud	9 Overall mean Pea		0,62 -19%	0,503	0,68 179	0,793		15% 0,458	0,43 0,19 -		0,65 -40	% 0,390	0,56 -1	4%
	Overall mean JS	Attn_Gradient_X_val	0,10 -10%	0,090	0,09 -119	6 0,080	0,15 -	19% 0,121	0,17		0,15 21	% 0,182		6%
	Std JS div Overall mean JS (Std JS div	Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad	0,12 -18%	0,099	0,02 0,13 -139 0,02	0,114	0,04 0,19 0,05	-7% 0,177	0,04 - 0,21 0,06	-	0,07 0,21 6 0,06	% 0,222	0,03 0,08	5%
ortho lstm		Attn_Gradient_X_val_		0,726	n/avail -	0,903	n/avail -	0,686	n/avail		-	0,878	n/avail -	
	Overall mean Pea	Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad	-	0,689	n/avail -	0,714	n/avail -	0,668	n/avail		-	0,858	n/avail -	
		Attn_Gradient_X_val_	-	0,143	n/avail -	0,126	n/avail -	0,163	n/avail			0,174	n/avail -	
		Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad		0,148	n/avail -	0,183	n/avail -	0,170	n/avail		-	0,152	n/avail -	
diversity lstm		Attn_Gradient_X_val_ Attn_Gradient_X_val_		0,849	0,89 39	6 0,917	0,79	1 <mark>8%</mark> 0,645	0,77 0.14 -	-	0,96 -5	% 0,914	0,80	3%
	Overall mean Pea	Attn_Integrated_Grad	0,79 3%	0,815	0,78 19	6 0,789	0,79	29% 0,559	0,77		0,67 29	% 0,867	0,74 -	5%
	Overall mean JS	Attn_Integrated_Grad Attn_Gradient_X_val_	0,08 18%	0,094	0,07 0,09 179	6 0,105	0,14 0,13	33% 0,173	0,14 - 0,12		0,11 0,06 180	0,168		4%
	Std JS div Overall mean JS o Std JS div	Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad	0,09 12%	0,101	0,01 0,13 139 0,02	6 0,147	0,04 0,19 0,05	-2% 0,187	0,04 - 0,12 0,04		0,04 0,07 91 0,05	% 0,134	0,07 0,15 -3 0,06	1%
RE 5 ibution of vanilla lstm		Source of info quant_pos_attn.png	(NOUN.	(NOUN,			[PUNC,I	ÍNOUN.			[NOUN,	(NOUN,	NOUN,	1
<u> </u>	%attention PUN0 %attention ADJ	:	0% 20%	0%		0% 28%	28%	0% 17%			2%	3%	10%	0
ortho lstm	Order of POS pre %attention PUN	quant_pos_attn.png		0%		0%		[NOUN,				[NOUN, 0%		[1
diversity lstm	%attention ADJ	quant_pos_attn.png	ÍNOLIN	25% (NOUN.		35%	(ADJ.NC	- INOUN.			INOUN.	10%	ÍNOUN.	[]
<u>urren sky istill</u>	%attention PUN %attention ADJ		0 28%	0% 25%		0% 35%	3% 30%	0% 17%			0% 10%	0% 8%	3% 13%	U
%Change	%attention PUN %attention ADJ	2	40%	47%		25%	76%	0%			-9%	0%	8%	

