Bhadauria, Divya; Sierra-Múnera, Alejandro; Krestel, Ralf

Conference Paper — Published Version

The Effects of Data Quality on Named Entity Recognition

The Effects of Data Quality on Named Entity Recognition

Divya Bhadauria
University of Potsdam
Potsdam, Germany
divya.bhadauria@uni-potsdam.de

Alejandro Sierra-Múnera
Hasso Plattner Institute
Potsdam, Germany
alejandro.sierra@hpi.de

Ralf Krestel
ZBW - Leibniz Information Centre for Economics
& Kiel University
Kiel, Germany
rkr@informatik.uni-kiel.de

Abstract

The extraction of valuable information from the vast amount of digital data available today has become increasingly important, making named entity recognition models an essential component of information extraction processes. This emphasizes the importance of understanding the factors that can compromise the performance of these models. Many studies have examined the impact of data annotation errors on NER models, leaving the broader implication of overall data quality on these models unexplored. In this work, we evaluate the robustness of three prominent NER models on datasets with varying amounts and types of noise. The results show that as the noise in the dataset increases, model performance declines, with a minor impact for some noise types and a significant drop in performance for others. The findings of this research can be used as a foundation for building more robust NER systems by enhancing dataset quality beforehand.

1 Introduction

Named entity recognition (NER) is an NLP task that identifies and categorizes mentions of named entities in texts into predefined categories within a given application context (Ehrmann et al., 2021). NER models are used in many downstream applications and are becoming an integral part of their implementation (Li et al., 2022). These models must be trained on task-specific data to work well with a specific application because an NER model learns the relationship between the data elements and applies this knowledge to find similar terms in the unseen data. If the model is trained on poor-quality data, it may not learn well and most likely fail to recognize or assign the wrong category to the named entities in new, unseen data.

The term "data quality" is used in information systems to measure the goodness of the data in fulfilling the requirements of a user (Wang and Strong, 1996). Data is considered high quality if it is suitable for the intended application and does not contain errors that can undermine its use (Hassenstein and Vanella, 2022).

With the advancement and easy access to digital technology, data in different domains is widely available and growing exponentially (Hassenstein and Vanella, 2022), thus creating the need to understand the fitness of the data for the desired application. This research aims to analyze the impact of various noise types to understand the effect of data quality on the performance of NER models.

The concept of data quality was discussed in detail by Wang and Strong (1996), and the idea was to look at the quality of data from the user's perspective and divide data quality into various categories to understand their origin and impact. This study analyzes the effect of four different types of noise: spelling errors, typo errors, optical character recognition (OCR) errors, and sentence shortening errors (SSE). These errors fall into the following data quality categories (Wang and Strong, 1996):

- The intrinsic quality dimension includes a subcategory called accuracy. It is concerned with the data's reliability and integrity. Spelling, typos, and OCR errors fall under this category, as the accuracy of any textual dataset is directly affected by characters, words, and even numeric values.

- Completeness is a quality dimension in the contextual category used to determine whether data is complete and appropriate for the chosen task. When sentence-shortening errors occur, context information is lost, affecting the data's completeness.

Many NER-specific ML models do not compare the performance based on the dataset quality. After a simple data cleaning step, the main focus is on finding suitable hyperparameters during training.

Proceedings of the Ninth Workshop on Noisy and User-generated Text (W-NUT 2024), pages 79–88

There is no denying that hyperparameter tuning is an essential part of a well-trained model. However, all data-dependent models must be trained on high-quality data to make reliable future predictions on unseen data (Budach et al., 2022). The limited number of research (Hamdi et al., 2020; Bodapati et al., 2019) about the impact of data quality on NER systems creates a natural curiosity to question whether a model trained on good-quality data will make better predictions than a model trained on noisy data and if the NER-based NLP models should include data quality checks. This study observes the behavior of various models and tests their robustness with variable proportions of each error type and their combination on the CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003), WNUT 16 (Strauss et al., 2016) and Ontonotes v5 (Pradhan et al., 2013) datasets. Specifically, the focus of this research is to answer the following questions:

- **RQ1:** What impact does data quality have on the performance of each NER model?
- **RQ2:** How do different types of individual noises affect NER model performance?
- **RQ3:** What effect does combining different types of noise have on the performance of an NER model?
- **RQ4:** What effect do different datasets with different noise types have on the performance of an NER model?

