# How (Un)Faithful is Attention?

**Hessam Amini** and **Leila Kosseim**
Computational Linguistics at Concordia (CLaC) Laboratory
Department of Computer Science and Software Engineering
Concordia University, Montreal, Canada
hessam.amini@mail.concordia.ca; leila.kosseim@concordia.ca

## Abstract

Although attention weights have been commonly used as a means to provide explanations for deep learning models, the approach has been widely criticized due to its lack of *faithfulness*. In this work, we present a simple approach to compute the newly proposed metric *AtteFa*, which can quantitatively represent the degree of faithfulness of the attention weights. Using this metric, we further validate the effect of the frequency of informative input elements and the use of contextual vs. non-contextual encoders on the faithfulness of the attention mechanism. Finally, we apply the approach on several real-life binary classification datasets to measure the faithfulness of attention weights in real-life settings.

## 1 Introduction

Attention mechanism (Bahdanau et al., 2015) has become an indispensable part of many state-of-the-art NLP models, and its application is becoming more and more prevalent in non-NLP use cases. In simple words and from a functionality perspective, attention can be described as a module which generates outputs from the representations of input elements by performing the following two steps:

1. Automatically compute weights corresponding to each input element
2. Use the computed weights to run a weighted average over the input representations

Due to attention's explicit mechanism to assign weights to input elements, attention weights have been frequently used as explanations for model predictions. A common approach has been to provide attention heat maps to which input elements the attention component has attended to (e.g. Wang et al., 2016; Lee et al., 2017; Lin et al., 2017; Ghaeini et al., 2018).

However, the use of attention weights as explanations has been widely challenged, with regards to the observation that they are not *faithful*, meaning that different attention weights can result in similar model predictions (Jain and Wallace, 2019; Serrano and Smith, 2019; Wiegreffe and Pinter, 2019). Therefore, the explanations provided by the attention weights are neither unique nor closely related.

In this work, we extend the work of Wiegreffe and Pinter (2019) to a one-shot adversarial setup that can be used to compute a quantitative metric for the faithfulness of attention weights. We call the metric *AtteFa* which simply stands for *Attention Faithfulness*. We consider the adversarial training setup one-shot in the sense that it can provide us with the AtteFa metric by running the adversarial training only once.

To perform a sanity check on AtteFa, we run experiments in a controlled setting using synthetic datasets and two types of encoders (a non-contextual MLP and a contextual LSTM) that could help us validate if the values of this metric reflect what we expect it to. We later compute this metric on some real-life binary text classification datasets to validate how faithful the attention weights are in those settings.

## 2 Related Work

Since the rise of deep learning models, researchers have focused on devising techniques that could provide an explanation for the functioning of these so-called "black-box" models. Among different classes of explainability techniques, the following can be mentioned:

Gradient-based methods attribute model decisions to input features using gradient signals (Sundararajan et al., 2017; Selvaraju et al., 2017; Aubakirova and Bansal, 2016; Karlekar et al., 2018). Perturbation-based methods try to provide an explanation for the model behavior by evaluating its reactions to perturbations in input features (Ribeiro et al., 2016; Zintgraf et al., 2017).



*Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 119 - 130

Attention-based methods act as an intuitive way of interpreting the model's decision. They use the probability distribution or weights provided by an attention mechanism as a feature importance measure to find the features that the model is attending to (Luong et al., 2015; Xie et al., 2017; Mullenbach et al., 2018).

Despite the popularity of the attention-based explainability approaches, the reliability of these methods has been called into question, with the special focus on the faithfulness of the explanations provided by the attention mechanism. Jain and Wallace (2019) perform different experiments to evaluate the meaningfulness of explanations provided by attention weights. Their results show that attention weights are not correlated with gradient-based feature importance scores. Furthermore, they show that it is often possible to have different attention probability distributions that result in a similar output, arguing that a specific distribution cannot be treated as the definitive cause behind a model decision.

Serrano and Smith (2019) investigate the ability of attention weights to act as importance measures through a different lens. They state that it is not sufficient for the weights to make sense to humans. The weights should also provide a faithful explanation for the model output in order to be considered reliable. Through performing multi-weight tests, they show that although there is a certain level of correlation between attention weights and the importance of features in the final prediction of the model, these weights in many cases cannot successfully identify the features that heavily impact a model's decision.

Wiegreffe and Pinter (2019) propose additional tests for evaluating the ability of the attention mechanism to provide explainability. They challenged the findings reported by Jain and Wallace (2019) as they treated the attention as a stand-alone component within a network that is independent from the rest of the components. Through an end-to-end adversarial setup to train models to similar outputs while coming up with different attention distributions in binary classification tasks, they show that the explanations provided by attention are not as unfaithful as Jain and Wallace (2019) found them to be.

In this paper, we extend the adversarial setup by Wiegreffe and Pinter (2019) so that it can be used in a one-shot pass, i.e. training the adversarial models only once. This approach results in a metric, which we call *AtteFa*, that can provide us with a quantitative insight on how faithful the explanations by the attention component are, given a specific model and a specific dataset. To the best of our knowledge, this is the first work that provides such a quantitative measure to evaluate the faithfulness of attention.

## 3 Method

### 3.1 Base Model Training

First, we train a base model on the data. The base model is comprised of an embedding layer, followed by an encoder (LSTM or MLP), which is in turn followed by an attention component, and finally a classification head. To train the base model, cross-entropy loss is used, and training is done for 8 epochs. The final base model is the trained model at the end of the epoch where the ROC-AUC score on the test dataset is minimum.

### 3.2 Adversarial Model Training

With the base model at hand, we train an adversarial model with the same architecture as the base model, but with the following two characteristics:

1. Having predictions as similar as possible to the base model, and
2. Having attention weight distributions as different as possible from the base model

In order to measure the difference between the two models' predictions, namely $\hat{y}_a$ and $\hat{y}_b$, we use Total Variation Distance (TVD), which is computed using Equation 1:

$$\text{TVD}(\hat{y}_a^j, \hat{y}_b^j) = \frac{1}{2}\sum_{j=1}^{|\mathcal{Y}|} |\hat{y}_a^j - \hat{y}_b^j| \quad (1)$$

where $|\mathcal{Y}|$ represents the number of output heads (which is equal to 1 in our binary classification setting).

To compute the difference between attention distributions $\alpha_a$ and $\alpha_b$, Jensen-Shannon Divergence (JSD) is used, which is computed using Equation 2:

$$\text{JSD}(\alpha_a, \alpha_b) = \frac{1}{2}\text{KL}(\alpha_a||\bar{\alpha}) + \frac{1}{2}\text{KL}(\alpha_b||\bar{\alpha}) \quad (2)$$

where $\bar{\alpha} = \frac{\alpha_a+\alpha_b}{2}$ and the Kullback–Leibler (KL) divergence is computed using Equation 3:

$$\text{KL}(\alpha_a||\alpha_b) = \sum_{k=1}^{|\alpha|} \alpha_a^k \times \Big(\log(\alpha_a^k + \epsilon) - \log(\alpha_b^k + \epsilon)\Big) \quad (3)$$

where $|\alpha|$ corresponds to the size of the attention weight vector. The inclusion of $\epsilon$ in the KL equation is to prevent the logs from becoming infinite in cases where the values of $\alpha$ become equal to zero due to mathematical underflow. In our experiments, we set the value of $\epsilon$ equal to 1e-10.