Figure 2: Replication results for single sequence tasks

			l experiment	SNLI		QQP		Babi-1		Babi-2		Babi-3		CNN	
	MODEL	Indicator	Source of info	PAPER delta rel%	TEST RU AVG	PAPER delta rel%	TEST RU AVG	PAPER delta rel%		PAPER delta rel%	TEST RU AVG		TEST RU AVG	PAPER delta rel%	
ABLE 2 enchmark da	vanilla Istm ta	test accuracy bes conicity_mean conicity_std	evaluate.json	0,782 -1% 0,560 5% n/avail -	0,773 0,589 0,079	0,787 0% 0,590 -1% n/avail -	0,784	0,991 19 0,560 379 n/avail -		0,401 36% 0,480 -10% n/avail -	0,544 0,432 0,048	0,477 -56% 0,430 116% n/avail -	0,211 0,927 0,044	0,631 -69 0,450 -129 n/avail -	
		test accuracy bes conicity_mean conicity_std	evaluate.json evaluate.json evaluate.json	0,765 0% 0,270 13% n/avail -	0,766 0,306 0,071	0,786 0% 0,330 -4% n/avail -		0,999 09 0,220 59 n/avail -		0,561 5% 0,210 -20% n/avail -	0,590 0,169 0,030	0,512 13% 0,120 12% n/avail -	0,577 0,135 0,018	0,543 -19 0,070 419 n/avail -	
	diversity lstm	test accuracy bes conicity_mean	evaluate.json evaluate.json	0,770 -4% 0,120 <mark>-64%</mark>	0,740 0,043	0,784 0% 0,040 -17%	0,782 0,033	1,000 0 9 0,070 5 9	6 1,000 6 0,073	0,402 36% 0,050 158%	0,546 0,129	0,509 11% 0,100 9%	0,563 0,109	0,582 -209 0,060 5289	<mark>%</mark> %
IGURE 3		conicity_std Indicator	evaluate.json	n/avail -	0,022	n/avail -	0,023	n/avail -	0,023	n/avail -	0,030	n/avail -	0,024	n/avail -	-
	vanilla lstm - ATT		importance_ranking_l importance_ranking_l	MAvDY_all.png MAvDY_all.png	0,330 0,100	1,00 0% 0,66 -39%	1,000 0,400	0,06 -339		-	0,020 0,010	-	0,030 0,010	-	
sual inspection	on paper RANDOM	3rd quartile	importance_ranking_	MAvDY_all.png	0,100 0,860 0.800	1,00 0%	1,000	0,10 -409		-	0,010	-	0,010	-	
		1st quartile 3rd quartile	importance_ranking_ importance_ranking_ importance_ranking_	MAvDY_all.png	0,800 0,530 1,000	1,00 0% 0,85 -18% 1,00 0%	0,700	0,90 -679	6 0,300 6 0,700	-	0,330 0,300 0,800	-	0,330 0,250 0,850	-	
	orth lstm - ATTN		importance_ranking_		0,300	0,60 17%	0,700	0,04 09		-	0,020	-	0,010		
		1st quartile 3rd quartile	importance_ranking_ importance_ranking_	MAvDY_all.png	0,100	0,20 25% 1.00 0%		0,01 1009	6 0,020	-	0,010	-	0,000	-	
	RANDOM		importance_ranking_ importance_ranking_	MAvDY_all.png	0,750 0,470	1,00 0% 0,66 -9%		0,45 119 0,25 209			0,550 0,300	-	0,500 0,250	-	
		3rd quartile	importance_ranking_		1,000	1,00 0%		0,23 207		-	0,800	-	0,250	-	
	diversity lstm - A	median 1st guartile	importance_ranking_importance_ranking_		0,460 0,100	0,80 9% 0,50 10%	0,870 0,550	0,04 09 0,01 1009			0,020 0,010	-	0,020 0,010		
	RANDOM	3rd quartile	importance_ranking_ importance_ranking_	MAvDY_all.png	0,820 0,820	1,00 0%		0,08 -259	6 0,060 6 0,500		0,030 0,550	-	0,030 0,500	-	
		1st quartile 3rd quartile	importance_ranking_ importance_ranking_	MAvDY_all.png	0,500	0,80 -6% 1,00 0%	0,750	0,35 -149		-	0,300 0,800	-	0,250 0,750	-	
GURE 4		Indicator	Source of info							_		_			
omparison iolin plots	vanilla lstm - [0.0 [0.25-0.50]		Permutation.png Permutation.png	-	-	-	0,070 0,100	-	- 0,820	-	0,800 0,920	-	0,500 0,650	-	
sual inspection	[0.50-0.75] [0.75-1.00]		Permutation.png Permutation.png	-	-	-	0,100 0,100	-	0,870 0,900	-	0,920 0,970	-	0,650 0,700	-	
	ortho lstm - [0.0	median	Permutation.png	-	-	-	0,070	-	0,400		0,750	-	0,700	-	
	[0.25-0.50] [0.50-0.75]		Permutation.png Permutation.png	-	-	-	0,100 0,100	-	0,750 0,980	-	0,900 0,900	-	0,750 0,850	-	
	[0.75-1.00]		Permutation.png	<u> </u>	-	-	0,150		0,980		0,950	-	0,950	-	-
	diversity lstm - [0 [0.25-0.50]	median	Permutation.png Permutation.png	-	-	-	0,070 0,170	-	- 0,850	-	0,820 0,950	-	0,750 0,850	-	
	[0.50-0.75] [0.75-1.00]		Permutation.png Permutation.png	-	-	-	0,250 0,200	-	0,870 0,870	-	0,970 0,970	-	0,900 0,950	-	
BLE 3 ean attentio		Indicator Rationale attenti	Source of info rationale_summary_t	act tyt		_									
			rationale_summary_t	est.txt -		-	-	-	-	-	-	-	-	-	-
			rationale_summary_t rationale_summary_t			-	2	-	-	-	-	-	-	-	1
			rationale_summary_t			-	-	-	-		-	-	-	-	-
		Rationale length	rationale_summary_t	est.txt -		-	-	-	-	-	-	-		-	
ABLE 4		Indicator					0.440	0.56 719	0,959						%
mparison to	vanilla lstm		Source of info Attn_Gradient_X_val_ Attn_Gradient_X_val		0,556	0,19 116%	0,410		0,555	0,16 294%	0,630	0,39 21%	0,470	0,58 359	-
omparison to orrelatons	v in paper to judg	Overall mean Pea Std Pearson corr Overall mean Pea	Attn_Gradient_X_val_ Attn_Gradient_X_val_ Attn_Integrated_Grad	0,33 0,38 1%	0,556 0,384	0,34 -0,06 <mark>-240%</mark>	0,084	0,34 0,33 1389	0,333 0,785	0,23 0,05 432%	0,630 0,266	0,24 -0,01 #####	0,470 0,362	0,25 0,45 429	_
omparison to orrelatons	: vanilla lstm v in paper to judg	Overall mean Pea Std Pearson corr Overall mean Pea Std Pearson corr Overall mean JS o	Attn_Gradient_X_val_ Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad Attn_Gradient_X_val_	0,33 0,38 1% 0,40 0,11 4%	0,384	0,34 -0,06 -240% 0,34 0,15 -33%		0,34 0,33 1389 0,37 0,33 -649	0,785	0,23 0,05 432% 0,22 0,53 -27%		0,24 -0,01 ##### 0,08 0,46 -43%		0,25 0,45 429 0,28 0,22 -299	%