We published the code of our study in `https://github.com/HPI-Information-Systems/ner-text-quality-impact`

2 Background

The effect of OCR errors on the predictive capability of four NER models was investigated in a study by Hamdi et al. (2020). The results indicate a subsequent decline in the model's performance when trained on datasets containing OCR errors. The study also suggests that understanding the impact of the frequency of this error type before applying the models can enhance the performance of NER models. The study by Bodapati et al. (2019) investigates the robustness of NER models with capitalization errors. It demonstrates that the NER models trained with the customary training procedures do not perform well when tested against textual data with either capital or small-cased letters, and the model's predictive capability suffers greatly. Multiple studies have also been conducted to understand the impact of different types of noise on AI systems, and their results show that many state-of-the-art models are susceptible to even slight variations in data (Budach et al., 2022; Belinkov and Bisk, 2018; Náplava et al., 2021; Gudivada et al., 2017). When the performance of character level and word level processing models is compared, the former models are more resilient to changes in individual characters and can still understand the meaning and context of a word if there is a minor modification in the characters of the word, such as spelling or typo errors (Heigold et al., 2017).

Gudivada et al. (2017) also discusses some of the significant issues that machine learning (ML) models face as a result of poor data quality in the ML pipeline at two stages: training and testing. Even only a few outliers in the training dataset have been shown to cause instability in the learning process of the model and show how noisy data affects the prediction capabilities of the model. Al Sharou et al. (2021) discuss the intricate relationship between data quality and NLP systems, providing a distinction between different aspects of the noise types. It categorizes noise into two categories, good and bad, and explains how it can help NLP models make better predictions. It suggests that an error that seems detrimental to one kind of task can increase the accuracy of an NLP model curated for another domain. So, the data cleaning task should not be fixed for every NLP model, and without understanding the impact of various error types, it is a challenge to build reliable data validation systems.

With numerous studies demonstrating that data quality affects model performance, this study focuses primarily on analyzing the impact of various error types and their combination in the training and prediction phase of an NER model. The findings of this study can aid in the development of data cleaning or validation systems that are required before feeding any input data to an ML pipeline.

3 Noise Types in Text

In the real world, noise is present in all textual data. Different noise types have distinct origins, thus affecting the functioning of every model differently. The following noise types have been chosen to study their effect on NER model performance.

3.1 Spelling Errors

A correctly spelled word in any language is one whose spelling matches the dictionary spelling or, if not in the dictionary, is widely accepted by well-known writers and most speakers (Al Sharou et al., 2021). Any variation in these known spellings falls under the category of spelling errors.

3.2 Typographical Errors

Typo errors occur due to mistakes in typing and are also called typos or misprints (Shah and de Melo, 2020). As more people use the internet to connect and communicate, the emphasis is not on writing everything carefully, resulting in many typos in online texts. These errors may appear to be spelling mistakes, but they are distinct because typos occur due to fast typing or fingers slipping on the keyboard.

3.3 OCR Errors

Optical character resolution, or OCR, is a technological process of converting various digitized documents into a format that computers understand (Kissos and Dershowitz, 2016). The documents generated by the OCR process can be edited like any document typed on a computer. Two contributing factors to OCR errors are the poor image quality of the documents used and the use of different training instances for the OCR image classifier.

3.4 Sentence Shortening Errors

Sentence shortening errors or cut-off (Shen et al., 2020) is a prevalent noise in textual data where a certain amount of words are missing due to informal writing, very commonly seen on social media platforms or in automatic speech recognition systems (ASR) (Cunha Sergio and Lee, 2021). Such partial removal is used to check the robustness of context-based models, especially language models, such as BERT (Devlin et al., 2019), which infer the meaning of a word in the context of the entire sentence.

4 Models

This section briefly describes the three well-known NER models selected for this study. Each model uses a different architecture to identify and extract named entities. The first is a machine learning model, and the next two are deep learning models.

4.1 Condition Random Field

Conditional random fields (CRFs) is a discriminative machine learning model that predicts data points related to each other (Sutton and McCallum, 2010). A discriminative model uses the input data to predict the output class label by creating a direct mapping between the input data and the output label (Ng and Jordan, 2002). Patil et al. (2020) explains that the CRF model uses an undirected graphical model for the named entity identification. This graph connects each observation to other observations without any specific direction. Given the context of an observation, CRFs calculate the probability of it being a particular named entity. The CRF uses the concept of feature functions to know about the various features of each variable and thus understand the relationship between them. For the study of NER datasets (Sutton and McCallum, 2010), named-entity labels are dependent on their adjacent observation, so the simplest form of CRF, called the linear CRF, is used.