Having the TVD of the predictions and the JSD of the attention weight distributions, we design the loss function so that it tries to minimize TVD and maximize JSD. The final loss formula is given in Equation 4:

$$\mathcal{L}(\mathcal{M}_a, \mathcal{M}_b)^{(i)} = \text{sTVD}(\hat{y}_a^{(i)}, \hat{y}_b^{(i)}) - \text{sJSD}(\alpha_a^{(i)}, \alpha_b^{(i)}) \quad (4)$$

In Equation 4, we use sTVD and sJSD to denote the scaled values of TVD and JSD, respectively. We apply the scaling in order to make sure that the value ranges for the TVD and JSD components of the loss are equal, and therefore the final value of the loss is affected equally by the two components. Knowing that the value of TVD is always between 0 and 0.5, sTVD is computed using Equation 5:

$$\text{sTVD}(\hat{y}_a, \hat{y}_b) = \text{TVD}(\hat{y}_a, \hat{y}_b)/0.5 \quad (5)$$

To compute sJSD Equation 6 is used:

$$\text{sJSD}(\alpha_a, \alpha_b) = \text{JSD}(\alpha_a, \alpha_b)/JSD_{max} \quad (6)$$

where $JSD_{max}$ is the calculated upper-bound for JSD when Equations 2 and 3 are used. $JSD_{max}$ is approximately equal to 0.6931, and is reached when $\alpha_1$ and $\alpha_2$ in Equation 2 are two one-hot vectors with the element 1 located in different indices.

The value of the loss is computed per sample. In order to compute the backpropagated loss value for each batch, we compute the average over the per-sample losses in the batch.

The training process is continued until the loss value on the test data does not improve for 10 consecutive epochs, or a maximum number of 80 epochs is reached. To calculate the total loss on the test data, instead of computing the per-sample losses and averaging them over the dataset, for simplicity and to leverage the metric implementations by Wiegreffe and Pinter (2019), we first average over the per-sample TVD and JSD in this dataset, and then compute the total loss using these averages. As the final adversarial model, we pick the one from the training epoch with the lowest value of loss on test data.

The key difference between our adversarial training setup with the one from Wiegreffe and Pinter (2019) is in the way the adversarial loss is computed. In Wiegreffe and Pinter (2019), KL divergence is used instead of JSD to compute the distribution divergence between the base model's attentions and the adversarial one. Since the value of KL is un-bounded, it is mandatory to use an additional hyperparameter $\lambda$ to avoid the final value of the loss getting dragged fully towards the attention divergence. Knowing that JSD has a specific lower and upper bound, including that in the adversarial loss formula allows us to do away with the additional hyperparameter $\lambda$, and to be able to do the adversarial training in one shot, which in turn provides us with an easy and systematic way to compute a metric value for the attention faithfulness.

### 3.3 Computing *AtteFa*

Having the TVD of the predictions and the JSD of the attention distributions on the test data between the base model $\mathcal{M}_b$ and adversarial version of the model $\mathcal{M}_a$, we compute the faithfulness score *AtteFa* of the attention module $\mathcal{A}_\mathcal{M}$ using Equation 7:

$$\text{AtteFa}(\mathcal{A}_\mathcal{M}) = \min\left(\frac{\text{sTVD}(\hat{y}_a, \hat{y}_b)}{\text{sJSD}(\alpha_a, \alpha_b)}, 1\right) \quad (7)$$

The formula is motivated by the assumption that, the degree of attention faithfulness has a direct relation with the value of the TVD of predictions, and an inverse relation with the value of the JSD of the attention weights. In other words, if the attention is faithful, meaning that the attention can find a limited set of informative sources, the adversarial setup will either converge to a point where both the TVD of predictions and the JSD of attention weights are low, or both of them are high. We believe that the second scenario is more probable, as the adversarial model has a much higher degree of freedom in order to converge to a different attention distribution from the base model than to achieve a similar output prediction. This will later be shown in Section 6 that, with the current adversarial setup, the adversarial model usually achieves a JSD close to its maximum value.

With this assumption, we believe that in most cases, the final value for $\text{sTVD}(\hat{y}_a, \hat{y}_b)$ should be lower than $\text{sJSD}(\alpha_a, \alpha_b)$, but we still do not rule out the opposite scenario, which is why we force the value of AtteFa to be bounded between 0 and 1 through the use of the $\min$ function in Equation 7.

## 4 Datasets

### 4.1 Synthetic Datasets

In principle, we hypothesize that the faithfulness of the attention has a direct relation with the rareness of the informative elements in the input. In the task of text classification, considering the input elements being textual tokens and with an attention that assigns weight to each token, if there are very few informative tokens that could help with the task, our assumption is that the attention should probably focus on those and not the other tokens, and finding alternative attention weight distributions that would lead to a similar outcome would be difficult. Whereas in cases when many input tokens are informative and helpful to the task, the attention can simply shift its focus from one set of tokens to another, therefore the faithfulness will be low.

In order to verify this scenario, we designed a set of synthetic sentiment analysis datasets that include different proportions of informative texts. To that end, we synthetically created samples in a way that a specific portion of their tokens are words with sentiment weights that align with the sentiment label of the sample[1], while filling the rest of the token slots with the uninformative token "something". This results in a simple-to-classify sentiment dataset that allows us to investigate the effect of the frequency of informative input elements on the faithfulness of attention, without the need to take into account the effectiveness of attention for the task at hand.

Our *Mock* datasets are comprised of 8000 training and 1000 testing samples. The distribution of the positive/negative labels is 50/50 in the datasets, and each sample has a random length between 50 and 100 tokens. These synthetic datasets are comprised of **Mock-1**, **Mock-2**, **Mock-5**, and **Mock-10** datasets with 1, 2, 5, and 10 informative tokens in each sample, respectively, and **Mock-1q**, **Mock-2q**, **Mock-3q**, and **Mock-4q**, in which 25%, 50%, 75%, and 100% of the tokens in each sample are informative.

### 4.2 Real-life Datasets

The datasets used are the ones utilized in the work of Jain and Wallace (2019) and Wiegreffe and Pinter (2019). The description of the datasets are provided in section 3 of Jain and Wallace (2019).

### 4.3 Dataset Statistics

Table 1 shows the average number of tokens across samples, along with the distribution of the positive/negative samples for each dataset. Since all the synthetic datasets include the same number of samples, class distributions, and average number of tokens across samples, we have included the statistics for them under *Mock-**.

| Dataset | Train | | Test | |
|---|---|---|---|---|
| | Size (neg/pos) | Avg Len (Tokens) | Size (neg/pos) | Avg Len (Tokens) |
| Mock-* | 4000/4000 | 75 | 500/500 | 75 |
| Diabetes | 6650/1416 | 1985 | 1389/340 | 2385 |
| Anemia | 1742/2912 | 2368 | 512/857 | 2396 |
| IMDB | 8673/8539 | 180 | 2189/2174 | 176 |
| SST | 3310/3610 | 17 | 912/909 | 17 |
| AgNews | 25508/25492 | 36 | 1900/1900 | 36 |
| 20News | 612/624 | 159 | 192/195 | 206 |

Table 1: Summary statistics of the datasets.

