Evaluating the Impact of OCR Quality on Short Texts Classification Task

Oxana Vitman[1(✉)], Yevhen Kostiuk[1], Paul Plachinda[2], Alisa Zhila[1,2,3], Grigori Sidorov[1], and Alexander Gelbukh[1]

[1] Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico City, Mexico
ovitman2021@cic.ipn.mx
[2] Idaho National Laboratory, 83415 Idaho Falls, ID, USA
paul.plachinda@inl.gov
[3] Ronin Institute for Independent Scholarship, Montclair, USA
alisa.zhila@ronininstitute.org

Abstract. The majority of text classification algorithms have been developed and evaluated for texts written by humans and originated in text mode. However, in the contemporary world with an abundance of smartphones and readily available cameras, the ever-increasing amount of textual information comes from the text captured on photographed objects such as road and business signs, product labels and price tags, random phrases on t-shirts, the list can be infinite. One way to process such information is to pass an image with a text in it through an Optical Character Recognition (OCR) processor and then apply a natural language processing (NLP) system to that text. However, OCR text is not quite equivalent to the 'natural' language or human-written text because spelling errors are not the same as those usually committed by humans. Implying that the distribution of human errors is different from the distribution of OCR errors, we compare how much and how it affects the classifiers. We focus on deterministic classifiers such as fuzzy search as well as on the popular Neural Network based classifiers including CNN, BERT, and RoBERTa. We discovered that applying spell corrector on OCRed text increases F1 score by 4% for CNN and by 2% for BERT.

Keywords: NLP · OCR · Text classification · Multi-class classification · CNN · BERT · RoBERTa · Fuzzy search · Short texts

1 Introduction

As of today, Optical Character Recognition (OCR) systems produce a significant amount of texts in all sorts of modern activities and businesses. Think of scanned documents, store receipts, and texts of any kind that come from the camera of a smartphone with built-in OCR, the latter becoming an ever-increasing mode of OCR text data generation. The text produced by OCR – referred to in this paper as OCRed text – needs to be processed in many automated tasks, just

O. Pichardo Lagunas et al. (Eds.): MICAI 2022, LNAI 13613, pp. 163–177, 2022.
https://doi.org/10.1007/978-3-031-19496-2_13

like regular human-typed text. The automation of text processing lies in the core goal of Natural Language Processing (NLP) domain. Nevertheless, so far the majority of research and applications in NLP have focused on texts directly typed by humans, and, hence, there is still little systematic evidence on how the quality of OCRed text affects the downstream NLP tasks and what methods are best to mitigate any negative effects.

For example, previously authors conducted research on OCR for NER [1,2]. Others looked into analysis of OCR quality and provided recommendations for improving OCRed documents [16]. In the research [14] the authors looked at classification of news articles into five classes. Other experiments have been performed on data corrupted by OCR to measure the impact of its quality on such NLP tasks as sentence segmentation, NER, dependency parsing, topic modelling [18] and part of speech tagging [12].

To the best of our knowledge, the previous research on the effect of OCRed text in NLP has not addressed one large area of NLP, namely, multi-class classification of short texts. This task is frequently encountered in a variety of real-world applications, in particular, e-commerce and consumer recommender systems, which is why it is an important subject and an impactful problem to study. In this paper, we compare various approaches to multi-class classification of short OCRed text ranging from deterministic fuzzy search to classification based on neural network with CNN, BERT, and RoBERTa as well as propose and analyze methods for classification improvement.

For this, we created a dataset of 6642 short OCRed texts obtained as a result of OCR-processing of beauty products images sourced from the publicly available data collected by Open Beauty Facts project[1]. The targets for our multi-class classification are 73 unique brands corresponding to the beauty products in the set.

Our study shows that training on the OCRed text decreases the performance of CNN-based classifier in comparison to training on the human-typed data (F1-score is equal to 0.69 for training on OCR and 0.99 for training on human-typed text) , introducing spell checking on the OCRed texts can actually increase the classification performance.

We also showed that applying spell checking on the OCRed texts improves experiment results for our baseline method - fuzzy search, as well as BERT and RoERTa classifiers.

Our paper contributes the following:

- we form a dataset of short OCRed texts annotated for multi-class classification;
- comparative analysis of various approaches –deterministic and ML-based– to multi-class classification of short OCRed text;
- comparative analysis of the effect of training data quality for the same task;
- analysis of the effect of spell checker application as a means for improvement of the classification task at hand.

[1] https://world.openbeautyfacts.org/.

This paper is organized as follows. The next section presents a brief survey of relevant research works in the literature. Section 3 describes data prepossessing steps. In Sect. 4 we present classification approaches. Section 5 presents experiments and results, general conclusions, and a short discussion.