4.2 BERT

Bidirectional encoder representation from transformers (BERT), proposed by Devlin et al. (2019), is a powerful, well-known, and revolutionary model in the field of NLP. The first step of BERT is pre-training, where the model is trained on an unlabeled, unstructured large dataset to understand the bidirectional context, resulting in pre-trained language models. This pre-training step is self-supervised and can be completed without labeled data leveraging the masked language modeling and next sentence prediction training objectives. Our study uses 'bert-based-cased' pre-trained model for training the models on the selected datasets. The second step is fine-tuning, where the model is further trained using an additional output layer. This training uses labeled data of specific domains or genres to learn the parameters of the new layer and update the pre-trained parameters. For the specific case of NER, each token in a sentence has a classification head responsible for identifying the labels under the IOB scheme (Ehrmann et al., 2021).

4.3 BiLSTM + Flair Embeddings

Flair is an NLP library based on the PyTorch framework, which supports multiple tasks, such as named entity recognition, part-of-speech tagging, and text classification (Akbik et al., 2019). Flair introduces its own character-based embedding technique and

provides support for various other embedding models. In this study, the Flair model uses the combination of Flair embeddings (Akbik et al., 2018) with classic word embeddings, e.g. GloVe (Pennington et al., 2014) for the CoNLL 2003 dataset, fastText (Bojanowski et al., 2017) for the OntoNotes v5 dataset, and GloVe(twitter) and fastText for the WNUT 16 dataset. Embeddings are created using the unified interface of the Flair library. This unified interface allows the implementation of various embeddings using the same code. The sequence labeling model of the Flair library is trained for NER using BiLSTMs to capture the information from both directions.

Each of the three models employs a different architecture to capture token and context meaning or any intricate information in the data. This diverse selection of models in this study is used to see which architecture is more resilient to the selected errors.

5 Datasets

Three well-known NER datasets are chosen for this experiment based on two criteria: the number of words with various class labels and the amount of noise in the dataset. The goal is to evaluate the models on small, moderate, and large datasets. All datasets contain information from different domains, and the noise level varies. Three text files containing the train, test, and validation sets are created for each dataset, following the IOB scheme. To have an idea of the amount of noise already present in the datasets, we measure existing misspellings using a spellchecker library.[1]

5.1 WNUT 16 Dataset

The first dataset selected for this research is the WNUT 16 dataset (Strauss et al., 2016). This dataset was created to analyze the challenges posed by the enormous amount of data generated on social media platforms, such as Twitter, which usually have user-generated noisy content. The WNUT 16[2] is a small-scale dataset as compared to the other two datasets considered for this study and consists of manually annotated tweets specially annotated to serve as a training ground for the NER systems. Out of the total words in the training and test set, $3,613$ (7.78%) and $7,274$ (11.75%) respectively are misspellings according to the spellchecker.

5.2 CoNLL 2003 Dataset

The second NER dataset selected for this study is the English CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003). The words in this dataset were annotated for four named entity types: person, location, organization, and miscellaneous. The English dataset was downloaded from the huggingface open source platform[3]. Out of the total words in the training and test set, $7,785$ (3.82%) and $2,584$ (5.56%) respectively are misspellings.

5.3 OntoNote v5 Dataset

The third dataset selected for this study is the OntoNotes v5 English dataset, the latest release in the OntoNotes (Pradhan et al., 2013) dataset series[4]. The dataset files were downloaded from the huggingface open source platform [5]. OntoNotes is a large-scale dataset, and along with classic NER entity types, it contains a large corpus of annotations. Out of the total words in the training and test set, $19,615$ (0.89%) and $2,822$ (0.12%) respectively are misspellings.

6 Experimental Setup

The most important task in this study is to create many different versions of train and test datasets with varying error types and rates. The subsections will briefly introduce the data augmentation steps, training process, and evaluation metrics selected for this study.

6.1 Dataset Modifications with Various Noise Types

The three datasets contain three files: train, validate, and test. The various noise types and their combinations are introduced in the train and test sets keeping the validation set untouched for all datasets in this study. For the WNUT 16 and CoNLL 2003 datasets, five datasets were generated from each train and test set for spelling, typos, OCR, and combination of all error types to conduct a thorough analysis. The error types are introduced using the NLPAug library.[6]

The number of word manipulations in a dataset varies for each error type. We decided, based on two separate studies, the minimum threshold for

[1] https://pypi.org/project/pyspellchecker/
[2] https://github.com/jinpeng01/hgn
[3] https://huggingface.co/datasets/conll2003
[4] https://doi.org/10.35111/xmhb-2b84
[5] https://huggingface.co/datasets/conll2012_ontonotesv5
[6] https://github.com/makcedward/nlpaug