## 5 Experimental Setup

The *LSTM* models are comprised of the following components:

1. A 300d word embedding layer
2. A bidirectional LSTM layer (Hochreiter and Schmidhuber, 1997) with 128 units
3. The attention module
4. A fully-connected layer

The *MLP* models include embedding, attention, and fully-connected modules similar to the *LSTM* models, but utilize a feed-forward projection layer with 128 nodes followed by a *tanh* activation, instead of the bi-LSTM layer.

The attention has a two layer fully-connected network that first projects the input to half its size in its first layer, applies a *tanh* activation, and then maps it to a single logit in the second layer. A softmax function is then used to convert the logit to a probability distribution, which is used to compute a weighted average over the inputs and form the output of the attention.

Similar to Jain and Wallace (2019) and Wiegreffe and Pinter (2019), for the Diabetes and Anemia datasets, 300d Word2Vec embeddings (Mikolov et al., 2013) are pre-trained on the combined text from the two datasets. The training is done using CBOW with a window size of 10. For the rest of the datasets, 300d publicly-pretrained FastText embeddings (Bojanowski et al., 2017) are used.

Adam (Kingma and Ba, 2015) is used as the optimizer during training, and the learning rate and

[1]We picked words with positive and negative sentiment from the following gazetteers, respectively: https://ptrckprry.com/course/ssd/data/positive-words.txt, https://ptrckprry.com/course/ssd/data/negative-words.txt

weight decay rates are set to 1e-3 and 1e-5, respectively. Weight decay is applied to every component in the network except the attention module.

## 6 Results and Discussion

First, we have included the F1 scores achieved by the base models in Table 2. In order to verify the correctness of our experiments, we have also included in the table the F1 scores reported by Wiegreffe and Pinter (2019).

| Dataset | LSTM | | MLP | |
|---|---|---|---|---|
| | Reported | Reproduced | Reported | Reproduced |
| Mock-1 | - | 0.974 | - | 0.975 |
| Mock-2 | - | 0.988 | - | 0.989 |
| Mock-5 | - | 0.999 | - | 1.000 |
| Mock-10 | - | 1.000 | - | 0.999 |
| Mock-1q | - | 1.000 | - | 1.000 |
| Mock-2q | - | 1.000 | - | 1.000 |
| Mock-3q | - | 1.000 | - | 1.000 |
| Mock-4q | - | 1.000 | - | 1.000 |
| Diabetes | 0.775 | 0.733 | 0.699 | 0.665 |
| Anemia | 0.938 | 0.935 | 0.920 | 0.915 |
| IMDB | 0.902 | 0.908 | 0.888 | 0.882 |
| SST | 0.831 | 0.830 | 0.817 | 0.816 |
| AgNews | 0.964 | 0.959 | - | 0.956 |
| 20News | 0.942 | 0.935 | - | 0.878 |

Table 2: F1 scores by the base model achieved on the test datasets. The F1 scores reported by Wiegreffe and Pinter (2019) have been included under the *Reported* columns. The MLP setup is equivalent to the *Trained MLP* setup from Wiegreffe and Pinter (2019).

Table 3 contains the results achieved by the adversarial setup. It includes the F1 scores of the adversarial models, the TVD of their predictions from the base models, the JSD of their attention distributions from the base models, the number of epochs that resulted in the best loss on test, and their attention faithfulness score *AtteFa*. The numbers are reported in terms of average and standard deviation runs with 9 different random seeds. Individual results for each seed is available in Tables 4 and 5 in Appendix A.

### 6.1 Effect of Contextualization

Comparing the *AtteFa* columns for the *LSTM* and *MLP* models in Table 3, we can observe that the attentions incorporated in models with LSTM as their encoder are significantly less faithful than their counterparts in the models with MLP as their encoder. This observation was not surprising, as a lower degree of contextualization in token representations should inherently result in higher faithfulness in the attention that is applied on top of those representations.

To better understand this, imagine the task of detecting whether a text is about sports or fruits. Now imagine that you want to classify the following sample: `football is life`. We can simply agree that the only informative word in the sample is `football`, as it clearly indicates a sport. In an ideal scenario, a faithful attention should have a distribution highly centered on this word. Using an MLP encoder, the input tokens will retain their information, therefore the representation of token `football` retains its informativeness. This is, however, not necessarily the case if a contextual encoder such as LSTM is used to compute the token representations, as it can simply manipulate the tokens in a way that another word, such as `is`, has the informative representation.

Going back to our adversarial setting, when LSTM is used, the encoder has the capacity to manipulate the token representations so that a different set of tokens bear the useful information to achieve the task. In this setting, the attention can simply focus on the new set and obtain similar information. On the other hand, a non-contextual MLP encoder does not have the capacity that LSTM holds, and will retain the informativeness of the representation for each token. Therefore, it becomes more challenging for the attention to find a new set of tokens to attend to. That is why the prediction TVD in the MLP models is significantly lower than the LSTM ones, resulting in the MLP models having a noticeably higher AtteFa.

Simply put, our results show that attention components applied on top of contextual encoders are generally less faithful than the ones on top of non-contextual encoders.

### 6.2 Effect of the Frequency of Informative Sources

Looking at the rows corresponding to the results on the *Mock-** datasets and the MLP model in Table 3, we can observe the general trend towards the reduction of AtteFa as the number of informative tokens increase. For the case of the MLP model, a relatively high AtteFa of 0.82 is achieved on the Mock-1 dataset, which only includes one informative token in each sample text, and the value drops to close to 0 for the case of Mock-3q and Mock-4q datasets. This shows that the faithfulness of the attention mechanism has an inverse correlation with the number of informative sources in the input.