2 Related Work

The importance of OCR cannot be overstated. OCR'd texts are most commonly used in search and mining operations on digital collections. Unfortunately, OCR'd texts often contain errors.

The research [2] reports on experiments on improving OCR quality of historical text by performing correction steps and measuring the impact on named entity recognition (NER) task. They used such correction steps as removing hyphens at the end of lines and correcting *f* letters to *s* if they were a "long s" in the original document. Suggested steps improved OCR error rate by 12%, however, they didn't use spell checking.

In [16] the authors analyzed OCR quality of various prints and manuscripts of different languages and eras of history by conducting interviews with a wide range of researchers. Authors formed nine broad recommendations for improving OCR documents, declaring that researchers should develop and distribute tools for training and adapting OCR post-correction models as well as perform quantitative evaluations of the effect of OCR errors on commonly used text-analysis methods.

The paper [1] is focused on the quality of historical text digitized through OCR and how it affects text mining and NER in the context of mining big data. Experiments were performed on data extracted from historical documents by Trading Consequences project. Results show that OCR errors decrease the number of correct commodity mentions recognized: in a random sample of documents picked from several historical text collections, 30.6% of false negative commodity and location mentions and 13.3% of all manually annotated commodity and location mentions contain OCR errors.

Van Strien [18] measured the impact of OCR quality on various NLP tasks, such as sentence segmentation, NER, dependency parsing, and topic modeling. They used datasets drawn from historical newspaper collections and based their tests and evaluation on OCR'd and human-corrected versions of the same texts. According to their findings, the performance of the examined NLP tasks was affected to various degrees, with NER progressively degrading and topic modeling diverging from the "ground truth", with the decrease of OCR quality. The study demonstrated that there is still a lack of knowledge on the effects of OCR errors on this type of application, and emphasized the importance of discovering accurate methods for measuring OCR quality.

In [12] experiments were performed on Part of Speech Tagging on data corrupted by OCR as well as using artificial experiments on data representation quality. Results indicate that already a small drop in OCR quality significantly increases the error rate both on English and German data. On the contrary, the

research [17] discovered that even a relatively high level of errors in the OCRed documents does not significantly affect stylistic classification accuracy.

In research [14] authors perform automatic text classification for English newswire articles to study the impact of OCR errors on the experiment accuracies. Five categories of articles (acq, crude, earn, grain, trade) were selected from Reuters-21578 text benchmark collection[2] for English text classification, 150 articles per category. Statistical classification techniques were applied such as absolute word frequency, relative word frequency, and their power transformations. The study reveals that classification rates of OCR texts decreased with increase in OCR errors, however, transformed features improved the performance of all used classifiers. Nonetheless, Support Vector Machine (SVM) method outperformed other techniques, such as linear discriminant function and the Euclidian distance. The text classification rates for the latest were more rapidly deteriorated.

3 Dataset

As a data source for our dataset creation, we used data and images from Open Beauty Facts[3]. Open Beauty Facts is a community-driven collaborative open project that stores data about cosmetics products. The data along with the uploaded images are gathered by volunteers from 150 countries and is available under Open Database License.

The original source data contained 27.5K entries with 15,830 entries labeled with a corresponding brand.

3.1 Selection of Classification Categories

For the sake of simplicity, we focused on the task of brand classification from the text printed on the label of a beauty jar. This setting is quite straightforward as the vast majority of beauty products clearly show their brand right on the package of a product or a jar or tube itself. The feature that interested us most was that brands are often printed in a variety of sophisticated and unusual fonts that might be tricky for OCR to read correctly. An illustrative example is depicted in Fig. 1.

There were nearly 4,000 unique brands with the 90% having 6 or fewer entries. As the sparse data distribution could create unnecessary technical difficulties for ML-based classifiers, and overcoming these difficulties is not the main goal of this paper, we dropped brands with fewer than 20 entries.

This left us with 73 unique brands, transformed into 73 categorical classes. The resulting dataset contained 6,642 entries correspondingly annotated with 73 unique brands.

[2] https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection.

[3] https://world.openbeautyfacts.org/data.

3.2 Obtaining OCRed Text

The remaining entries contained URLs of corresponding beauty product images taken by volunteers. First, we fetched the images from Open Beauty Facts database. An example is shown in Fig. 1.

Fig. 1. Example of an image fetched from the Open Beauty Facts database. The corresponding brand is *Uriage*.

Further, we processed these images through native Apple OCR[4], thus, obtaining corresponding OCRed texts for all 8,456 annotated entries. A sample of these OCRed texts is provided in Table 1.