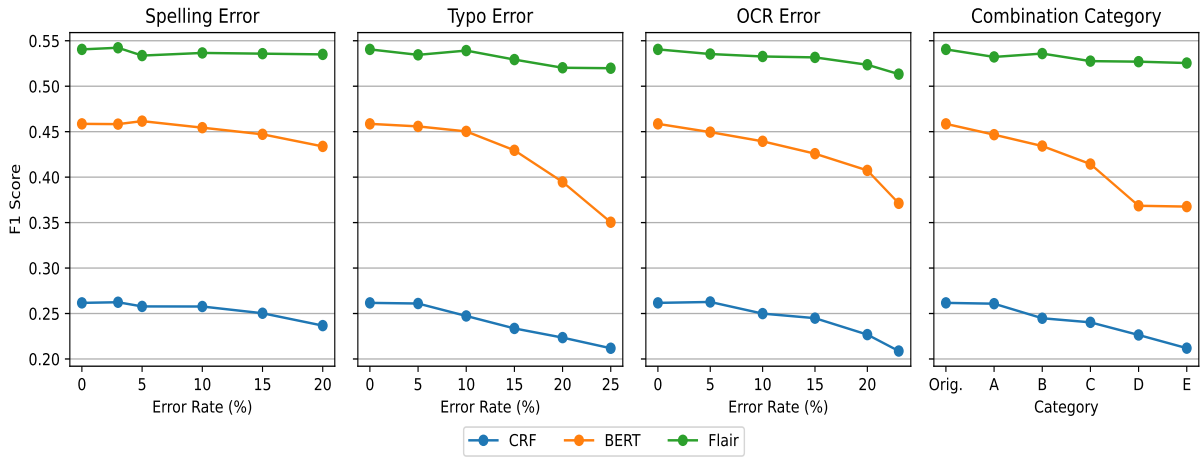

Figure 1: WNUT 16 dataset results with CRF, BERT, and Flair with various error rates for Spelling, Typo, OCR, and Combination of errors in train set

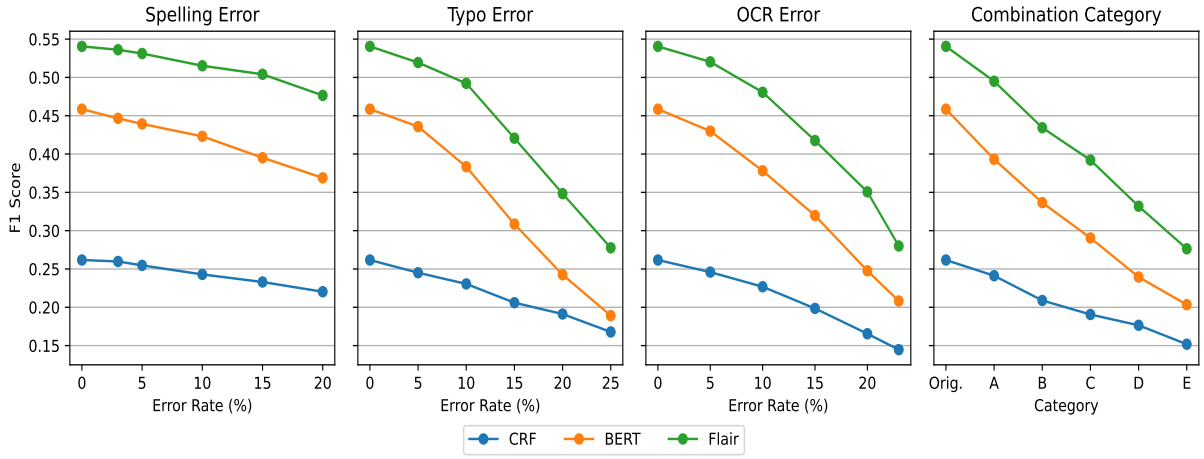

Figure 2: WNUT 16 dataset results with CRF, BERT, and Flair with various error rates for Spelling, Typo, OCR, and Combination of errors in test set

spelling (Flor et al., 2015) and typos (Rodríguez-Rubio and Fernández-Quesada, 2020) errors. The maximum threshold for OCR was taken from the study of Tong and Evans (2002), and errors are introduced in descending order from a higher to a lower number. The process of creating modified datasets with spelling, typo, and OCR errors is as follows:

- Five datasets are created for spelling errors with an increasing error rate of 3%, 5%, 10%, 15%, and 20%.
- Similar to spelling errors, five new datasets are generated for typos with the increasing error rate of 5%, 10%, 15%, 20%, and 25%.
- For OCR error, five datasets are created with an error rate of 5%, 10%, 15%, 20%, and 23%.