The trend is still observable in the case of the

| dataset | LSTM | | | | | MLP | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | epoch | F1 | TVD | JSD | AtteFa | epoch | F1 | TVD | JSD | AtteFa |
| Mock-1 | 12±5 | 0.947±0.020 | 0.015±0.009 | 0.693±0.000 | 0.0304±0.0176 | 20±24 | 0.221±0.312 | 0.231±0.012 | 0.393±0.000 | 0.8153±0.0425 |
| Mock-2 | 9±7 | 0.977±0.010 | 0.016±0.005 | 0.693±0.000 | 0.0329±0.0096 | 1±0 | 0.221±0.312 | 0.246±0.003 | 0.670±0.000 | 0.5086±0.0064 |
| Mock-5 | 7±5 | 1.000±0.001 | 0.002±0.000 | 0.693±0.000 | 0.0037±0.0008 | 9±11 | 0.147±0.275 | 0.247±0.001 | 0.686±0.000 | 0.4987±0.0027 |
| Mock-10 | 14±8 | 1.000±0.000 | 0.001±0.000 | 0.693±0.000 | 0.0012±0.0000 | 23±31 | 0.147±0.275 | 0.249±0.001 | 0.686±0.000 | 0.5028±0.0027 |
| Mock-1q | 23±12 | 1.000±0.000 | 0.000±0.000 | 0.693±0.000 | 0.0006±0.0000 | 37±31 | 0.465±0.480 | 0.135±0.120 | 0.689±0.001 | 0.2721±0.2407 |
| Mock-2q | 35±32 | 1.000±0.001 | 0.000±0.000 | 0.678±0.004 | 0.0008±0.0010 | 42±35 | 0.751±0.324 | 0.099±0.109 | 0.691±0.000 | 0.1983±0.2188 |
| Mock-3q | 21±21 | 1.000±0.000 | 0.000±0.000 | 0.681±0.008 | 0.0003±0.0001 | 13±13 | 0.999±0.001 | 0.001±0.001 | 0.691±0.000 | 0.0013±0.0011 |
| Mock-4q | 8±4 | 1.000±0.000 | 0.000±0.000 | 0.680±0.004 | 0.0002±0.0003 | 3±0 | 1.000±0.000 | 0.000±0.000 | 0.690±0.000 | 0.0002±0.0000 |
| Diabetes | 22±5 | 0.729±0.003 | 0.018±0.001 | 0.693±0.000 | 0.0367±0.0020 | 42±27 | 0.134±0.076 | 0.147±0.004 | 0.691±0.000 | 0.2945±0.0072 |
| Anemia | 20±6 | 0.901±0.018 | 0.058±0.011 | 0.693±0.000 | 0.1164±0.0211 | 23±10 | 0.832±0.007 | 0.093±0.004 | 0.692±0.000 | 0.1861±0.0083 |
| SST | 21±6 | 0.823±0.002 | 0.034±0.002 | 0.626±0.006 | 0.0760±0.0034 | 23±15 | 0.605±0.028 | 0.173±0.001 | 0.656±0.002 | 0.3645±0.0024 |
| IMDB | 49±12 | 0.889±0.006 | 0.038±0.004 | 0.691±0.001 | 0.0769±0.0090 | 21±14 | 0.158±0.056 | 0.190±0.001 | 0.689±0.000 | 0.3826±0.0019 |
| AgNews | 49±18 | 0.958±0.001 | 0.007±0.001 | 0.683±0.002 | 0.0136±0.0015 | 24±12 | 0.610±0.032 | 0.172±0.005 | 0.671±0.001 | 0.3558±0.0097 |
| 20News | 18±5 | 0.865±0.013 | 0.046±0.007 | 0.689±0.001 | 0.0931±0.0149 | 24±18 | 0.340±0.149 | 0.208±0.004 | 0.650±0.008 | 0.4444±0.0133 |

Table 3: Average and standard deviation of the results from our adversarial setup. The results for every row are reported from 9 different runs with different random seed initializations. The column *epoch* includes the the number of training epoch for each selected model.

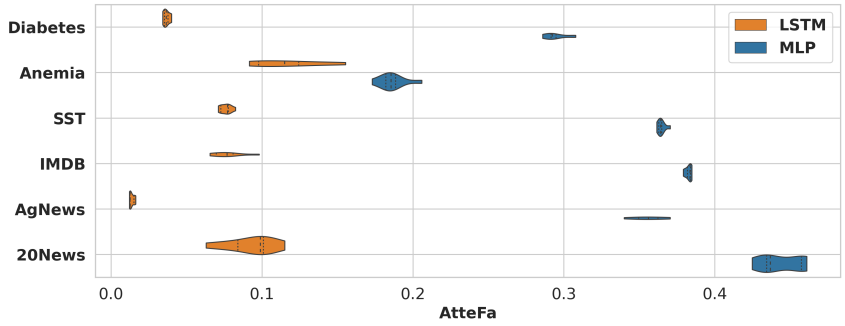


Figure 1: Distribution of AtteFa across different models and real-life datasets.

LSTM models, but with a magnitude that is considerably lower than what we have for the MLP models, as the AtteFa on the Mock-1 dataset is only 0.03. As discussed in Section 6.1, the contextualized LSTM encoder has the flexibility to re-distribute the task-relative information across different input tokens. Regardless of that, we can still observe the general trend towards the drop of AtteFa as we move from Mock-1 to Mock-4, which shows that, even with the case of contextualization, the frequency of informative elements in the source input can still affect the faithfulness of the attention mechanism.

We can observe anomalies in the trend mentioned before. For example, we can observe bumps in the AtteFa in Mock-1 to Mock-2 and Mock-1q to Mock-2q for the case of the LSTM model, and from Mock-5 to Mock-10 in the case of the MLP model. This can be partially justified by the behavior of the base model in terms of how successful it is in detecting informative tokens. An example of this can be found in Table 2, where the MLP model has achieved a lower F1 score on Mock-5 in comparison to Mock-10, meaning that the attention used in the MLP model was more successful in identifying informative tokens in the Mock-5 dataset than in Mock-10.

We can also observe a 19% gap between the AtteFa of the MLP model trained on the Mock-1 dataset and the maximum value of AtteFa (i.e. 1). We argue that this is also related (at least partially) to how the base model performs. We can see in Table 2 that the base model trained on the Mock-1 dataset does not have an F1 score of 1 on the test dataset. This could partially be due to the failure of attention to detect the informative tokens and highly focus on them.

Overall, we conclude that there is generally an inverse relation between the frequency of informative sources in the input data and the faithfulness of the attention module trained on it. But there is still some noise in the AtteFa metric which is attributed to how well the base model performs. Although we do not think that this rules out AtteFa as a suitable metric to compute the faithfulness of attention, we believe there is room for exploring alternative metrics that, for example, also incorporate the performance of the base models in their

computation.

### 6.3 AtteFa on Real-life Datasets

Looking at Table 3, we can see that, for the case of the MLP models, the values of AtteFa on all the real-life datasets are significantly lower than the ones on Mock-1 to Mock-10. As discussed in Section 6.2, this could show that there is quite a large number of informative tokens in the samples belonging to these datasets, which allows the attention to shift its focus among them. This shows that, the attention mechanism in MLP models trained on all these datasets is not very faithful.

For the case of the LSTM model, however, we can observe that the AtteFa on these real-life datasets is comparable and sometimes higher than their counterparts on the Mock-* datasets. However, focusing only on the real-life datasets, the AtteFa of the LSTM models are still lower than the MLP ones. This can also be visually observed in Figure 1, which includes the violin plots of the distribution of AtteFa across the different datasets and models. We hypothesize that, in real-life datasets, we have a significantly lower number of completely uninformative tokens as we had in the Mock-* datasets. Although the LSTM encoder still retains its flexibility to redistribute information across different tokens, the lower number of completely uninformative tokens reduces the degree of the information redistribution capacity. This is something that we have not explored in our experiments with the synthetic datasets, and therefore, leaves room for more studies on this aspect.

One may argue that the number of input tokens on its own can affect the distribution of attention weights and can in turn affect the value of the attention JSD of the adversarial models, hence the final value of AtteFa. While we do not rule this out, we believe that it is not merely the input lengths that would affect the attention JSD, but rather the frequency of informative input tokens that could increase as the input lengths become higher. We also believe that the way information is distributed among their representations used by the attention component also plays a big role here.

In Figure 1, we can see that for the case of the MLP models, the values of AtteFa on datasets with lengthier samples, namely Diabetes and Anemia, are generally lower than the ones on the other datasets. This is, however, not the case for the LSTM models, as we can observe a relatively high AtteFa on the Anemia dataset with respect to the rest of the datasets. Even for the case of the MLP model, we can see that the AtteFa on the 20News dataset is higher than SST and AgNews that have lower average input lengths (see Table 1).