omparison to orrelatons	: <u>vanilla Istm</u> v in paper to judg	Overall mean Pea Std Pearson corr Overall mean Pea Std Pearson corr Overall mean JS of Std JS div Overall mean JS of	Attn_Gradient_X_val_ Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad Attn_Gradient_X_val_ Attn_Gradient_X_val_ Attn_Integrated_Grad	0,33 0,38 1% 0,40 0,11 4% 0,07 0,16 -10%	0,384	0,34 -0,06 -240% 0,34 0,15 -33% 0,08 0,19 -22%	0,084	0,34 0,33 138 9 0,37 0,33 -649 0,12 0,43 -439	0,785	0,23 0,05 432% 0,22 0,53 -27% 0,09 0,58 -12%	0,266	0,24 -0,01 ##### 0,08 0,46 -43% 0,08 0,64 -46%	0,362	0,25 0,45 429 0,28 0,22 -299 0,07 0,30 -258	% %
omparison to orrelatons /e use std.de	: <u>vanilla Istm</u> v in paper to judg	Overall mean Pea Std Pearson corr Overall mean Pea Std Pearson corr Overall mean JS of Std JS div Overall mean JS of Std JS div	Attn_Gradient_X_val_ Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad Attn_Gradient_X_val_ Attn_Gradient_X_val_	0,33 0,38 1% 0,40 0,11 4% 0,07 0,16 -10% 0,09	0,384 0,114	0,34 -0,06 <mark>-240%</mark> 0,34 0,15 <mark>-33%</mark> 0,08	0,084 0,101	0,34 0,33 1389 0,37 0,33 -649 0,12	0,785 0,120	0,23 0,05 432% 0,22 0,53 -27% 0,09	0,266 0,389	0,24 -0,01 ###### 0,08 0,46 -43% 0,08	0,362 0,263	0,25 0,45 429 0,28 0,22 -299 0,07	% %
omparison to orrelatons /e use std.de	: <u>vanilla Istm</u> v in paper to judg <u>ortho Istm</u>	Overall mean Pea Std Pearson corr Overall mean Pea Std Pearson corr Overall mean JS o Std JS div Overall mean JS o Std JS div Overall mean Pea Std Pearson corr	Attn_Gradient_X_val_ Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad Attn_Gradient_X_val_ Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad Attn_Gradient_X_val_ Attn_Gradient_X_val	0,33 0,38 1% 0,40 0,11 4% 0,07 0,16 -10% 0,09 0/svail - 0/svail -	0,384 0,114 0,143	0,34 -0,06 -240% 0,34 0,15 -33% 0,08 0,19 -22%	0,084 0,101 0,148 0,378	0,34 0,33 138 9 0,37 0,33 -649 0,12 0,43 -439	0,785 0,120 0,247	0,23 0,05 432% 0,22 0,53 -27% 0,09 0,58 -12%	0,266 0,389 0,510	0,24 -0,01 ##### 0,08 0,46 -43% 0,08 0,64 -46%	0,362 0,263 0,348	0,25 0,45 429 0,28 0,22 -299 0,07 0,30 -258	%
omparison to orrelatons /e use std.de	: <u>vanilla Istm</u> v in paper to judg v	Overall mean Pea Std Pearson corr Overall mean Pea Std Pearson corr Overall mean JS of Std JS div Overall mean Pea Std Pearson corr Overall mean Pea Std Pearson corr	Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad Attn_Integrated_Grad Attn_Integrated_Grad Attn_Integrated_Grad Attn_Integrated_Grad Attn_Integrated_Grad Attn_Integrated_Grad Attn_Integrated_Grad Attn_Integrated_Grad	0,33 0,38 1% 0,40 0,11 4% 0,07 0,16 -10% 0,09 0/4xvii 0,09 0/avvii 0/avvii 0/avvii 1 0/avvii 1 0/avvii	0,384 0,114 0,143 0,506 0,365 sonr.csv	0,34 -0,06 -240% 0,34 0,15 -33% 0,08 0,19 -22%	0,084 0,101 0,148 0,378 0,238	0,34 0,33 1389 0,37 0,33 649 0,12 0,43 439 0,13 n/avail - n/avail -	 0,785 0,120 0,247 0,867 0,709 	0,23 0,05 432% 0,22 0,53 -27% 0,09 0,58 -12%	0,266 0,389 0,510 0,709 0,227	0,24 -0,01 ##### 0,08 0,46 -43% 0,08 0,64 -46%	0,362 0,263 0,348 0,665 0,044	0,25 0,45 429 0,28 0,22 -299 0,07 0,30 259 0,10 n/avail -	%
omparison to orrelatons 'e use std.de	: <u>vanilla Istm</u> v in paper to judg <u>ortho Istm</u>	Overall mean Pei Std Pearson corr Overall mean Pei Std Pearson corr Overall mean Sei Std JS div Overall mean JS of Std JS div Overall mean Pei Std Pearson corr Overall mean Pei Std Pearson corr Overall mean Pei Std Pearson corr Overall mean Pei Std Pearson corr	Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad Attn_Integrated_Grad Attn_Gradient_X_val_ Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Integrated_Grad Attn_Gradient_X_val_ Attn_Gradient_X_val_ Attn_Gradient_X_val_	0,33 0,38 0,14 0,11 0,15 0,16 0,07 0,07 0,07 0,07 0,09 n/avail - n/a - n/avail - n/avail - n/avail - n/avail - n/a - n/a - n/a - n/avail - n/a - n/a - n/ - n/a - n/a - - n/a - n/a - n/a - - - n/a - - - - - - - - - - - - - - - - - - -	0,384 0,114 0,143 0,506 0,365 5007.csv 0,120 0,146	0,34 -0,06 <mark>-240%</mark> 0,34 0,15 - 33% 0,08 0,19 -22% 0,10 n/avail -	0,084 0,101 0,148 0,378	0,34 0,33 138 9 0,37 0,33 -649 0,12 0,43 -439	0,785 0,120 0,247 0,867	0,23 0,05 0,22 0,33 0,09 0,58 -12% 0,09 0,69 0,09	0,266 0,389 0,510 0,709	0,24 -0,01 ##### 0,46 -439 0,46 -439 0,08 0,64 -45% 0,05 n/avail -	0,362 0,263 0,348 0,665	0,25 0,45 429 0,28 0,22 -299 0,07 0,30 -259 0,10 0/avail - 0/avail -	%