For the OntoNotes v5 dataset, we use only the lowest and highest error rates for each error type. As the model training using OntoNotes requires a much longer training time than the other two datasets, only two error rates are evaluated.

We follow a different process for SSE errors than the other error types. We divide the dataset into chunks of 450 words.[7] Then, we use a uniform distribution of 1 to 10 to remove words from the end of this chunk, thus creating a new dataset that simulates sentence shortening at the end of physical pages.

For the combination of errors, first, the SSE error

[7]On average, an A4 page contains 400 to 500 words, assuming it has a default margin, 12-point font size, and 1.5 line spacing. So, an average of 450 words per page is assumed for SSE errors.

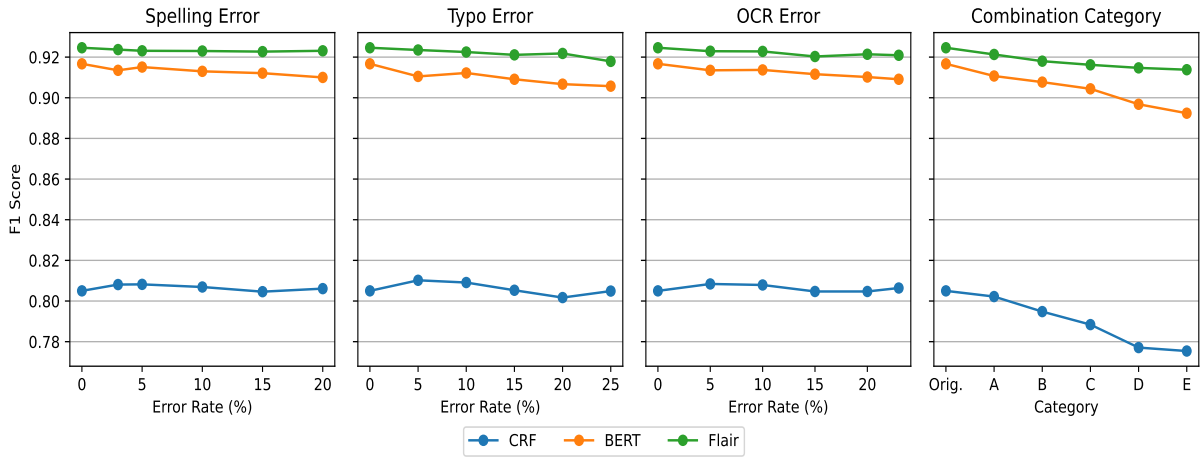

Figure 3: CoNLL 2003 dataset results with CRF, BERT, and Flair with various error rates for Spelling, Typo, OCR, and Combination of errors in train set

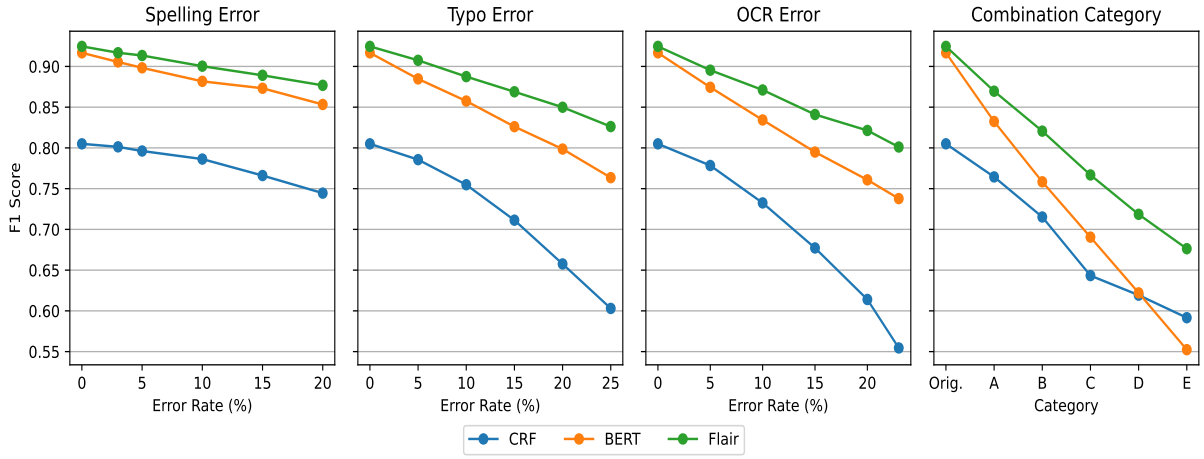

Figure 4: CoNLL 2003 dataset results with CRF, BERT, and Flair with various error rates for Spelling, Typo, OCR, and Combination of errors in test set

procedure is applied to the dataset, and then various combinations[8] of error rates are introduced to this dataset.