3.3 The Human Typed Text and Its Approximations

As the aim of this work is to analyze the effect that training of ML-based multiclass classifiers on OCRed text produces on the resulting quality of classification compared with the training on originally typed text, we needed to obtain reasonable originally typed equivalents for the OCRed texts. The authors admit that the ideal approach would be manual correction of the results of OCR processing. However, at a scale of 8,456 strings, it was not feasible within the resources of this research.

[4] https://developer.apple.com/documentation/vision/vnrecognizetextrequest.

Table 1. Illustration of OCRed text.

#	OCRed text examples
1	Smooth on Sioin PALMOLIVE NATURALS AOOr CAMELLIA OIL & ALMOND EL OE ROOCRONON
2	NIVEA Deme 100 ml
3	Tdi Signal® INTEGRAL S ACTER MOWEAU (94nt) (CO) EXPERT SENSIBILITE Arocne don oeto aca pone toude to botcne CHnE UTICACITE CLINIQUEMENT PROUVEE • SOIN APAISANT
4	P YVES ROCHER Pomme Rouge Red Apple E MUINS • HAND CREAN
5	Baby talc Peaux délicates Atuchan

Therefore, we proposed and implemented 2 approaches that served as approximations of originally typed text. By design, these approaches bear varying levels of spelling errors introduced by OCR.

Concatenated Product Name and Brand. The Open Beauty Fact database has multiple columns representing various characteristics of a beauty product. While some of them, e.g., `city`, `store`, are irrelevant for our purpose, others contained textual information that corresponded to some extent to the information provided on the label of the product. After thorough analysis of the fields, we concluded that concatenating fields `name` and `brand` would serve as the best available approximation to what is actually typed on the product face.

We would like to note that the information in those columns has been typed in by the volunteers. Therefore, it includes only as many words corresponding to the label as the volunteer provided. A few examples of the resulting texts and the corresponding OCRed texts are provided in Table 2.

Table 2. Examples of HTT texts.

OCRed text	Human Typed Text (HTT) text
NIVEA comon protect & Care 4 PROTECTION A COL CEETATU	Nivea Protect & Care - Déodorant anti-transpirant 48h
compressé compressé Dove OFFRE invisible dry LOT to c 48h x2 clean touch anti traces bipe vtoot	Dove Dove Déodorant Femme Spray Anti Transpirant Invisible Dry Compressé 100ml Lot de 2
Monsavon au lait y/ Fleur Goton Go wute l6gère Vion Geste raschd 48-8	Monsavon Monsavon Déodorant Anti-transpirant Spray Femme Fleur de Coton 200ml
NOUVEAU Sc warz. opf SMOOTH'N SHINE SOIN HUILE HYDRATANT ue Huiles 4Moringa O Olive t slmot el Caraseed	Schwarzkopf,Smooth'n shine Soin huile hydratant
WELEDA After Shave Balsam pflegt und beruhigt natürlich frisch Ganzheitliche Naturkosmetik	Weleda After Shave Balsam

While the texts in the HTT dataset variant are not properly spelled, they were typed by humans. Therefore, we assume that the distribution of spelling errors is different from the type of distortion introduced by OCR. Our goal is to analyze whether and how the performance of classifiers differs for HTT texts from OCRed texts.

Spell-corrected OCRed Texts. Another approximation to "humanizing" the OCRed text consisted in passing it through a spelling corrector. For spell correction, we employed FuzzyWuzzy[5] library, version 0.18.0. This library uses Levenshtein Distance to calculate the differences between sequences of characters. The spell checker relies on the `difflib.ratio`. This function returns a measure of the sequences' similarity (float in [0,1]). Where T is the total number of elements in both sequences, and M is the number of matches, the matching score is calculated as is $\frac{2*M}{T}$. Note that this is 1 if the sequences are identical, and 0 if they have nothing in common. For each word, it scans over the user-provided dictionary and returns the string in which each token is replaced by a word from the user-provided dictionary if the similarity score obtained from `fuzz.ratio` is higher than 0.75. We used the list of unique brands as our user-provided dictionary.

The illustration of resulted texts is provided in Table 3.

Table 3. Examples of spell-corrected OCRed texts.

OCRed text	Spell-corrected OCRed text
GARNICR Ultra DOUX CORPS	garnier Ultra DOUX CORPS
LSEVE HULE	elseve HULE
WVELEDA Depuis 1921 Bébé CALENDULA	weleda Depuis 1921 Bebe CALENDULA

As a result, we obtained 3 variants of the annotated dataset with varying degree of text distortion introduced by OCR:

1. "as-is" OCRed texts directly from an OCR processor;
2. a spell-checked version of the OCRed texts;
3. human typed texts approximately lexically equivalent to the OCRed text.

