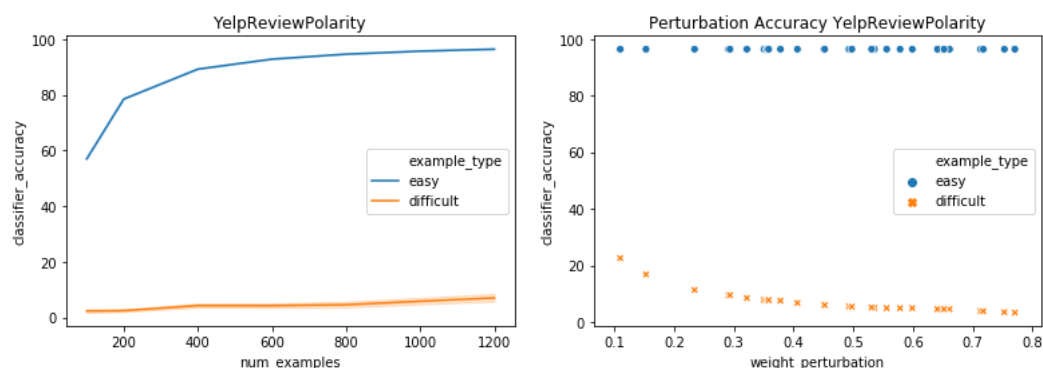


Thank you all for your insightful comments. This actually inspired us to do new sets of experiments to validate our claims. We validated our claims on linguistic resilience by performing additional experiments on programmatic sentence perturbations in both embedding space and by token substitution. In figure 1, we mine difficult examples and present accuracy results of a classifier on the same.

We also do extensive experiments on three other datasets YelpReviewPolarity, AGNEWS, and SogouNews since there is now a convenient way to access them through torchtext's newest release. A wide range of datasets of different sizes with different numbers of output classes should hopefully make for a stronger case for our approach. We hope to finish some more dataset experiments over this week and update the results.

Figure 1: Classifier Performance on difficult examples

Here we take the Joulin et al 2016 based text classification approach (and plot classifier accuracy as a function of the step size of the perturbation. The number of examples in the x-axis represents the number of examples based on the eigenvalue. So, 200 refers to the 200 easy examples and 200 difficult examples. We then perturb these examples in the direction of the eigenvector and check if the classifier prediction flipped. As can be seen that the classifier accuracy remains very high for easy examples and is significantly low for difficult examples. On the second diagram, for easy examples, the classifier still exhibits minimal performance drop when the weight of the perturbation along the eigenvector is increased. Thus our method is dataset agnostic.