The process is repeated for two more seed values on the training set of each dataset, creating 15 modified training sets for each spelling, typo, OCR, and combination of errors and 3 datasets with SSE. Similarly, the test set of each dataset is infiltrated with various noise types but with only one seed value.

[8]Apply SSE then create five new datasets, A: 3% spelling error, 5% typos, and 5% OCR errors, B: 5% spelling error, 10% typos and 10% OCR errors, C: 10% spelling error, 15% typos and 15% OCR errors, D: 15% spelling error, 20% typos and 20% OCR errors, and E: 20% spelling error, 25% typos, and 23% OCR errors

6.2 Training Process

At first, each model is trained using the original train and validation sets. Then, for analyzing the impact of various noise types, the process is divided into two parts:

1. **Training the model with altered training datasets:** The model with the same configuration as the original dataset is trained with the modified train datasets. We make predictions on the unaltered test dataset to compare the model's performance with the original dataset.

2. **Testing the original model with noisy test datasets:** The model trained on the original dataset is used for predictions on noisy train datasets to analyze the effectiveness of models trained on less noisy data to predict noisy text.

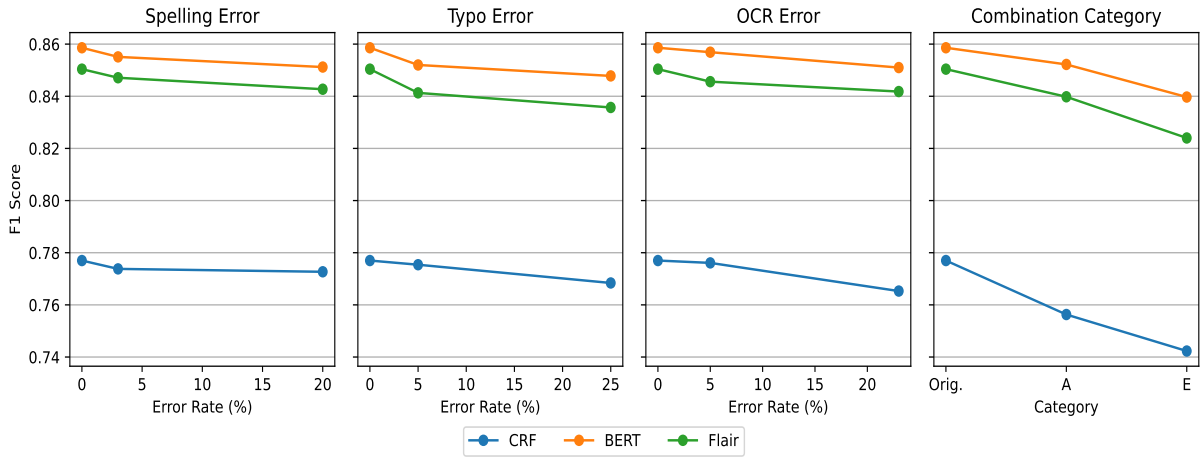

Figure 5: OntoNotes v5 dataset results with CRF, BERT, and Flair with various error rates for Spelling, Typo, OCR, and Combination of errors in train set