omparison to prrelatons /e use std.der	: <u>vanilla Istm</u> v in paper to judge <u>ortho Istm</u>	Overall mean Peis Std Pearson corr Overall mean JS of Std JS div Overall mean JS of Std JS div Overall mean JS of Std JS div Overall mean Peis Std Pearson corr Overall mean Peis Std Pearson corr Overall mean JS of Std JS div Overall mean JS of Std JS div	Atth, Gradient, X. val. Atth, Gradient, X. val. Atth, Integrated, Grad. Atth, Integrated, Grad. Atth, Gradient, X. val. Atth, Integrated, Grad. Atth, Integrated, Grad. Atth, Integrated, Grad.	0,33 0,38 0,40 0,11 0,14 0,07 0,07 0,07 0,07 0,07 0,07 0,09 n/avail - 0earson:csv 0,14 0,16 -10% 0,09 n/avail - 0,14 0,10 0,10 -10% 1,00 0,09 0,10 -10% 1,00 0,09 0,10 -10% 1,00 0,09 0,10 -10% 1,00 0,09 0,10 -10% 1,00 0,09 0,10 -10% 1,00 0,09 0,10 -10% 1,00% -10% 1,00% -10% 1,00% -10%	0,384 0,114 0,143 0,506 0,365 sonr.csv 0,120 0,146	0,34 -0,06 -240% 0,34 0,15 -33% 0,09 -22% 0,10 - n/avai - n/avai - n/avai - n/avai -	0,084 0,101 0,148 0,378 0,238 0,120 0,140	0,34 0,33 0,37 0,33 0,43 0,43 0,13 n/avail - n/avail - n/avail -	0,785 0,120 0,247 0,867 0,709 0,212 0,282	0,23 0,05 0,22 0,53 0,53 0,58 -12% 0,09 n/avail - n/avail - n/avail - n/avail -	0,266 0,389 0,510 0,709 0,227 0,384 0,540	0,24 -0,01 anaur 0,08 0,46 -43% 0,64 -43% 0,64 -46% 0,05 π/avail - π/avail - π/avail -	0,362 0,263 0,348 0,665 0,044 0,426 0,639	0,25 0,45 0,28 0,22 0,07 0,30 0,10 n/avail n/avail n/avail - n/avail	%
omparison to prrelatons 'e use std.de	vanilla lstm v in paper to judge ortho lstm <u>diversity lstm</u>	Overall mean Pey Std Pearson corr Overall mean JS Std Pearson corr Overall mean JS Std JS div Overall mean JS Std JS div Overall mean Pey Std Pearson corr Overall mean Ps Std Pearson corr Overall mean JS Std JS div Overall mean JS Std JS div Overall mean Ps Std JS div Overall mean Ps Std JS div	Atth, Gradient, X, vol. Atth, Gradient, X, vol. Atth, Integrated, Grad Atth, Gradient, X, vol. Atth, Gradient, X, vol.	0,33 10,38 10,40 0,11 4% 0,07 10,16 -10% 10,00 10	0,384 0,114 0,143 0,506 0,365 5007.csv 0,120 0,146 50 0,498	0,34 -0,06 (240%) 0,34 0,15 (33%) 0,08 0,19 (22%) 0,10 n/avai - n/avai - n/avai - n/avai - n/avai -	0,084 0,101 0,148 0,378 0,238 0,120 0,140 0,494	0,34 0,33 1385 0,37 0,33 0,12 0,43 0,13 n/avail - n/avail - n/avail - n/avail - 0,91 -79 0,10	 0,785 0,120 0,247 0,867 0,709 0,212 0,282 0,850 	0,23 0,05 4325 0,03 7,33 0,09 0,58 -12% 0,09 7,4xati - 7,4xati - 7,4xati - 7,4xati - 7,4xati - 7,4xati - 7,4xati -	0,266 0,389 0,510 0,709 0,227 0,384 0,540 0,612	0,24 -0,01 π10011 0,08 0,64 -232 0,08 0,64 -232 0,05 π/avail - π/avail - π/avail - π/avail - π/avail -	0,362 0,263 0,348 0,665 0,044 0,426 0,639 0,736	0,25 0,45 0,28 0,22 0,22 0,20 0,00 0,00 0,00 0,10 0,25 0,10 0,20	%
omparison to orrelatons /e use std.der	<u>vanilla Istm</u> v in paper to judge <u>ortho Istm</u> <u>diversity Istm</u>	Overall mean Peis Std Pearson corr Overall mean Peis Std JS div Overall mean JS of Std JS div Overall mean JS of Std JS div Overall mean Peis Std Pearson corr Overall mean Peis Std JS div Overall mean JS of Std JS div Overall mean Peis Std JS div Overall mean Peis Std Pearson corr Overall mean Peis Std Pearson corr Overall mean Peis Std Pearson corr	Atth, Gradient X, vol. Atth, Gradient X, vol. Atth, Integrated, Grad Atth, Gradient X, vol. Atth, Gradient X, vol.	0,33 10,38 10,40 0,11 0,07 0,16 10,06 10,09 10,09 10,09 10,09 10,09 10,09 10,09 10,09 10,00	0,384 0,114 0,143 0,506 0,365 0,165 0,120 0,146 0,498 0,274	0.34 -0.06 2409 0.34 0.15 338 0.08 0.19 228 0.19 228 0.19 7 7/2021 - 1 7/2021 - 1 7/2021 - 1 7/2021 - 1 7/2021 - 1 7/2021 - 1 7/2021 - 1 7/2021 - 7 7/2021 - 7 7/20 7/2021 - 7 7/2021 - 7 7/20 7/20 7/20 7/20 7/20 7/20 7/20 7/	0,084 0,101 0,148 0,378 0,238 0,120 0,140 0,494 0,116	0.34 0.33 0.37 0.37 0.37 0.32 0.32 0.12 0.12 0.12 0.12 0.12 0.12 0.12 0.1	 0,785 0,120 0,247 0,867 0,709 0,212 0,282 0,850 0,824 	0,23 0,05 4225 0,33 0,09 0,09 0,58 -12% 0,09 0,58 -12% 0,09 0,58 -12% 0,09 0,13 0,73 0,13 0,75 33%	0,266 0,389 0,510 0,709 0,227 0,384 0,540 0,612 0,771	0,24 -0,01 memii 0,08 0,64 -23% 0,08 0,64 -23% 0,05 r/avai - r/avai - r/avai - r/avai - r/avai - r/avai -	0,362 0,263 0,348 0,665 0,044 0,426 0,639 0,736 0,265	0,25 0,45 0,28 0,22 0,07 0,00 n/ovail	%
omparison to prrelatons 'e use std.de	<u>vanilla Istm</u> vin paper to judge <u>ortho Istm</u> <u>diversity Istm</u>	Overall mean Pe Std Peason core Verall mean Pe Std Sd verall mean Si Std Sd Verall mean Si Std Sd Verall mean Pe Std Sd Verall mean Pe Std Sd Verall mean Pe Std Sd Verall mean Pe Std Sd Verall mean Si Std Sd Verall mean Pe Std Pason core Verall mean Pe	Atth, Gradient X, vol. Atth, Gradient X, vol. Atth, Integrated, Grad Atth, Gradient X, vol. Atth, Gradient X, vol.	0.33 0.36 0.40 0.11 0.45 0.40 0.11 0.45 0.47 0.15 0.45 0.45 0.45 0.45 0.45 0.45 0.45 0.4	0,384 0,114 0,143 0,506 0,365 0,120 0,120 0,146 0,498 0,274 0,109	0.34 -0.06 (240%) 0.15 (33%) 0.19 (22%) 0.10 (27%) 0.10 (27%) 1/2v21 - 1/2v21 -	0,084 0,101 0,148 0,378 0,238 0,120 0,120 0,140 0,494 0,116 0,110	0.34 0.33 0.37 0.37 0.37 0.33 0.42 0.12 0.12 0.43 0.43 0.43 0.444 0.443 0.444	0,785 0,120 0,247 0,867 0,709 0,212 0,282 0,282 6 0,824 6 0,824	0.23 0.05 4728 0.22 4728 0.30 4728 0.30 4728 0.30 4728 0.30 4728 0.30 4728 0.424 0.4444 0.44444 0.44444 0.44444 0.44444 0.44444 0.44444 0.444444 0.444444 0.4444444 0.44444444	0,266 0,389 0,510 0,709 0,227 0,384 0,540 0,612 0,771 0,396	0,24 -0,01 mmmf 0,46 0.33% 0,64 0.5% 0,64 0.6% 0,65 1/2v01 - 1/2v01 - 1/2v0	0,362 0,263 0,348 0,665 0,044 0,426 0,639 0,736 0,265 0,359	0.25 0.45 0.28 0.22 0.22 0.20 0.20 0.20 0.20 0.20	% % %
