Make some noise for no noise! A framework for a BERT robust to noise

Sarah Zhao
M.I.T.
sarahaz@mit.edu

Blisse Kong
M.I.T.
blisse@mit.edu

Ruiyi Wang
M.I.T.
ruiwangk@mit.edu

Abstract

State-of-the-art Natural Language Processing (NLP) models are often trained on large amounts of data; for example, BERT was pre-trained on English Wikipedia. However, noise, such as slang, misspellings, or emojis, exists in many datasets (most notably those involving unprocessed, uncleaned language from the Internet), and BERT has been shown to perform worse on data with weak supervision noise compared to baseline metrics. BERT has appeared to be robust against synthetic noise, but weak supervision noise is crucial to target since parts of the research community may not have access to high-quality labeled training data. Large, clean datasets are sometimes inaccessible to researchers due to financial barriers or lack of manpower to process massive amounts of data, so building models that are robust to noisy datasets can democratize developments in NLP research at large. This project proposes a data augmentation approach to handling noisy datasets for named entity recognition tasks.

1 Introduction

Access to large amounts of data is no longer an issue in our modern world, particularly with the rise of the Internet and other technologies that have allowed researchers to easily gain access to more data than ever before. However, access to clean, pre-processed, or otherwise high-quality data is not universal, and researchers with limited financial or computing resources may have trouble accessing or creating large, clean datasets. Datasets taken from the Internet directly are often filled with misspellings, emojis, slang, or other content that may make it difficult for state-of-the-art NLP models to process text. For our purposes, we will refer to all non-standard textual content as *noise*.

Availability of noisy datasets has increased significantly in recent years, particularly datasets taken from social media platforms such as Twitter and Reddit. Datasets from social media platforms notably have much more non-standard textual content than, for example, Wikipedia or other more formalized platforms. These recent noisy datasets have given NLP researchers many new directions to explore, but have also presented the field with new needs and challenges. Given that noisy data is only going to become more and more prevalent, as well as more and more widely available, we believe it is of the utmost importance to explore methods that allow state-of-the-art NLP models to better interact with and process noisy datasets.

Developing language models that are robust to noise can also help in extending NLP research efforts to low-resource languages, which are defined as languages that do not have sufficiently large corpora or linguistic resources available to the extent that languages commonly used in training and testing datasets (e.g. English) are. Thus efforts in this field also work towards increasing the accessibility of NLP to more languages.

A couple possible approaches for this problem are developing methods for cleaning datasets efficiently before every experiment and designing models that are inherently robust to noisy data. The former solution, however, can be computationally costly and may necessitate lots of time since the nature and shape of datasets varies widely depending on the task at hand. The latter solution is a more universal approach that, if investigated, could allow the research community to benefit from more robust datasets, regardless of the task currently being studied.

In this project, we start by providing an overview of previous work in the field of handling noise in data to then propose a data augmentation approach to tackle this issue. Our data augmentation approach includes defining a noise function that closely mimics the errors and noise found in, for

example, language on the Internet. We then apply this noise function to increase the size of our training set to include noise-added versions of training data. Finally, we demonstrate the impact of this data augmentation through the model's ability to handle noisy test sets.

2 Related Work

Several previous works have attempted to tackle the issue of noisy datasets. However, many of these approaches involved using general machine learning techniques for mitigating noise or were tested on datasets that had artificial noise injected into them. We wish to further investigate how to make models robust to noise through data augmentation techniques.

OCR Noise. Previously, researchers have discussed challenges posed by OCR (optical character recognition)-introduced error, causing noise in NLP pipelines. Typically, these are corrected via human transcription, but this is resource-intensive and not feasible for all datasets. One group of researchers proposed a framework to simulate OCR-induced noise to learn a model for noise-invariant representations (Xu et al., 2021).

Existing noise reduction methods. Researchers have also analyzed many existing noise-handling methods and found that current methods do not always improve the performance of NLP models (Zhu et al., 2022). The methods analyzed include co-teaching, noise matrices, noise models with regularization, and label smoothing. Researchers found that benefits of noise-handling methods were only obvious under high-noise levels and that BERT was resistant to injected noise but not necessarily resistant to weak supervision noise, suggesting that further research and development of noise-handling methods is needed (Zhu et al., 2022).

Taxonomy. Other researchers have proposed a framework around which to discuss noise in NLP, addressing common sources like misspelled words, slang, symbols, and time-sensitive words. The paper concludes that catch-all or general approaches to reduce noise will often succeed on certain tasks but fail on others, suggesting that a more nuanced approach is necessary (Al Sharou et al., 2021).