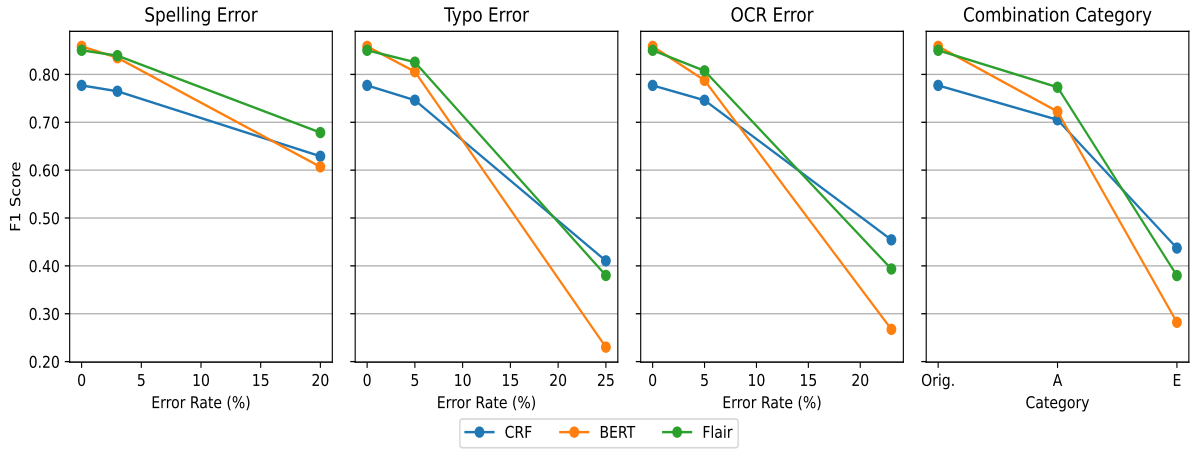

Figure 6: OntoNotes v5 dataset results with CRF, BERT, and Flair with various error rates for Spelling, Typo, OCR, and Combination of errors in test set

6.3 Evaluation Metrics

The results of all three models are presented using the micro-averaged F1 score, and for any further investigation, individual F1 scores with precision and recall for each class label are analyzed. We report the average over the different seeds.

7 Results

Two sets of experiments are performed for each model with a dataset, as mentioned in subsection 6.2. The results for each dataset are shown in two diagrams containing four subplots for spelling, typo, OCR, and combinations of all errors. The two figures for each dataset show the results with various error rates in the training and testing dataset. In plots, the numeric value 0 and the term "Orig." are used for a dataset without any added noise types.

The term F1 score in all diagrams and table shows the micro F1 score obtained from all the experiments. The result of SSE for each dataset is shown in Table 1. The F1 score obtained after all experiments indicates that the SSE does not have any significant impact on the performance of the selected models.

7.1 WNUT 16 Dataset

Of all the models' performances on the WNUT 16 dataset, the BiLSTM combined with Flair embeddings has shown the best result on the original dataset, and models trained on a noisy training set. Figure 1 shows a constant decline in the performance of both the BERT and CRF models, and the most decline in performance is observed with the combinations of various error types (0.02 for Flair, 0.09 for BERT, and 0.05 for CRF).

			SSE	
Datasets	Model	Original	Train	Test
CoNLL 2003	CRF	0.8050	0.8041	0.8030
	BERT	0.9167	0.9146	0.9152
	Flair	0.9246	0.9252	0.9233
WNUT 16	CRF	0.2617	0.2626	0.2612
	BERT	0.4586	0.4534	0.4563
	Flair	0.5405	0.5384	0.5391
OntoNotes v5	CRF	0.7770	0.7630	0.7761
	BERT	0.8586	0.8578	0.8544
	Flair	0.8504	0.8450	0.8492

Table 1: The Table shows the F1 score obtained from all three models on each dataset for SSE. The train column contains the F1 score when SSE was introduced in the training set and the F1 score is obtained on the original test set. The test column contains the F1 score when the model trained on the original dataset is tested on a test set containing SSE errors.

Figure 2 shows that the model with the original WNUT dataset, when used on noisy test datasets with an increasing error rate, suffers a steep decline in prediction capability. For the combination of errors, BiLSTM combined with Flair embeddings F1 score decreased by 0.26, BERT by 0.26, and CRF by 0.09). The BiLSTM combined with Flair embeddings, which was very robust with errors in the training dataset, did not perform well on noisy test data.

7.2 CoNLL 2003 Dataset

Figure 3 shows the overall performance of each model on the CoNLL 2003 training datasets. The BiLSTM combined with Flair embeddings performed the best on the original dataset, but the CRF model is most robust towards individual errors. Its performance declines with a combination of errors. Out of all models, BERT's performance is affected by all error types, and the most decline in its performance is observed with the combination of errors where the F1 score has dropped from 0.9167 to 0.8924. Figure 4 shows the performance of the CoNLL 2003 model trained with the original dataset and tested on the noisy test dataset. The performance of CRF on noisy test datasets shows continuous declining performance.

7.3 OntoNotes v5 Dataset

Figure 5 shows the results of models trained on a noisy training set of the OntoNotes v5 dataset. The results of BiLSTM combined with Flair embeddings show robustness to individual errors, but performance suffers when multiple error types are combined. The performance of the BERT and CRF models does not degrade significantly.

The performance of models trained on the original OntoNotes v5 dataset declines continuously, similar to the results of the WNUT 16 and CoNLL 2003 on the test dataset. Figure 6 shows that the BiLSTMs with the Flair embeddings performance is the most affected by all individual and combination errors out of all models. The model's F1 score has come down from 0.8504 to 0.2302 with typo errors in the test dataset.