omparison to prrelatons 'e use std.de	<u>vanilla istm</u> v in paper to judg <u>ortho Istm</u> <u>diversity Istm</u>	Overall mean Pe Std Peason core Verall mean Pe Std Sd verall mean Si Std Sd Verall mean Si Std Sd Verall mean Pe Std Sd Verall mean Pe Std Sd Verall mean Pe Std Sd Verall mean Pe Std Sd Verall mean Si Std Sd Verall mean Pe Std Pason core Verall mean Pe	Atth, Gradient, X. vol. Atth, Gradient, X. vol. Atth, Integrated, Grad. Atth, Gradient, X. vol. Atth, Gradient, X. vol.	0,33 0,38 0,40 0,11 0,16 0,07 0,16 0,09 0,16 0,09 0,16 0,09 0,09 0,09 0,09 0,09 0,09 0,09 0,0	0,384 0,114 0,143 0,506 0,365 0,165 0,120 0,146 0,498 0,274	0,34 -0.06 (.2405) 0,34 0,15 -336 0,08 0,10 0,10 0,10 0,10 0,10 0,22 0,23 - 0,24 - 0,53 -15% 0,31 - 0,24 - 45% 0,10 0,26 - 45% 0,10 0,10 0,10 - 0,25% - 15% 0,10 - 0,25% - 15%	0,084 0,101 0,148 0,378 0,238 0,120 0,120 0,140 0,494 0,116 0,110	0,34 0,33 0,37 0,33 0,33 0,33 0,33 0,33 0,33	0,785 0,120 0,247 0,867 0,709 0,212 0,282 0,282 6 0,824 6 0,824	0,23 0,05 4,323 0,53 2,24 0,53 2,25 0,53 0,55 7/avail	0,266 0,389 0,510 0,709 0,227 0,384 0,540 0,612 0,771	0,24 -0,01 memeri 0,46 43 9 0,46 43 9 0,65 0,65 0,67 10% 0,72 10% 0,67 10% 0,47 10% 0,47 10% 0,47 30% 0,47 30% 0,48 30%0,48 30% 0,48 30% 0,48 30% 0,48 30%0,48 30% 0,48 30% 0,48 30%0,48 30% 0,48 30% 0,48 30%0,48 30% 0,48 30% 0,48 30%0,48 30%0,48 30% 0,48 30%0,49 30%0,49 30% 0,49 30%0,49 30%0,49 30%0,49 30%0,49 30%	0,362 0,263 0,348 0,665 0,044 0,426 0,639 0,736 0,265	0,25 0,45 0,28 0,22 0,20 0,20 0,00 0,00 0,00 0,00	%
omparison to orrelatons /e use std.der	<u>vanilla Istm</u> ortho Istm diversity Istm	Overall mean Pe Std Pearson core Voerall mean Pe Std Pearson core Voerall mean IS Std J Sdiv Overall mean IS Std J Sdiv Overall mean Pe Std Pearson corr Overall mean Pe Std Sdiv Overall mean Pe	Atth, Gradient X, vol. Atth, Gradient X, vol. Atth, Integrated, Grad. Atth, Integrated, Grad. Atth, Gradient X, vol. Atth, Gradient X, vol.	0,33 0,38 0,40 0,40 0,11 0,40 0,11 0,15 0,10 0,15 0,07 0,10 0,10 0,10 0,10 0,10 0,10 0,10	0,384 0,114 0,143 0,506 0,365 500r.csv 0,120 0,146 0,498 0,274 0,109 0,144	0.34 0.34 0.4 0.4 0.4 0.5 0.5 0.5 0.5 0.5 0.15 0.5 0.5 0.7 0.10 0.00 0.10 0.00 0.10 0.00	0,084 0,101 0,148 0,238 0,120 0,140 0,494 0,116 0,110 0,141	0.34 0.33 0.33 0.33 0.33 0.33 0.33 0.33	0,785 0,120 0,247 0,867 0,709 0,212 0,282 0,850 0,850 0,850 0,824 0,226 6 0,221	0.25 0.05 0.25 0.25 0.25 0.29 0.99 0	0,266 0,389 0,510 0,709 0,227 0,384 0,540 0,612 0,771 0,396 0,582	0,24 -0,0,1 -0,0,1 -0,0,1 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,0,5 -0,0,0,0,5 -0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	0,362 0,263 0,348 0,665 0,044 0,426 0,639 0,736 0,265 0,359 0,641	0,25 0,46 (32,22 0,28 (29) 0,27 (29) 0,20 (29) 0,20 (29) 0,07 (29) 7/(29) (29) (29) 7/(29)	% % % %
omparison to orrelatons /e use std.der	<u>vanilla Istm</u> vin paper to judge <u>ortho Istm</u> <u>diversity Istm</u>	Overall mean Pe Std Peason core Voerall mean Pe Std Peason core Voerall mean JS Std Sdiv Overall mean JS Std Peason core Overall mean Pe Std Pearson core Overall mean Pe	Attn. Gradient X, vol. Attn. Gradient X, vol. Attn. Integrated. Grad Attn. Gradient X, vol. Attn. Gradient X, vol.	0,33 0,38 0,40 0,40 0,11 0,40 0,11 0,15 0,10 0,15 0,07 0,10 0,10 0,10 0,10 0,10 0,10 0,10	0,384 0,114 0,143 0,506 0,365 500r.csv 0,120 0,146 0,498 0,274 0,109 0,144	0.34 0.34 0.4 0.4 0.4 0.5 0.5 0.5 0.5 0.5 0.15 0.5 0.5 0.7 0.10 0.00 0.10 0.00 0.10 0.00	0,084 0,101 0,148 0,378 0,238 0,120 0,120 0,140 0,494 0,116 0,110	0.34 0.33 0.33 0.33 0.33 0.33 0.33 0.33	0,785 0,120 0,247 0,867 0,709 0,212 0,282 0,282 6 0,824 6 0,824	0.25 0.05 0.25 0.25 0.25 0.29 0.99 0	0,266 0,389 0,510 0,709 0,227 0,384 0,540 0,612 0,771 0,396	0,24 -0,0,1 -0,0,1 -0,0,1 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,5 -0,0,0,5 -0,0,0,0,5 -0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	0,362 0,263 0,348 0,665 0,044 0,426 0,639 0,736 0,265 0,359	0,25 0,46 (32,22 0,28 (29) 0,27 (29) 0,20 (29) 0,20 (29) 0,07 (29) 7/(29) (29) (29) 7/(29)	% % % %
	<u>vanilla Istm</u> ortho Istm diversity Istm vanilla Istm ortho Istm.	Overall mean Pe Std Peason core Voerall mean Pe Std Peason core Voerall mean JS Std Sdiv Overall mean JS Std Peason core Overall mean Pe Std Pearson core Overall mean Pe	Atth, Gradient X, vol. Atth, Gradient X, vol. Atth, Integrated, Grad Atth, Gradient X, vol. Atth, Gradient X, vol.	0,33 0,38 0,40 0,40 0,11 0,40 0,11 0,15 0,10 0,15 0,07 0,10 0,10 0,10 0,10 0,10 0,10 0,10	0,384 0,114 0,143 0,506 0,365 500r.csv 0,120 0,146 0,498 0,274 0,109 0,144	0.34 0.34 0.4 0.4 0.6 0.7 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.7 0.15 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7	0,084 0,101 0,148 0,378 0,238 0,120 0,140 0,494 0,116 0,110 0,141 NOUN, 9%	0.34 0.33 0.33 0.33 0.33 0.33 0.33 0.33	0,785 0,120 0,247 0,867 0,709 0,212 0,282 0,282 0,850 0,850 0,850 0,850 0,824 0,226 0,221	0.25 0.05 0.25 0.25 0.25 0.29 0.99 0	0,266 0,389 0,510 0,709 0,227 0,384 0,540 0,540 0,540 0,540 0,540 0,542 0,582 (NOUN) 1%	0,24 -0.01 (1996) 0,06 (1996) 0,46 (1996) 0,46 (1996) 0,47 (1996) 0,47 (1996) 0,47 (1996) 0,47 (1996) 0,47 (1996) 0,47 (1996) 0,47 (1996) 0,47 (1996) 0,41 (1996) 0,98 (1996)	0,362 0,263 0,348 0,665 0,044 0,426 0,639 0,736 0,265 0,359 0,641 (NOUN, 8% 0% 0%	0,25 0,46 (32,22 0,28 (29) 0,27 (29) 0,20 (29) 0,20 (29) 0,07 (29) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(20) 7/(20) (20) 7/(20	% % %