CNN with transition layer. Additionally, some researchers have explored the possibility of using a novel convolutional DNN model to handle noisy labels during training for sentence-level sentiment classification (Wang et al., 2019). The proposed model uses two separate alternately trained convolutional neural networks: one to handle input noisy labels using a learned noisy transition matrix and another to handle predicting "clean" labels. The model outperforms many baselines and provides a way to learn from noisy-labeled data, which is often easier to collect than "clean" data (Wang et al., 2019).

Data augmentation. Attempts at data augmentation have also been made. Easy data augmentation (EDA), which consisted of synonym replacement, random insertion, random swap, and random deletion of words in a sentence was shown to improve the performance of convolutional and recurrent neural networks for text classification. The strongest results were found in smaller datasets, and researchers found that when training with EDA, they were able to achieve similar performance using only half of available data, suggesting that EDA may be particularly useful in situations where data is scarce or not as widely available (Wei and Zou, 2019).

3 Methodology

3.1 Dataset

We trained on the CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) dataset for English Named Entity Recognition, which consists of 22137 sentences. This task was chosen for investigation because it has strong ties to other canonical NLP research tasks like part-of-speech tagging and coreference resolution. Additionally, proper nouns are more susceptible to containing typos or other non-standard textual errors, making this dataset a good candidate for this task.

3.2 Models

Due to computing resource limitations, all of our experiments were conducted using DistilBERT (Sanh et al., 2019), an approximation of the state-of-the-art BERT (Devlin et al., 2019) model. Our approach is to first create a function that adds noise to our dataset. We then create some noisy test data using the noise function. For our baseline, we will train DistilBERT on the normal, clean training data and evaluate using the noisy test data. For our experiments, we will augment our training data by using the noise function on the training data and adding the result to the training data, effectively doubling our training data. We will then train Distil-

k	Example
0.01	EU rejects German call to boycott British lamb.
0.02	EU rejects German call to boycott British lamb.
0.03	EU rejects German call 58to boycott British lamb.
0.04	EU reje102cts Germa50n call to boycott British la10mb.
0.05	EU rejects German call to 40boycott British lamb.
0.06	EU rejects 6G2erman call to boycott Br31itish lamb.
0.07	EU reject69s German call to bo18y85c87ott British lamb.
0.08	EU rejects German call to 41boyco3tt Bri58tish lamb 99.
0.09	28EU rejects German ca89ll to boycott British lamb.
0.1	Eu rej69ects German 3call t190 boycott British l25amb.

Table 1: Sample Sentences

BERT on the augmented training data and evaluate using the noisy test data.

Baseline hyperparameters. The learning rate was set to $2e{-}5$, the model was trained on 3 epochs, and the model was optimized using the Adam optimizer.

3.3 Noise Function

For our noise function, we wanted to find a way to simulate non-standard textual content through adding misspellings to the CoNLL-2003 dataset. To this end, we wrote up functions that injected realistic misspellings into the dataset. Some example sentences can be found in Table 1.

Keyboard proximity. Some text on the web generated by humans, especially by those on mobile devices, suffers from misspellings that occur with characters in the vicinity of the correct character. As such, we created a noise-generation module that generates typos intended to mimic the misspellings that occur from mistypings by substitution or insertion of nearby characters on a QWERTY keyboard. The probability that a character in a word is substituted for another is 0.5, the probability that a character is missing entirely is 0.2, the probability of a pair of characters being swapped is 0.25, and the probability of a character being inserted is 0.05. These probabilities were assigned after estimating that slips of the fingers probably result in more frequent substitutions and character swaps as opposed to missing or extra characters.

CoNLL-2003 modification. We then applied this noise function to the CoNLL-2003 dataset with a hyperparameter k indicating the level of noise.

Algorithm 1 Keyboard Distances

```
letter_coordinates ← 3-dimensional numeric coordinates according to QWERTY keyboard, (position in row left to right, row number, shift)
for character c_i in QWERTY keyboard do
    for character c_j in QWERTY keyboard do
        distances ← numerical distance between characters c_i and c_j using letter_coordinates
    end for
end for
Normalize distances
```

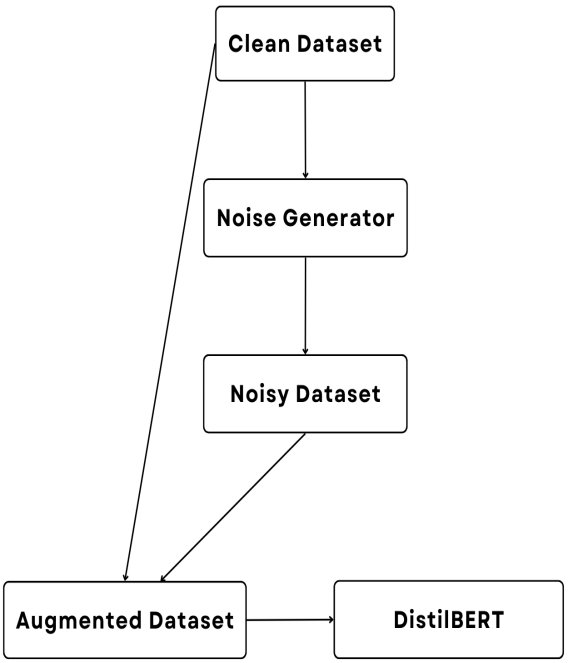
distances functions as the probability matrix of character c_j being chosen as a substitution typo for character c_i