The observations with respect to the research questions stated in the introduction are as follows:

RQ1: What impact does data quality have on the performance of each NER model?
The quality of a dataset has a different impact on different architectures. The BiLSTM combined with Flair embeddings shows more resilience and the best F1 score on both the original and variations of the training dataset for the WNUT 16 and CoNLL 2003 datasets. With the variations of all noise types in the test set, all models show a steep decline in performance.

RQ2: How do different types of individual noises affect NER model performance?

Individual error analysis reveals that all models are more resistant to spelling errors than typos or OCR errors. Furthermore, for the NER task, removing a small percentage of data for SSE has little effect on model performance.

RQ3:What effect does combining different types of noise have on the performance of an

NER model?

A combination of all errors, even with a small percentage of each noise type, has always resulted in decreased performance for all models on all datasets.

RQ4: What effect do different datasets with different noise types have on the performance of an NER model?

On the high-quality CoNLL 2003 dataset, the performance of each model with increased noise is not affected as much as the addition of noise to the already noisy WNUT 16 datasets.

8 Conclusion

This paper investigated the effect of different types of textual noise on NER models by artificially adding noise to training and testing datasets at different rates. Our goal was to experiment with different levels of noise based on real-world, observed levels for each category. The results showed that each error has a different impact on the NER models, with the OCR and combination of all errors having the most significant impact. The influence of errors in the test dataset is severe compared to that in the training set, and in a few cases, the high error rate shows the models' inability to make useful predictions.

Acknowledgements

This research was partially funded by the HPI Research School on Data Science and Engineering.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

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.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation.

Sravan Bodapati, Hyokun Yun, and Yaser Al-Onaizan. 2019. Robustness to capitalization errors in named entity recognition. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 237–242, Hong Kong, China. Association for Computational Linguistics.

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.

Lukas Budach, Moritz Feuerpfeil, Nina Ihde, Andrea Nathansen, Nele Noack, Hendrik Patzlaff, Felix Naumann, and Hazar Harmouch. 2022. The effects of data quality on machine learning performance.

Gwenaelle Cunha Sergio and Minho Lee. 2021. Stacked debert: All attention in incomplete data for text classification. *Neural Networks*, 136:87–96.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. 2021. Named entity recognition and classification on historical documents: A survey.

Michael Flor, Yoko Futagi, Melissa Lopez, and Matthew Mulholland. 2015. Patterns of misspellings in l2 and l1 english: a view from the ets spelling corpus. volume 6.

Venkat N. Gudivada, Amy W. Apon, and Junhua Ding. 2017. Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations.

Ahmed Hamdi, Axel Jean-Caurant, Nicolas Sidere, Mickaël Coustaty, and Antoine Doucet. 2020. Assessing and minimizing the impact of ocr quality on named entity recognition. Springer International Publishing.

Max J. Hassenstein and Patrizio Vanella. 2022. Data quality; concepts and problems. *Encyclopedia*, 2(1):498–510.

Georg Heigold, Günter Neumann, and Josef van Genabith. 2017. How robust are character-based word embeddings in tagging and mt against wrod scramlbing or randdm nouse?

Ido Kissos and Nachum Dershowitz. 2016. Ocr error correction using character correction and feature-based word classification. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 198–203.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2022. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Andrew Ng and Michael Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Adv. Neural Inf. Process. Sys*, 2.

Jakub Náplava, Martin Popel, Milan Straka, and Jana Straková. 2021. Understanding model robustness to user-generated noisy texts.

Nita Patil, Ajay Patil, and Bhausaheb Pawar. 2020. Named entity recognition using conditional random fields. *Procedia Computer Science*, 167:1181–1188.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Santiago Rodríguez-Rubio and Nuria Fernández-Quesada. 2020. Towards Accuracy: A Model for the Analysis of Typographical Errors in Specialised Bilingual Dictionaries. Two Case Studies. *Lexikos*, 30:386 – 415.

Kshitij Shah and Gerard de Melo. 2020. Correcting the autocorrect: Context-aware typographical error correction via training data augmentation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6930–6936, Marseille, France. European Language Resources Association.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the WNUT16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, Osaka, Japan. The COLING 2016 Organizing Committee.

Charles Sutton and Andrew McCallum. 2010. An introduction to conditional random fields.

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.

Xiang Tong and David Evans. 2002. A statistical approach to automatic ocr error.

Richard Y. Wang and Diane M. Strong. 1996. Beyond accuracy: What data quality means to data consumers. 12(4):5–33.