We therefore conclude that the distribution of task-related information across the input token representations used by the attention component plays a key role in the faithfulness of the attention.

### 6.4 Comparison of Our Adversial Setup with Wiegreffe and Pinter's

In Figure 2, we have plotted the prediction TVD and attention JSD of our adversarial LSTM models against the results reported in Wiegreffe and Pinter (2019). The dotted lines in the plots resemble the ones in figure 5 from Wiegreffe and Pinter (2019).

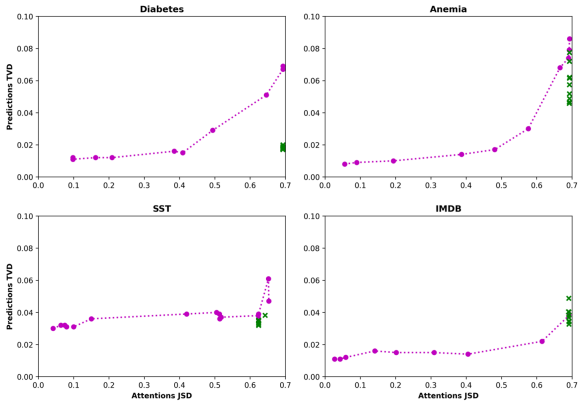


Figure 2: Visual comparison of averaged per-instance test set JSD and TVD from base model for each model variant between our adversarial setup and the one from Wiegreffe and Pinter (2019). The • show results from Wiegreffe and Pinter (2019), and the × show results from our setup.

We can see that, with our adversarial setup, we have achieved comparable prediction TVDs to Wiegreffe and Pinter's on the Anemia, SST and IMDB datasets. However, on the Diabetes dataset, our prediction TVDs are significantly lower than Wiegreffe and Pinter's. Given that our adversarial setups are pretty similar, we believe that this is mainly due to our inability to properly reproduce their base LSTM model on the Diabetes dataset. We can observe this from the 0.042 drop in the F1 score of our model from what was reported in Wiegreffe and Pinter (2019).

Looking at Figure 2, we can see that the adversarial results that we have achieved are towards the higher-end of the attention JSDs reported by Wiegreffe and Pinter (2019). This is very close to the calculated upper-bound for JSD, which is

0.6931. Wiegreffe and Pinter used the hyperparameter $\lambda$ in order to reduce the effect of the attention JSD in the value of their loss. With the removal of this hyperparameter in our setup (which is the equivalent of setting it to 1), the adversarial training leads the model to primarily maximize the attention JSD, as it is an easier objective than to minimize the prediction TVD. Therefore, we usually end up with an almost maxed-out attention JSD, and it is mainly the prediction TVD that determines the value of AtteFa. However, we argue that the JSD is not always fully maxed-out (see the plot for the SST dataset in Figure 2), and therefore, we cannot simply disregard it in the computation of AtteFa.

## 7 Limitations

There are certain limitations with the current work, in terms of both the methodology used to compute AtteFa, and the different factors affecting the attention faithfulness. In this section, we explore the ones that we believe are the most important:

The current methodology to compute AtteFa is scoped solely on binary text classification. In order to have AtteFa as a widely accepted metric in the NLP community, the methodology needs to be extended to other NLP tasks, such as multi-class classification, text retrieval, question answering, machine translation, etc.

In the current work, we have studied the effect of the frequency of informative tokens on the faithfulness of attention through running experiments on the Mock-* datasets, which are synthetic datasets for sentiment classification. The current selection of sentiment words and their positioning within the input texts were done in a random fashion. A more thorough experiment would explore the effect of the distribution of informative tokens across the input texts (centered towards the start/end/middle vs. scattered evenly), along with a more careful selection of the words to be used as the informative tokens (e.g. differentiating between words with strong vs. weak sentiments).

In terms of investigating the effect of encoder contextualization on the faithfulness of attention, we have explored using token-level MLP as a non-contextual encoder and LSTM as a contextual one. This can be extended to exploring other encoder architectures, such as CNNs (LeCun et al., 1999), GRUs (Cho et al., 2014), and transformers (Vaswani et al., 2017).

Another aspect in the current study which has room for exploration is the evaluation of the effect of softmax temperature on the faithfulness of attention. We believe that higher faithfulness may be achieved by using lower temperatures in the case of datasets with infrequent informative tokens, and higher temperature in the case of datasets with frequent informative tokens within their input.

Last, but not least, the experiments in this work are only focused on a specific type of single-head attention. We believe that the current approach does not transfer properly to multi-headed attentions, as we may still consider a multi-headed attention faithful if the only way for the adversarial model to come up with the same predictions as the base model is to change the order of the attention heads and not the attention weights computed by them. Due to the frequent use of multi-headed attentions in state-of-the-art NLP models, the extension of AtteFa to multi-headed attentions would play a big role in its widespread adoption by the NLP community.

## 8 Conclusion

In this paper, we presented an adversarial training approach for binary text classification tasks, which can provide us with the metric *AtteFa* that quantitatively measures the degree of faithfulness in the attention weights. We, then, measured the effect of contextualization, as well as the effect of the frequency of informative tokens on the attention faithfulness. Finally, we computed and evaluated AtteFa for models trained on several real-life binary text classification datasets.

We hope that the presented approach can act as a motivation for researchers to further explore automatic approaches to quantitatively measure the degree of model explainability or its different aspects (e.g. faithfulness, plausibility, sufficiency, etc.).

As future directions, we plan to address the limitations specified in Section 7 to come up with a more reliable and more widely applicable metric to measure the faithfulness of attention. We also plan to measure attention faithfulness in other settings, e.g. the use of different types of attention such as multi-headed and scaled dot-product (Vaswani et al., 2017), the use of attention components in different layers of a model, etc.

## Acknowledgments

We would like to express our gratitude to Sarah Wiegreffe, Yuval Pinter, Sarthak Jain, and Byron Wallace for the availability of their high quality code, which greatly helped us with the current work. We also thank the anonymous reviewers for their comments on an earlier version of this paper.

This work was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

## References

Malika Aubakirova and Mohit Bansal. 2016. Interpreting neural networks to improve politeness comprehension. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2035–2041, Austin, Texas, USA.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations, (ICLR 2015)*, San Diego, California, USA.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1724–1734, Doha, Qatar.

Reza Ghaeini, Xiaoli Fern, and Prasad Tadepalli. 2018. Interpreting recurrent and attention-based neural models: a case study on natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 4952–4957, Brussels, Belgium.

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

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 3543–3556, Minneapolis, Minnesota, USA.

Sweta Karlekar, Tong Niu, and Mohit Bansal. 2018. Detecting linguistic characteristics of Alzheimer's dementia by interpreting neural models. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 701–707, New Orleans, Louisiana.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR 2015)*, San Diego, California, USA.

Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. 1999. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, pages 319–345.

Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. 2017. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017): System Demonstrations*, pages 121–126, Copenhagen, Denmark.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Workshop Proceedings of the International Conference on Learning Representations (ICLR 2013)*, Scottsdale, Arizona, USA.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 1101–1111, New Orleans, Louisiana.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016): Demonstrations*, pages 97–101, San Diego, California, USA.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization.