omparison tc orrelatons vie use std.der	<u>vanilla Istm</u> ortho Istm diversity Istm <u>vanilla Istm</u> <u>ortho Istm.</u>	Overall mean Pe Std Peason.core Voerall mean Pe Std Peason.core Voerall mean S. Std J Sdiv Overall mean IS. Std J Sdiv Overall mean Pe Std Peason.corr Overall mean Peason.cor Overall mean Peason.corr Ove	Atth, Gradient, X.vol. Atth, Gradient, X.vol. Atth, Integrated, Grad. Atth, Gradient, X.vol. Atth, Gradient, X.vol	0,33 0,38 0,40 0,40 0,11 0,40 0,11 0,15 0,10 0,15 0,07 0,10 0,10 0,10 0,10 0,10 0,10 0,10	0,384 0,114 0,143 0,506 0,365 0,365 0,320 0,140 0,120 0,146 0,274 0,109 0,144 (NOUN,	0.34 0.34 0.62 207 0.76 207 0.76 207 0.76 207 0.76 207 0.77 207 0.77 207 0.77 207 0.77 207 0.77 207 0.77 207 0.75	0,084 0,101 0,148 0,378 0,238 0,120 0,140 0,140 0,110 0,110 0,141 (NOUN, 9% 9% (NOUN, 9%	0.34 0.33 0.33 0.33 0.33 0.33 0.33 0.33	0,785 0,120 0,247 0,867 0,709 0,212 0,282 0,282 0,850 6 0,850 6 0,824 6 0,221 0,226 6 0,221	0.25 0.05 0.25 0.25 0.25 0.29 0.99 0	0,266 0,389 0,510 0,709 0,227 0,384 0,540 0,612 0,396 0,582 (NOUN) 1% (NOUN)	0.24 0.0.1 [000] 0.08 0.46 0.56 0.64 0.64 0.64 0.65 0.67 0.72 0.72 0.75	0,362 0,263 0,348 0,665 0,044 0,426 0,639 0,736 0,736 0,255 0,359 0,641 (NOUN, 8% 0%	0,25 0,46 (32,22 0,28 (29) 0,27 (29) 0,20 (29) 0,20 (29) 0,07 (29) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(20) 7/(20) (20) 7/(20	8 8 8 8 8 8 8 8 8 8 8 8 1 1
omparison to orrelatons /e use std.der	<u>vanilla Istm</u> ortho Istm diversity Istm <u>vanilla Istm</u> ortho Istm. diversity Istm	Overall mean Pe Std Peason.core Voerall mean Pe Std Peason.core Voerall mean S. Std J Salv Overall mean S. Std J Salv Overall mean Pe Std Peason.corr Overall mean Peason.c	Attn. Gradient X, vol. Attn. Gradient X, vol. Attn. Integrated. Grad Attn. Gradient X, vol. Attn. Gradient X, vol.	0,33 0,38 0,40 0,40 0,11 0,40 0,11 0,15 0,10 0,15 0,07 0,10 0,10 0,10 0,10 0,10 0,10 0,10	0,384 0,114 0,143 0,506 0,365 0,365 0,320 0,140 0,120 0,146 0,274 0,109 0,144 (NOUN,	0.34 0.34 0.4 0.4 0.6 0.7 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.15 0.7 0.7 0.15 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7	0,084 0,101 0,148 0,378 0,238 0,120 0,140 0,140 0,110 0,110 0,111 (NOUN, 9% (NOUN, 17%	0.34 0.33 0.33 0.33 0.33 0.33 0.33 0.33	0,785 0,120 0,247 0,867 0,709 0,212 0,282 0,282 0,282 0,285 0,850 0,824 0,226 0,221 (NOUN) 0% 0% (NOUN)	0.25 0.05 0.25 0.25 0.25 0.29 0.99 0	0,266 0,389 0,510 0,709 0,227 0,384 0,540 0,540 0,540 0,540 0,542 (NOUN) 1% (NOUN) 1% 0% 0%	0.24 0.0.1 [000] 0.08 0.46 0.56 0.64 0.64 0.64 0.65 0.67 0.72 0.72 0.75	0,362 0,263 0,348 0,665 0,044 0,426 0,639 0,265 0,359 0,265 0,359 0,641 (NOUN, 8% 0% 0%	0,25 0,46 (32,22 0,28 (29) 0,27 (29) 0,20 (29) 0,20 (29) 0,07 (29) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(20) 7/(20) (20) 7/(20	% % % % % % (1
Imparison to irrelatons e use std.der	vanilla Istm ortho Istm diversity Istm vanilla Istm ortho Istm. diversity Istm Schange	Overall mean Pe Std Peason core Voerall mean Pe Std Peason core Voerall mean S Std Sdiv Overall mean S Std Peason core Overall mean S Std Peason core Overall mean Pe Std Peason core Overall mean S Std Peason core Over	Attn, Gradient, X.vol. Attn, Gradient, X.vol. Attn, Integrated, Grad. Attn, Gradient, X.vol. Attn, Gradient, X.vol	0,33 0,38 0,40 0,40 0,11 0,40 0,11 0,15 0,10 0,15 0,07 0,10 0,10 0,10 0,10 0,10 0,10 0,10	0,384 0,114 0,143 0,506 0,365 0,365 0,320 0,140 0,120 0,146 0,274 0,109 0,144 (NOUN,	0.34 0.4 0.6 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7	0,084 0,101 0,148 0,378 0,238 0,120 0,140 0,140 0,110 0,110 0,141 0,141 0,141 0,141	0.34 0.33 0.33 0.33 0.33 0.33 0.33 0.33	0,785 0,120 0,247 0,867 0,709 0,212 0,282 4 0,282 4 0,282 4 0,282 4 0,282 4 0,282 4 0,221 0% (NOUN) 0% 0% (NOUN) 0%	0.25 0.05 0.25 0.25 0.25 0.29 0.99 0	0,266 0,389 0,510 0,709 0,227 0,384 0,540 0,540 0,540 0,540 0,540 0,542 (NOUN) 1% 0% 0% 0% 0%	0.24 0.0.1 [000] 0.08 0.46 0.56 0.64 0.64 0.64 0.65 0.67 0.72 0.72 0.75	0,362 0,263 0,348 0,665 0,044 0,426 0,639 0,736 0,265 0,359 0,641 (NOUN, 8% 0% 0% 0% 0% 0%	0,25 0,46 (32,22 0,28 (29) 0,27 (29) 0,20 (29) 0,20 (29) 0,07 (29) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(200) (20) 7/(20) 7/(20) (20) 7/(20	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