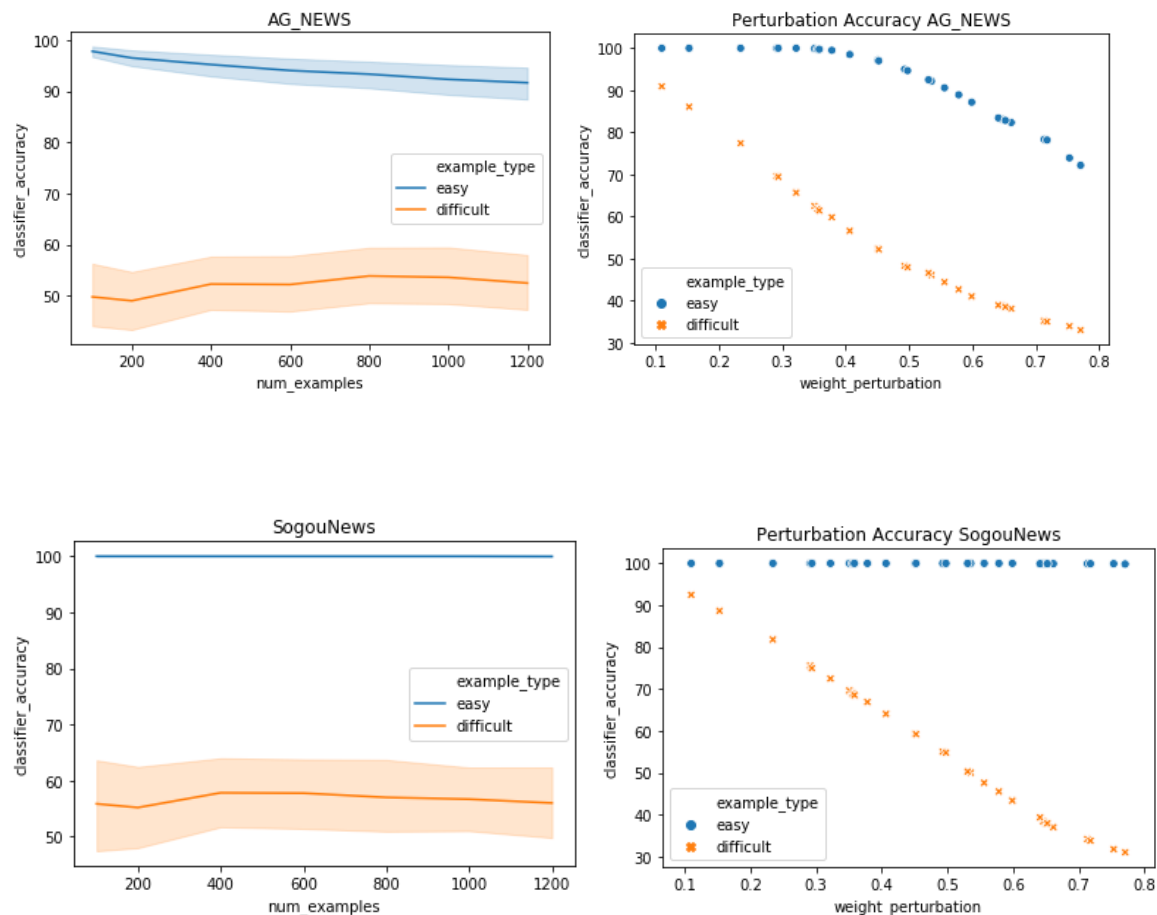
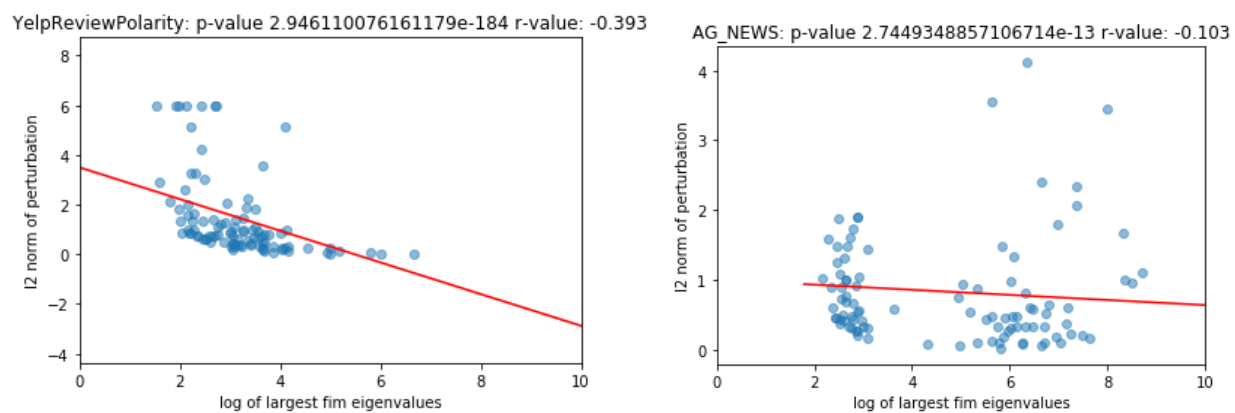
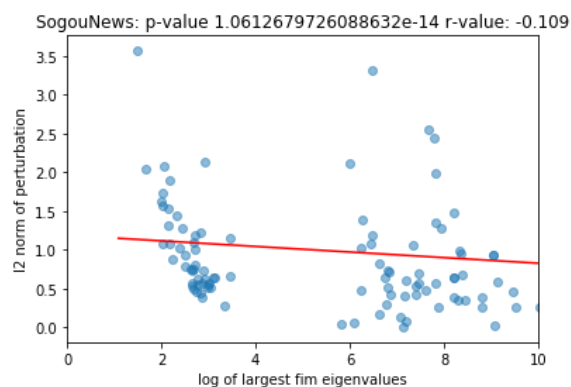