In *Proceedings of the 2017 IEEE international conference on computer vision (ICCV 2017)*, pages 618–626, Venice, Italy.

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

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 2017 International Conference on Machine Learning (ICML 2017)*, pages 3319–3328, Sydney, Australia.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Proceedings of the 2017 Conference on Advances in Neural Information Processing Systems (NIPS 2017)*, pages 5998–6008. Long Beach, California, USA.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 606–615, Austin, Texas, USA.

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 2019)*, pages 11–20, Hong Kong, China.

Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard Hovy. 2017. An interpretable knowledge transfer model for knowledge base completion. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 950–962, Vancouver, Canada.

Luisa M Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. 2017. Visualizing deep neural network decisions: Prediction difference analysis. In *Proceedings of the 2017 International Conference on Learning Representations (ICLR 2017)*.

## A All Results on the Adversarial Setup

Tables 4 and 5 are the extended versions of Table 3, which include the results from the adversarial setup for each individual random seed that was used in the training of the adversarial models.

| dataset | seed | LSTM | | | | | MLP | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | epoch | F1 | TVD | JSD | AtteFa | epoch | F1 | TVD | JSD | AtteFa |
| Mock-1 | 10 | 7 | 0.923 | 0.025 | 0.693 | 0.050 | 47 | 0.662 | 0.222 | 0.393 | 0.784 |
| | 50 | 16 | 0.971 | 0.004 | 0.693 | 0.008 | 1 | 0.000 | 0.244 | 0.393 | 0.860 |
| | 257 | 15 | 0.943 | 0.017 | 0.693 | 0.033 | 1 | 0.000 | 0.219 | 0.393 | 0.774 |
| | 500231 | 20 | 0.960 | 0.011 | 0.693 | 0.021 | 76 | 0.662 | 0.223 | 0.393 | 0.787 |
| | 100078 | 13 | 0.968 | 0.006 | 0.693 | 0.011 | 9 | 0.000 | 0.247 | 0.393 | 0.872 |
| | 12504 | 7 | 0.916 | 0.029 | 0.693 | 0.057 | 11 | 0.000 | 0.249 | 0.393 | 0.879 |
| | 90754789 | 16 | 0.963 | 0.008 | 0.693 | 0.015 | 7 | 0.000 | 0.218 | 0.393 | 0.767 |
| | 8988812 | 3 | 0.926 | 0.026 | 0.693 | 0.052 | 30 | 0.662 | 0.223 | 0.393 | 0.785 |
| | 2 | 9 | 0.952 | 0.012 | 0.693 | 0.025 | 2 | 0.000 | 0.235 | 0.393 | 0.828 |
| Mock-2 | 10 | 2 | 0.982 | 0.015 | 0.692 | 0.030 | 1 | 0.662 | 0.243 | 0.670 | 0.504 |
| | 50 | 4 | 0.977 | 0.016 | 0.693 | 0.032 | 1 | 0.000 | 0.248 | 0.670 | 0.513 |
| | 257 | 12 | 0.978 | 0.016 | 0.693 | 0.032 | 1 | 0.000 | 0.244 | 0.670 | 0.506 |
| | 500231 | 5 | 0.964 | 0.024 | 0.693 | 0.047 | 2 | 0.662 | 0.242 | 0.670 | 0.502 |
| | 100078 | 15 | 0.975 | 0.016 | 0.693 | 0.033 | 1 | 0.000 | 0.247 | 0.670 | 0.510 |
| | 12504 | 19 | 0.960 | 0.024 | 0.693 | 0.048 | 1 | 0.000 | 0.250 | 0.670 | 0.519 |
| | 90754789 | 3 | 0.978 | 0.017 | 0.693 | 0.034 | 1 | 0.000 | 0.242 | 0.670 | 0.502 |
| | 8988812 | 4 | 0.984 | 0.012 | 0.693 | 0.024 | 1 | 0.662 | 0.243 | 0.670 | 0.504 |
| | 2 | 19 | 0.997 | 0.007 | 0.693 | 0.015 | 1 | 0.000 | 0.250 | 0.670 | 0.519 |
| Mock-5 | 10 | 12 | 1.000 | 0.002 | 0.693 | 0.003 | 1 | 0.662 | 0.249 | 0.686 | 0.504 |
| | 50 | 3 | 1.000 | 0.002 | 0.693 | 0.003 | 17 | 0.000 | 0.246 | 0.686 | 0.497 |
| | 257 | 2 | 0.998 | 0.003 | 0.693 | 0.006 | 4 | 0.000 | 0.246 | 0.686 | 0.497 |
| | 500231 | 20 | 1.000 | 0.002 | 0.693 | 0.003 | 2 | 0.000 | 0.247 | 0.686 | 0.499 |
| | 100078 | 5 | 0.999 | 0.002 | 0.693 | 0.004 | 16 | 0.000 | 0.246 | 0.686 | 0.497 |
| | 12504 | 7 | 1.000 | 0.002 | 0.693 | 0.003 | 2 | 0.000 | 0.246 | 0.686 | 0.497 |
| | 90754789 | 3 | 1.000 | 0.002 | 0.693 | 0.003 | 34 | 0.000 | 0.246 | 0.686 | 0.497 |
| | 8988812 | 5 | 1.000 | 0.002 | 0.693 | 0.003 | 1 | 0.662 | 0.249 | 0.686 | 0.504 |
| | 2 | 7 | 1.000 | 0.002 | 0.693 | 0.003 | 2 | 0.000 | 0.246 | 0.686 | 0.497 |