Figure 3: Replication results for dual sequence tasks

332 Appendix C: Results of BiLSTM extension

Color coding: Relative differe	ence vs. Vanilla Bi-	LSTM		dataset imdb			dataset 20News			dataset QQP			dataset Babi-1		
TABLE 2	ted improvement MODEL <u>vanilla Istm</u>	Indicator test accuracy bes		0,895		TEST RU AVG 0,893	PAPER 0,936	delta rel%	TEST RU AVG 0,908	PAPER 0,787	delta rel%	TEST RU AVG 0,784	PAPER 0,991	delta rel%	TEST RU AVG 1,000
Benchmark dat	a	conicity_mean conicity_std	evaluate.json evaluate.json	0,690 n/avail		0,602 0,135	0,770 n/avail		0,761 0,189	0,590 n/avail		0,583 0,131	0,560 n/avail		0,766 0,023
	<u>bi lstm</u>	test accuracy best conicity_mean conicity_std	evaluate.json evaluate.json evaluate.json			0,905 0,590 0,118			0,919 0,818 0,136			0,790 0,610 0,103		-	0,997 0,692 0,042
	<u>ortho bi Istm</u>	test accuracy best conicity_mean conicity_std	evaluate.json evaluate.json evaluate.json		-1% -70%	0,893 0,176 0,045		-4% <mark>-54%</mark>	0,885 0,373 0,156		-1% <mark>-44%</mark>	0,785 0,341 0,068		0% -68% -	1,000 0,220 0,055
	<u>diversity bi lstm</u>	test accuracy best conicity_mean conicity_std	evaluate.json evaluate.json evaluate.json	-	-1% -76%	0,892 0,144 0,025	-	1% -80%	0,924 0,164 0,074		-1% -94%	0,786 0,035 0,019		0% -93%	1,000 0,046 0,016
FIGURE 3	<u>bi lstm - ATTN</u>	Indicator median 1st quartile	Source of info importance_ranking_N importance_ranking_N	-		0,980 0,670			0,520 0,520			0,830 0,380			0,030 0,020
	RANDOM	3rd quartile median 1st quartile 3rd quartile	importance_ranking_N importance_ranking_N importance_ranking_N importance_ranking_N	-		1,000 0,920 0,920 1,000			1,000 0,980 0,980 1,000		- - -	1,000 0,740 0,740 1,000			0,080 0,600 0,320 0,820
	<u>ortho bi lstm - A'</u>	1st quartile 3rd quartile	<pre>importance_ranking_N importance_ranking_N importance_ranking_N</pre>		-93% -97% -87%	0,070 0,020 0,130		-52% -92% -61%	0,250 0,040 0,390		-19% -39% 0%	0,670 0,230 1,000		-33% -50% -50%	0,020 0,010 0,040
	<u>RANDOM</u>	median 1st quartile 3rd quartile	<pre>importance_ranking_N importance_ranking_N importance_ranking_N</pre>		0% -22% -3%	0,920 0,720 0,970		-3% -17% 0%	0,950 0,810 1,000		-15% -15% 0%	0,630 0,630 1,000		-13% -16% -7%	
	diversity bi lstm RANDOM	1st quartile 3rd quartile	importance_ranking_N importance_ranking_N importance_ranking_N importance_ranking_N		-93% -97% -67% 0%	0,070 0,020 0,330 0,920		-88% -94% -87% -8%	0,060 0,030 0,130 0,900		-40% 32% 0% -1%	0,500 0,500 1,000 0,730		-33% -50% -50% -13%	0,020 0,010 0,040 0,520
		1st quartile 3rd quartile	<pre>importance_ranking_N importance_ranking_N</pre>		-24% -2%	0,700 0,980		-27% -2%	0,720 0,980		-1% 0%	0,730 1,000		-16% -6%	0,270 0,770
FIGURE 4 Comparison Violin plots visual inspectio	vanilla lstm - [0.0 [0.25-0.50] [0.50-0.75] [0.75-1.00]	median median	Source of info Permutation.png Permutation.png Permutation.png Permutation.png	0,16 0,17 n/appl n/appl		0,180 0,200 -	0,05 0,15 0,23 0,28		0,020 0,080 -			0,070 0,100 0,100 0,100			- 0,820 0,870 0,900
	<u>bi lstm - [0.00-0.</u> [0.25-0.50] [0.50-0.75]	median median median	Permutation.png Permutation.png Permutation.png	-		0,100 0,090			0,010 0,200 0,270		- - -	0,020 0,080 0,210		- -	- 0,570 0,920
	[0.75-1.00] ortho bi lstm - [0 [0.25-0.50]	median median	Permutation.png Permutation.png Permutation.png		330% 422%	- 0,430 0,470		4100% 95%	0,730 0,420 0,390		-50%	0,430 0,010 0,080		- 75%	0,970 - 1,000
	[0.50-0.75] [0.75-1.00] diversity bi lstm	median	Permutation.png Permutation.png		300%	0,490 0,420 0.400		59% -44% 4500%	0,430 0,410 0,460	_	-10% -42% 150%	0,190 0,250 0.050		9% 3%	1,000 1,000
	[0.25-0.50] [0.50-0.75] [0.75-1.00]	median median	Permutation.png Permutation.png Permutation.png		367%	0,420 0,520		135% 74% -33%	0,470 0,470 0,490		88% 5% -23%	0,150 0,220 0,330		- -5% -10%	- 0,870 0,870
TABLE 4 Comparison to	<u>vanilla Istm</u>	Overall mean Pea Overall mean Pea Overall mean JS d	Source of info Attn_Gradient_X_val_ Attn_Integrated_Gradi Attn_Gradient_X_val_ Attn_Integrated_Gradi	0,68 0,09		0,863 0,793 0,080 0,114	0,72 0,65 0,15 0,21		0,513 0,390 0,182 0,222	0,19 -0,06 0,15 0,19		0,410 0,084 0,101 0,148	0,56 0,33 0,33 0,43		0,959 0,785 0,120 0,247
	<u>bi Istm</u>	Overall mean Pea Overall mean JS d	Attn_Gradient_X_val_ Attn_Integrated_Gradi Attn_Gradient_X_val_ Attn_Integrated_Gradi			0,817 0,680 0,091 0,147			0,535 0,411 0,116 0,141			0,345 0,018 0,155 0,213			0,867 0,643 0,210 0,343
	<u>ortho bi Istm</u>	Overall mean Pea Overall mean JS d	Attn_Gradient_X_val_ Attn_Integrated_Grad Attn_Gradient_X_val_ Attn_Integrated_Grad		14% 25% -5% -29%	0,930 0,849 0,086 0,105		-35% -4% 258% 174%	0,346 0,394 0,415 0,386		32% 1894% -14% -31%	0,457 0,359 0,133 0,148		<mark>-1%</mark> 15% -5% -15%	0,862 0,738 0,199 0,292
	<u>diversity bi lstm</u>	Overall mean Pea Overall mean JS d	Attn_Gradient_X_val_ Attn_Integrated_Gradi Attn_Gradient_X_val_ Attn_Integrated_Gradi		11% 25% <mark>19%</mark> -16%	0,908 0,847 0,108 0,124		79% 114% -16% -23%	0,960 0,880 0,097 0,108		50% 1756% -34% -39%	0,516 0,334 0,103 0,130		-65% -11% 106% 5%	0,307 0,571 0,433 0,359

Figure 4: Performance of BiLSTM on evaluation metrics

Appendix D: Selected data examples and illustration of model behavior 333

SST examples

SST, Vanilla-LSTM, tr	ained on default	(tanh) atter	ntion (test	acc= <mark>.805</mark> , d	conicity= <mark>.</mark>	. <mark>697</mark>)
<pre>Sentence : <sos> Attentions : 0.00 SM(h_i): 0.00 Label: 1 , Prediction</sos></pre>	0.04 0.08 0.09 0.17	0.20	0.30	horror <mark>0.24</mark> 0.17	0.13	0.00
SST, Ortho-LSTM, trai	ned on default (tanh) attenti	ion (test ac	cc= <mark>.776</mark> , cor	nicity= <mark>.28</mark>	<mark>33</mark>)
<pre>Sentence : <sos> Attentions : 0.00 SM(h_i): 0.00 Label: 1 , Prediction</sos></pre>	0.00 0.03 0.15 0.14	0.64	0.18	0.04	0.11	0.00
SST, Diversity-LSTM,	trained on defau	ult (tanh) att	cention (tes	st acc= <mark>.800</mark> ,	conicity	/= <mark>.188</mark>)
<pre>Sentence : <sos> Attentions : 0.00 SM(h_i): 0.00 Label: 1 , Predictio</sos></pre>	0.00 0.12 0.02 0.04	0.86	0.01	0.01	0.01	0.00