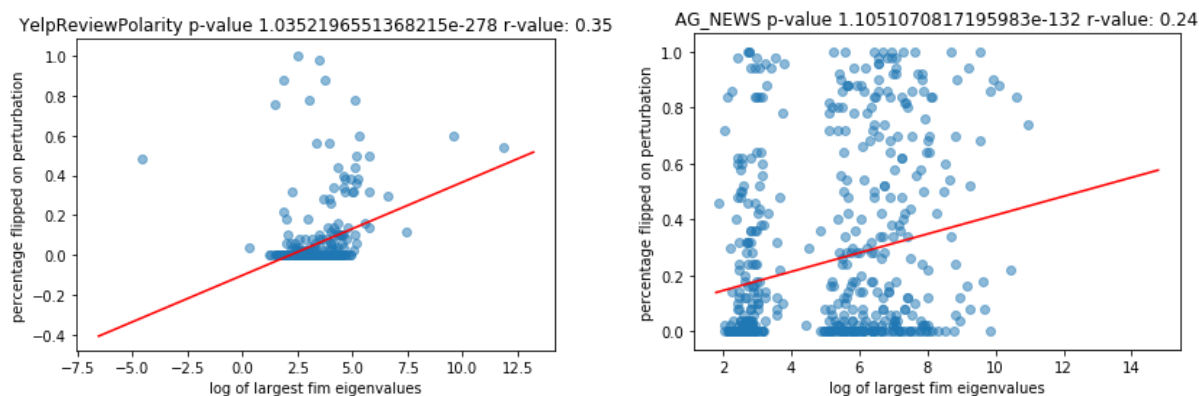
Figure 2: Eigenvalue vs minimum perturbation strengths to flip classifier performance.

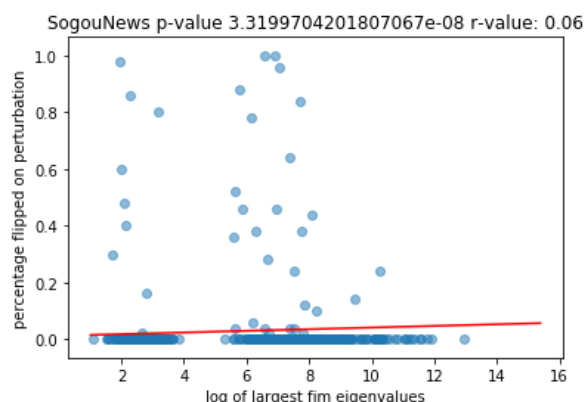




We use the eigenvector with the largest eigenvalue as a perturbation in embedding space. As we can see there is a linear relationship between log lambda max and the perturbation strength needed to flip the classifier output. We use binary search between the range (0,6) to discover the minimal L2 norm perturbation along the largest eigenvector which can successfully flip the classifier's output. For each dataset, we select 500 random examples from the test set for the perturbation. Note the negative slope in all the datasets. The p-value and the r-value are reported at the top. The log of the largest feature eigenvalue is on the x-axis. Thus, an example with a small eigenvalue requires a larger perturbation to flip classifier prediction than an example with small feature eigenvalue.

Figure 3: Random flip success probability vs log feature eigenvalue





We notice the linear relationship between the log of eigenvalue and the percentage of successful word flips. Since the length of each document varies drastically for these document classification problems, we decide the number of words to flip as 10% of the document length. For each sentence, we randomly sample 10% of the words in that sentence and substitute the words from the vocabulary of that dataset. We then measure the percentage of predictions whose classification label changes and the log of fim eigenvalue. The p_value and the r_value are reported at the top.

Implications

As we demonstrated above FIM captures resilience to linguistic perturbations (Figure 1,2 and 3). This makes our approach a simple versatile formulation to interpret the vulnerability of NLP models.

Dataset Name	Model Accuracy	Num of Test Examples	No of classes
AG_NEWS	90.1%	7600	4
SogouNews	95.6%	60000	5
YelpReviewPolarity	93.7%	38000	2

Reference:

1. Joulin, Armand, et al. "Bag of tricks for efficient text classification." *arXiv preprint arXiv:1607.01759* (2016).