| Mock-10 | 10 | 21 | 1.000 | 0.001 | 0.693 | 0.001 | 1 | 0.662 | 0.251 | 0.686 | 0.508 |
| | 50 | 25 | 1.000 | 0.001 | 0.693 | 0.001 | 80 | 0.000 | 0.248 | 0.686 | 0.501 |
| | 257 | 6 | 1.000 | 0.001 | 0.693 | 0.001 | 13 | 0.000 | 0.248 | 0.686 | 0.501 |
| | 500231 | 8 | 1.000 | 0.001 | 0.693 | 0.001 | 2 | 0.000 | 0.249 | 0.686 | 0.503 |
| | 100078 | 16 | 1.000 | 0.001 | 0.693 | 0.001 | 80 | 0.000 | 0.248 | 0.686 | 0.501 |
| | 12504 | 4 | 1.000 | 0.001 | 0.693 | 0.001 | 6 | 0.000 | 0.248 | 0.686 | 0.501 |
| | 90754789 | 9 | 1.000 | 0.001 | 0.693 | 0.001 | 24 | 0.000 | 0.248 | 0.686 | 0.501 |
| | 8988812 | 28 | 1.000 | 0.001 | 0.693 | 0.001 | 1 | 0.662 | 0.251 | 0.686 | 0.508 |
| | 2 | 12 | 1.000 | 0.001 | 0.693 | 0.001 | 4 | 0.000 | 0.248 | 0.686 | 0.501 |
| Mock-1q | 10 | 20 | 1.000 | 0.000 | 0.693 | 0.001 | 70 | 0.000 | 0.247 | 0.690 | 0.497 |
| | 50 | 40 | 1.000 | 0.000 | 0.693 | 0.001 | 10 | 1.000 | 0.001 | 0.688 | 0.002 |
| | 257 | 39 | 1.000 | 0.000 | 0.693 | 0.001 | 79 | 0.093 | 0.235 | 0.690 | 0.473 |
| | 500231 | 13 | 1.000 | 0.000 | 0.693 | 0.001 | 80 | 0.093 | 0.235 | 0.690 | 0.473 |
| | 100078 | 17 | 1.000 | 0.000 | 0.693 | 0.001 | 6 | 1.000 | 0.002 | 0.688 | 0.003 |
| | 12504 | 14 | 1.000 | 0.000 | 0.693 | 0.001 | 6 | 1.000 | 0.002 | 0.689 | 0.003 |
| | 90754789 | 12 | 1.000 | 0.000 | 0.693 | 0.001 | 44 | 0.000 | 0.247 | 0.690 | 0.497 |
| | 8988812 | 15 | 1.000 | 0.000 | 0.693 | 0.001 | 28 | 0.000 | 0.247 | 0.690 | 0.497 |
| | 2 | 39 | 1.000 | 0.000 | 0.693 | 0.001 | 6 | 1.000 | 0.002 | 0.689 | 0.004 |
| Mock-2q | 10 | 25 | 1.000 | 0.000 | 0.678 | 0.001 | 80 | 0.305 | 0.203 | 0.690 | 0.408 |
| | 50 | 80 | 1.000 | 0.000 | 0.681 | 0.000 | 79 | 0.700 | 0.212 | 0.690 | 0.426 |
| | 257 | 79 | 1.000 | 0.000 | 0.681 | 0.000 | 4 | 0.997 | 0.002 | 0.691 | 0.003 |
| | 500231 | 5 | 0.998 | 0.002 | 0.678 | 0.003 | 40 | 0.995 | 0.003 | 0.691 | 0.005 |
| | 100078 | 23 | 0.999 | 0.001 | 0.681 | 0.001 | 80 | 0.089 | 0.236 | 0.690 | 0.474 |
| | 12504 | 8 | 1.000 | 0.000 | 0.682 | 0.000 | 79 | 0.683 | 0.230 | 0.690 | 0.461 |
| | 90754789 | 12 | 1.000 | 0.000 | 0.681 | 0.000 | 4 | 0.998 | 0.001 | 0.691 | 0.002 |
| | 8988812 | 3 | 1.000 | 0.000 | 0.670 | 0.000 | 10 | 0.997 | 0.002 | 0.691 | 0.004 |
| | 2 | 79 | 1.000 | 0.000 | 0.670 | 0.000 | 6 | 0.999 | 0.001 | 0.690 | 0.001 |
| Mock-3q | 10 | 5 | 1.000 | 0.000 | 0.674 | 0.000 | 5 | 0.998 | 0.001 | 0.691 | 0.002 |
| | 50 | 16 | 1.000 | 0.000 | 0.675 | 0.000 | 4 | 0.997 | 0.002 | 0.691 | 0.003 |
| | 257 | 9 | 1.000 | 0.000 | 0.674 | 0.001 | 2 | 1.000 | 0.000 | 0.690 | 0.000 |
| | 500231 | 12 | 1.000 | 0.000 | 0.691 | 0.000 | 44 | 1.000 | 0.000 | 0.691 | 0.000 |
| | 100078 | 10 | 1.000 | 0.000 | 0.691 | 0.000 | 13 | 1.000 | 0.000 | 0.691 | 0.000 |
| | 12504 | 4 | 1.000 | 0.000 | 0.689 | 0.000 | 5 | 0.999 | 0.001 | 0.691 | 0.001 |
| | 90754789 | 46 | 1.000 | 0.000 | 0.675 | 0.000 | 11 | 1.000 | 0.000 | 0.691 | 0.000 |
| | 8988812 | 71 | 1.000 | 0.000 | 0.686 | 0.000 | 25 | 0.999 | 0.001 | 0.691 | 0.001 |
| | 2 | 12 | 1.000 | 0.000 | 0.674 | 0.000 | 5 | 0.998 | 0.001 | 0.691 | 0.002 |
| Mock-4q | 10 | 6 | 1.000 | 0.000 | 0.683 | 0.000 | 3 | 1.000 | 0.000 | 0.690 | 0.000 |
| | 50 | 7 | 1.000 | 0.000 | 0.683 | 0.000 | 3 | 1.000 | 0.000 | 0.690 | 0.000 |
| | 257 | 5 | 1.000 | 0.000 | 0.683 | 0.000 | 3 | 1.000 | 0.000 | 0.690 | 0.000 |
| | 500231 | 12 | 1.000 | 0.000 | 0.683 | 0.000 | 2 | 1.000 | 0.000 | 0.690 | 0.000 |
| | 100078 | 9 | 1.000 | 0.000 | 0.683 | 0.000 | 3 | 1.000 | 0.000 | 0.690 | 0.000 |
| | 12504 | 16 | 1.000 | 0.000 | 0.682 | 0.000 | 3 | 1.000 | 0.000 | 0.691 | 0.000 |
| | 90754789 | 7 | 0.999 | 0.001 | 0.670 | 0.001 | 3 | 1.000 | 0.000 | 0.690 | 0.000 |
| | 8988812 | 8 | 1.000 | 0.000 | 0.674 | 0.000 | 3 | 1.000 | 0.000 | 0.690 | 0.000 |
| | 2 | 4 | 1.000 | 0.000 | 0.683 | 0.000 | 4 | 1.000 | 0.000 | 0.690 | 0.000 |