Algorithm 2 Typo Generation Algorithm

```
typo_probability ← random number from 0 to 1
if typo_probability < 0.5 then
return substitute with nearby character on keyboard based on probability matrix distances
else if typo_probability < 0.7 then
return omission
else if typo_probability < 0.95 then
return transposition
else
return insertion
end if
```

3.4 Baseline Results (Table 2)

We trained DistilBERT on the unaltered CoNLL-2003 dataset and evaluated by adding noise to our test data. We evaluated using the precision, recall, F1, and accuracy metrics. We used noise levels of $k = 0$ (no noise), $k = 0.03$ (some noise), and $k = 0.08$ (very noisy). We chose $k = .03$ and $k = .08$ based on what we saw in example sentences (such as the ones shown in Table 1); we wanted to choose a k that represented a relatively low and reasonable amount of typos ($k = .03$) and another k that represented a higher but still reasonable amount of typos ($k = .08$).



Process Diagram

k	Precision	Recall	F1	Accuracy
0	.865	.903	.883	.968
.03	.772	.844	.806	.947
.08	.654	.747	.697	.917

Table 2: Baseline Results

From our baseline results, we can see obvious performance degradation as the noise level (k) increases. This indicates that the DistilBERT model is not very robust to noisy data.

4 Results

4.1 Training on the Augmented Datasets (Table 3)

For our results, we first augmented our dataset to include noisy samples using noise level $k = .03$ or $k = .08$ and then trained and evaluated the model using the augmented dataset. We then ran predictions using the completed model on our noisy test data and calculated the metrics shown above. We found that for both $k = .03$ and $k = .08$, training using the augmented dataset with noisy samples significantly increased the model's robustness to noise. Compared to the baseline, the models trained on the augmented dataset perform much better across all metrics, especially for $k = .08$. We can see that the models trained on the augmented dataset achieved F1 scores greater than 85% for both $k = .03$ and $k = .08$.

k	Precision	Recall	F1	Accuracy
.03	.853	.893	.872	.964
.08	.835	.881	.857	.960

Table 3: Results

Train k	Test k	Precis.	Recall	F1	Acc.
.03	0	.870	.903	.886	.967
.08	0	.870	.902	.886	.968
.03	.08	.815	.874	.843	.957
.08	.03	.859	.895	.877	.966

Table 4: Results on Varying Train/Test k Values

4.2 Model Performance on Varying Noise Levels (Table 4)

After obtaining our initial results, we wanted to see how our models trained on $k = .03$ and $k = .08$ augmented datasets performed on test datasets with no noise or a different amount of noise than the training dataset. We found there was no performance degradation when testing on the clean ($k = 0$) dataset. When testing our $k = .03$ trained model on data with $k = .08$, we found that the model still performed relatively well compared to our baseline, with an F1 score of about 84%. When testing our $k = .08$ trained model on data with $k = .03$, our model performed at the same level as what we found in our initial results.

These results show that training with noise does not impact performance on noiseless or clean data. These results also suggest that training with any amount of noise, even a small amount, helps increase model robustness to high levels of noise, as evidenced by the performance of the $k = .03$ model on the $k = .08$ data. Lastly, these results suggest that training on a higher level of noise maintains robustness to lower levels of noise, as shown by the performance of the $k = .08$ model on the $k = .03$ data. This seems to suggest that trainining with any amount of noise is better than training with no noise and that perhaps there is not a big downside to researchers augmenting training datasets with a slightly higher level of noise than they expect to encounter.

5 Conclusion

Augmenting datasets with noisy samples signficantly increases DistilBERT's robustness to noise. However, this robustness has tradeoffs.

Resources. Increasing the size of the training dataset also necessitates an increase in the amount time and computational resources for fine-tuning DistilBERT. In the case of this experiments detailed above, fine-tuning DistilBERT and training on the augmented dataset took about 10 minutes, which was about twice as long compared to working with the baseline coNLL dataset. Although the amount of additional resources and time used is highly dependent on how many noisy samples are added to the dataset, this tradeoff is something researchers should consider in the future if they choose to augment their datasets. Future directions to mitigate this issue include decreasing the number of samples in the dataset in order to maintain a workable training time.