Yelp examples

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

Sentence : <sos></sos>	Been	going	here	for	years.	A	great	place!	<eos></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	n: [0.98	4035]							

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

<pre>Sentence : <sos> Attentions : 0.00 SM(h_i): 0.00 Label: 1 , Prediction</sos></pre>	Been goi 0.53 0.0 0.18 0.0 on: [0.9904576	0 0.00 18 0.05	for 0.00 0.07	years. 0.08 0.04	A 0.00 0.03	great <mark>0.22</mark> 0.34	place! 0.17 0.21	<eos> 0.00 0.00</eos>
Yelp, Diversity-LSTM,	trained on de	fault (tanh)	attention	(test acc=	= <mark>.938</mark> , c	onicity= <mark>.</mark>	. 347)	
<pre>Sentence : <sos> Attentions : 0.00 SM(h_i): 0.00 Label: 1 , Prediction</sos></pre>	Been goi 0.41 0.1 0.18 0.0 on: [0.9969946	.6 0.00 05 0.02	for 0.00 0.04	years. 0.02 0.03	A 0.09 0.06	great <mark>0.21</mark> 0.40	place! 0.10 0.21	<eos> 0.00 0.00</eos>

Figure 5: Examples of single input sequence tasks

CNN examples

CNN, Vanilla-LSTM, trained with default (tanh) attention (test acc=<mark>.595</mark>, conicity=<mark>.395</mark>)

P path Attentic	: <sos< th=""><th>> (0.00</th><th>Centity2</th><th>2) 0.00</th><th></th><th>entity</th><th>1 cit:</th><th></th><th></th><th>killed</th><th>and</th><th>anot 0.00</th><th></th><th>jured</th><th>in 0.00</th><th>what</th></sos<>	> (0.00	Centity2	2) 0.00		entity	1 cit:			killed	and	anot 0.00		jured	in 0.00	what
police	are	calling	a.000	suspected	teri		attack		nesdav	night			@entity6		Renti	
0.00	0.00	0.00	0.00	-	0.00		0.00	0.0		0.00	0.0		0.00		0.07	CT C
spokesma				a 37	_	vear	_	old	entity1		rist	from	@entit		truck	two
0.02	0.00	· · · · ·		0.00 0.00		0.00		0.00	0.43	0.00		0.00	0.03	-	.00	0.00
people	standing			stop	in	the	Gentity	15	section	of		ity		one	victim	
0.00	0.00		.00 0.00	0.00			0.00					.00				0.00
identifi	ed by	police	as <mark>en</mark> t	tity18 ,	26	,	died	at	the	hospita	al .	a	20	-	year	-
0.00		0.00	0.00 0.20	0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0 0.0	0 0.00	0.00	0.00	0.00
old wo	oman rema	ains in	serious	s conditi	.on ,	accord	ding	to	@entit	y7	th	e	driver	has	beer	n
0.00 0.	00.00	0.0	0.00	0.00	0.00	0.00		0.00	0.00		0.00 0.		0.00	0.00	0.0	
arrested	l and i	is unde:	r invest	tigation b	y th	ne <mark>der</mark>	ntity24			from	the	in	vestigat	ion	and	first
0.00	0.00 (0.00 0.00	0.00		.00 0.	.00 0.0	02		.00 0.00	0.00	0.00	Ο.			0.00	0.00
findings	s, tl	nere is		trong sus	picion		we	're	talkin	g al	out	a	terror	atta	ack	,
0.00	0.00 0.	.00 0.00		.00 0.0				0.00	0.00	0.		0.00	0.00	0.00		0.00
	entity7		. am:		ongoing		stigatio			magist		court	has	issu		a
		0.00	0.00 0.0		0.00	0.00			.00 0.00	0.00		0.00	0.00	0.00		0.00
5.5	order 0.00	on de 0.00 0.	tails 00	of the 0.00 0.0		ident	0.00	<eo< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></eo<>								

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= $\frac{.536}{.536}$, conicity= $\frac{.099}{.099}$)

P path Attent	n : <s< b=""> tions : 0.</s<>		<mark>0er</mark> .00 0.1	<mark>ntity2</mark> 19) 0.00	one 0.00	Centity	1 citiz 0.00	en was 0.01	killed			injured 0.00	in 0.02	what 0.00
police	e are	calli	ng a	a s	uspecte	ed te	rror	attack	wednesda	ay night	near	@enti	ty6	. @en	tity8
0.00	0.00	0.00			.00	0.		0.00	0.00	0.01	0.00	0.00			
spokes	sman @en	tity7	said	a	37 -	yea	r <mark>-</mark>	old	@entity10	motoris	t <mark>from</mark>	0entit	y11	struck	two
0.00	0.0		0.03	0.02	0.00 0	.02 0.0	0 0.01	0.00	0.00	0.00	0.01	0.00		0.01	0.00
people	e <mark>standi</mark>	ng at	a	bus	stop	in	the (entity15	section	of <mark>the</mark>	city	. 0	ne v	ictim	,
0.00	0.05	0.1	3 0.06	5 0.00	0.00	0.15	0.01 (0.00	0.04	0.03 0.03	1 0.00	0.02 0	.00 0	.00	0.00
identi	ified by	poli	ce as	@enti	ty18 ,	26	,	died a	it <mark>the</mark>	hospital		<mark>a</mark> 20		year	
0.00	0.0	0 0.00	0.00	0.00		.00 0.0	0.00	0.00 0	.00 0.02	0.00	0.00	0.01 0.	0.00	2 0.00	0.01
old	woman r	emains	in	serio	us co	ondition	,	accordi	ng to	@entity7		the	driver	has	been
0.00	0.00 0	.00	0.00	0.00	0.		0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
arrest		is	under	inve	stigati	on by	the	@entit		" from	the	investi	gation	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
findir	ngs <mark>,</mark>	there	is		strong	suspic	ion that		're	talking	about		rror	attack	,
0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.00	0.00	0.00 0.		0.00	0.00
"	@entity7	said	. a	amid	the	ongoing	invest:	igation	, a	magistr	ate co	urt h	as i	ssued	a
0.00	0.00	0.00	0.01 0	0.00	0.00	0.00	0.00		0.00 0	.00 0.00	0.	00 0	.00 0		0.00
gag	order	on	details	s of	the	e inc	ident	. <	EOS>						
0.00	0.00	0.00	0.00	0.0	0.0	0.0		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