Table 4: All results from our adversarial setup on the synthetic datasets.

| dataset | seed | LSTM | | | | | MLP | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | epoch | F1 | TVD | JSD | AtteFa | epoch | F1 | TVD | JSD | AtteFa |
| Diabetes | 10 | 21 | 0.732 | 0.017 | 0.693 | 0.035 | 58 | 0.180 | 0.145 | 0.691 | 0.291 |
| | 50 | 25 | 0.730 | 0.018 | 0.693 | 0.037 | 76 | 0.203 | 0.142 | 0.691 | 0.286 |
| | 257 | 23 | 0.730 | 0.018 | 0.693 | 0.037 | 66 | 0.167 | 0.146 | 0.690 | 0.293 |
| | 500231 | 22 | 0.730 | 0.018 | 0.693 | 0.035 | 21 | 0.112 | 0.147 | 0.690 | 0.295 |
| | 100078 | 25 | 0.726 | 0.020 | 0.693 | 0.040 | 34 | 0.227 | 0.143 | 0.691 | 0.287 |
| | 12504 | 20 | 0.735 | 0.019 | 0.693 | 0.038 | 80 | 0.159 | 0.146 | 0.691 | 0.292 |
| | 90754789 | 31 | 0.728 | 0.018 | 0.693 | 0.035 | 24 | 0.151 | 0.145 | 0.691 | 0.291 |
| | 8988812 | 10 | 0.723 | 0.020 | 0.693 | 0.040 | 15 | 0.006 | 0.153 | 0.690 | 0.307 |
| | 2 | 18 | 0.729 | 0.017 | 0.692 | 0.034 | 5 | 0.000 | 0.153 | 0.691 | 0.308 |
| Anemia | 10 | 32 | 0.900 | 0.057 | 0.693 | 0.115 | 30 | 0.833 | 0.094 | 0.692 | 0.189 |
| | 50 | 14 | 0.877 | 0.072 | 0.693 | 0.144 | 13 | 0.842 | 0.086 | 0.692 | 0.173 |
| | 257 | 15 | 0.923 | 0.047 | 0.692 | 0.093 | 45 | 0.841 | 0.095 | 0.693 | 0.190 |
| | 500231 | 19 | 0.914 | 0.049 | 0.693 | 0.098 | 11 | 0.829 | 0.091 | 0.693 | 0.182 |
| | 100078 | 26 | 0.894 | 0.061 | 0.693 | 0.123 | 23 | 0.828 | 0.093 | 0.692 | 0.185 |
| | 12504 | 14 | 0.877 | 0.077 | 0.693 | 0.155 | 29 | 0.817 | 0.103 | 0.693 | 0.206 |
| | 90754789 | 26 | 0.912 | 0.052 | 0.693 | 0.104 | 19 | 0.835 | 0.093 | 0.692 | 0.186 |
| | 8988812 | 19 | 0.888 | 0.062 | 0.693 | 0.124 | 23 | 0.831 | 0.091 | 0.692 | 0.183 |
| | 2 | 17 | 0.927 | 0.046 | 0.692 | 0.092 | 14 | 0.834 | 0.091 | 0.693 | 0.182 |
| SST | 10 | 23 | 0.825 | 0.038 | 0.643 | 0.082 | 23 | 0.657 | 0.172 | 0.658 | 0.362 |
| | 50 | 16 | 0.822 | 0.033 | 0.624 | 0.072 | 10 | 0.598 | 0.172 | 0.656 | 0.363 |
| | 257 | 29 | 0.821 | 0.032 | 0.624 | 0.071 | 28 | 0.577 | 0.172 | 0.655 | 0.364 |
| | 500231 | 11 | 0.823 | 0.033 | 0.624 | 0.073 | 22 | 0.580 | 0.173 | 0.657 | 0.365 |
| | 100078 | 30 | 0.822 | 0.035 | 0.624 | 0.078 | 15 | 0.576 | 0.174 | 0.653 | 0.370 |
| | 12504 | 20 | 0.828 | 0.033 | 0.624 | 0.074 | 61 | 0.590 | 0.173 | 0.660 | 0.365 |
| | 90754789 | 18 | 0.818 | 0.035 | 0.624 | 0.078 | 21 | 0.630 | 0.173 | 0.657 | 0.364 |
| | 8988812 | 24 | 0.823 | 0.035 | 0.624 | 0.077 | 9 | 0.636 | 0.173 | 0.654 | 0.366 |
| | 2 | 18 | 0.823 | 0.035 | 0.624 | 0.078 | 14 | 0.599 | 0.172 | 0.656 | 0.363 |
| IMDB | 10 | 62 | 0.896 | 0.034 | 0.691 | 0.069 | 13 | 0.210 | 0.191 | 0.689 | 0.385 |
| | 50 | 42 | 0.889 | 0.040 | 0.689 | 0.081 | 18 | 0.230 | 0.188 | 0.689 | 0.379 |
| | 257 | 50 | 0.891 | 0.037 | 0.691 | 0.074 | 10 | 0.101 | 0.191 | 0.689 | 0.384 |
| | 500231 | 26 | 0.893 | 0.038 | 0.691 | 0.077 | 45 | 0.073 | 0.190 | 0.689 | 0.382 |
| | 100078 | 44 | 0.889 | 0.033 | 0.691 | 0.066 | 11 | 0.180 | 0.191 | 0.689 | 0.384 |
| | 12504 | 41 | 0.892 | 0.035 | 0.691 | 0.070 | 12 | 0.195 | 0.190 | 0.689 | 0.382 |
| | 90754789 | 60 | 0.890 | 0.040 | 0.691 | 0.081 | 29 | 0.100 | 0.191 | 0.690 | 0.384 |
| | 8988812 | 65 | 0.875 | 0.049 | 0.691 | 0.098 | 45 | 0.215 | 0.189 | 0.689 | 0.380 |
| | 2 | 54 | 0.890 | 0.038 | 0.691 | 0.077 | 10 | 0.120 | 0.191 | 0.689 | 0.385 |
| AgNews | 10 | 27 | 0.959 | 0.006 | 0.680 | 0.013 | 11 | 0.630 | 0.164 | 0.670 | 0.340 |
| | 50 | 28 | 0.958 | 0.006 | 0.681 | 0.013 | 17 | 0.567 | 0.174 | 0.671 | 0.359 |
| | 257 | 33 | 0.957 | 0.008 | 0.685 | 0.016 | 10 | 0.653 | 0.169 | 0.671 | 0.349 |
| | 500231 | 60 | 0.958 | 0.006 | 0.680 | 0.012 | 49 | 0.639 | 0.172 | 0.670 | 0.356 |
| | 100078 | 37 | 0.958 | 0.006 | 0.680 | 0.013 | 17 | 0.610 | 0.167 | 0.671 | 0.345 |
| | 12504 | 78 | 0.957 | 0.008 | 0.685 | 0.016 | 19 | 0.597 | 0.176 | 0.672 | 0.362 |
| | 90754789 | 68 | 0.959 | 0.007 | 0.685 | 0.015 | 27 | 0.633 | 0.171 | 0.671 | 0.352 |
| | 8988812 | 66 | 0.958 | 0.007 | 0.686 | 0.014 | 23 | 0.549 | 0.179 | 0.673 | 0.369 |
| | 2 | 47 | 0.958 | 0.006 | 0.680 | 0.012 | 39 | 0.611 | 0.179 | 0.671 | 0.370 |
| 20News | 10 | 23 | 0.849 | 0.057 | 0.689 | 0.115 | 34 | 0.246 | 0.207 | 0.661 | 0.433 |
| | 50 | 26 | 0.863 | 0.038 | 0.690 | 0.077 | 34 | 0.176 | 0.200 | 0.653 | 0.425 |
| | 257 | 21 | 0.850 | 0.050 | 0.688 | 0.100 | 23 | 0.315 | 0.207 | 0.656 | 0.437 |
| | 500231 | 17 | 0.858 | 0.050 | 0.687 | 0.101 | 57 | 0.397 | 0.206 | 0.659 | 0.434 |
| | 100078 | 16 | 0.894 | 0.031 | 0.689 | 0.063 | 3 | 0.460 | 0.211 | 0.639 | 0.457 |
| | 12504 | 13 | 0.859 | 0.052 | 0.689 | 0.105 | 4 | 0.440 | 0.211 | 0.641 | 0.456 |
| | 90754789 | 16 | 0.874 | 0.049 | 0.688 | 0.099 | 44 | 0.043 | 0.205 | 0.654 | 0.435 |
| | 8988812 | 7 | 0.863 | 0.047 | 0.688 | 0.094 | 11 | 0.495 | 0.214 | 0.644 | 0.461 |
| | 2 | 19 | 0.875 | 0.042 | 0.689 | 0.084 | 10 | 0.492 | 0.213 | 0.642 | 0.461 |

Table 5: All results from our adversarial setup on the real-life datasets.