More viable datasets. On the other hand, robustness to noise may save other computational resources and time, such as allowing researchers to test models on data that has not undergone rigorous cleaning or preprocessing. Depending on the resources available and the expected level of noisiness in test data, augmenting datasets may be a relatively simple and effective solution to making models more robust to noise.

Potentially more robust models. Because the noise function described above is intended to mimic human-like errors (misspellings and typos) in text on the Internet, training the model on the augmented dataset allows it to comprehend and parse noisy text in a manner similar to how humans process typos, bringing it one step closer to human-like understanding.

5.1 Future Work

Extending to SOTA Models. We have demonstrated on CoNLL-2003 that it is possible to improve the robustness of DistilBERT via augmentation. In the future, we suggest attempting to use data augmentation on BERT and other state-of-the-art large language models; we did not do so in this paper due to computational resource limitations.

Noise Generator. To actually implement this model for real-life application, further training is needed to develop a usable model. One aspect to consider is developing a more realistic noisy dataset by integrating Damerau-Levenshtein, an edit distance metric, into the data preprocessor. Due to computation resource limits, Damerau-Levenshtein was not used to generate typos, and the set of k's tested on was limited to two values. Optimizing the parameters such as the value of k as well as the weights assigned to different types of typos could improve the realism of the generator.

Human-generated datasets. Another aspect to consider is mining for human-generated noisy datasets, such as on social media sites, or also applying the noise generator on top of these already noisy datasets to create an augmented dataset. Human-generated noisy datasets may also help improve noise generators to be more realistic and create human-like typos.

Other tasks. This paper focused on named entity recognition, as proper nouns are often candidates for misspellings and other non-standard textual human errors. However, there are many other common task-specific text datasets, such as BillSum for text summarization, IMDB movie reviews for sentiment analysis, and glue STS-B for textual similarity, that can also be augmented with noisy samples. In this paper, we have only examined how training on an augmented dataset impacts the named entity recognition task and have not generalized the effectiveness of this approach to other NLP tasks.

Other languages. There are two sets of languages that may be of interest for extending this project: languages with markedly different grammar/linguistic structures and low-resource languages. To investigate, future works could involve running these typo generation algorithms on smaller, noisier datasets in languages other than English to see if the noise generation generalizes to other languages with different grammar structures and linguistic forms, such as Korean.

Low-resource languages. As for low-resource languages, it would also be useful to see if it is possible to gather potentially larger datasets for these languages, albeit at the cost of large amounts of noise. Then, running the model trained on augmented data may allow for the power of these models to be extended to low-resource languages as well, increasing accessibility and the use of these techniques beyond conventionally studied languages in natural language processing.

6 Acknowledgements

We thank the professors of 6.8610, Yoon Kim and Chris Tanner, for their consistent guidance, support, and teaching over the past few months. We also thank our TA and project mentor, Bowen Pan, for providing detailed insight on our project every step of the way.

7 Impact Statement

Here we discuss the ethical and societal ramifications of our work. As mentioned in the introduction, access to high-quality data sets with minimal noise is not necessarily easy to obtain for all researchers. As such, examining how to use state-of-the-art models like BERT with more noisy data which may be easier to obtain is crucial to ensure the equity of data access for researchers in natural language processing. Furthermore, not all researchers have access to reliable high-computing resources, and being able to use noisy data on smaller models like DistilBERT can help democratize progress in the research community.

However, if it can be shown that noisy data sets can train BERT to perform well on noisy data, there is the concern that more data will be gathered for use on BERT; in particular, perhaps data that was collected without consent (e.g. scraped from private channels on the web, for instance, which probably contains noise via misspellings and slang). Current state-of-the-art, low-noise data sets are often compiled with the end purpose of data collection; in other words, the subjects brought in to produce the text go through a rigorous informed consent process. Afterwards, extensive steps are usually taken to anonymize the data to protect the privacy of the users. However, gathering large amounts of data from the Internet removes these privacy-protecting steps. Thus it is also important to examine these language models and training improvements for them in the larger context of the people who contributed to the set and whom could be affected by the data collection methods.

References

Khetam Al Sharou, Zhenhao Li, and Lucia Specia. 2021. Towards a Better Understanding of Noise in Natural Language Processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 53–62, Held Online. INCOMA Ltd.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Hao Wang, Bing Liu, Chaozhuo Li, Yan Yang, and Tianrui Li. 2019. Learning with Noisy Labels for Sentence-level Sentiment Classification.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks.

Guowei Xu, Wenbiao Ding, Weiping Fu, Zhongqin Wu, and Zitao Liu. 2021. Robust Learning for Text Classification with Multi-source Noise Simulation and Hard Example Mining. ArXiv:2107.07113 [cs].

Dawei Zhu, Michael A. Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, and Dietrich Klakow. 2022. Is BERT Robust to Label Noise? A Study on Learning with Noisy Labels in Text Classification.