To provide numerical evaluation of the degree of distortion introduced by OCR, we calculated the aggregated numbers of exact brand string matches for each dataset and then computed the decrease (in percent) of exact match as compared to the Human Typed Text, see Table 4. As expected, the spell correction procedure removes certain fraction of the distortion placing the spell-checked OCRed variant between the OCRed one and HTT, slightly closer to the OCRed text.

[5] https://pypi.org/project/fuzzywuzzy/.

Table 4. Aggregated counts of exact substring match of a brand string in text entries for each dataset variant. The last column evaluates the distortion introduced by OCR as the decrease of matched brand strings as compared to the Human Typed Text.

Test set	# of entries with exact match	% of dataset size	% decrements from HTT
OCR	727	54.7	−27.6
SC OCR	837	63.0	−16.7
HTT	1004	75.5	0

3.4 Train/Test Split

The dataset was split into train and test parts, leaving 80% for the train part, which is equal to 5313 samples, and 20% for the test part, which equals 1329 samples accordingly. The train and test parts contain same row IDs across all variants ensuring fair comparison in our experiments.

The resulting dataset including three variants and the train-test split has been made publicly available[6].

4 Classification Approaches

We performed the analysis for several popular classifiers based on neural networks such as CNN, BERT-based classifier, RoBERTa-based classifier. For each of the classifiers, we first performed hyper-parameter tuning to determine the configuration that worked best for each variant of text. In particular, for CNN we validated input text representation via different embeddings, FastText, $\text{GloVe}_{\text{Wiki}}$, and $\text{GloVe}_{\text{CommonCrawl}}$. For BERT and RoBERTa we checked the cased and uncased models as well as base and large versions.

We also included a deterministic "classifier" based on fuzzy substring search in our analysis.

4.1 Fuzzy Substring Search

Substring search can be considered a deterministic form of a classifier. If a given brand string is found within a given text, it may be considered a "hit". Of course, it happens that more than one brand string may be found within the same text. To resolve these conflicts, we use a score provided by the fuzzy search algorithms.

Similar to the spell correction procedure described in Sect. 3.3, we used `FuzzyWuzzy` library, version 0.18.0, and Levenshtein Distance to provide a score for a potential match.

The fuzzy search relies on the `fuzz.partial_token_sort_ratio` function. It attempts to account for similar strings without regard to the token order. It sorts tokens in each string and then calls `fuzz.partial_ratio`, which in turn calls `fuzz.ratio` using the shortest string (length n) against all n-length substrings of the larger string and returns the highest score.

[6] https://github.com/Wittmann9/DataImpactOCRQuality.

This returns best matching brand as a substring of the OCRed text string, alongside with the matching score.

4.2 CNN

In this paper, we used the Convolutional Neural Network (CNN) architecture after [10]. The majority of the hyperparameters were set as proposed in [21].

CNN takes as its input text representation via vector embeddings. Commonly, popular word embeddings such as Glove [15] and FastText [9] are used. As those embeddings have been trained on different kinds of text corpora ranging from Wikipedia to CommonCrawl Corpus[7], they may have different initial information that may be helpful to a different degree for the task at hand. One can argue,for example, that while Wikipedia may contain articles on the majority of famous beauty brands, it's unlikely that it has information on all possible products produced by these brands and, hence, sufficient exposure to the vocabulary used on beauty jar labels.

On the other hand, Common Crawl Corpus is a huge corpus of a large chunk of the Internet, and the corresponding embeddings have been built from an extremely assorted vocabulary that is likely to have included beauty products.

In this paper, we have conducted preliminary experiments with different embeddings to select the optimal representation for OCRed texts in beauty product domain.

For out-of-vocabulary words the embeddings were initialized randomly (random vector sampled from U[-0.25, 0.25]).

The overall architecture can be described as follows. The padded embedded sentences are processed via the CNN cells. Next, the ReLU activation function and Max Pooling are applied. The concatenated outputs from the previous step are processed by linear layers to produce final class distribution.

Text Preprocessing. Before classification, all texts in all three dataset variants were preprocessed by converting them to lower case and standardizing Unicode symbols by applying `unidecode` package[8] for Python. It takes a Unicode character and represents it as an ASCII character mapping between two character sets in such a way that a human with a US keyboard layout would choose. For example, é → e.

GloVe Embeddings. As mentioned earlier, GloVe has various versions trained on different text corpora and varying in word vector dimension length. In particular, one version of GloVe embeddings was trained on Wikipedia and an archive of English newswire text named Gigaword 5 which contains 6 billion tokens. This version is commonly used in research. Another version is GloVe Common Crawl, 300-dimensional word vectors trained on 42 billion tokens. We compared which

[7] https://nlp.stanford.edu/projects/glove/.
[8] https://pypi.org/project/Unidecode/.

of the versions would provide the best results for our dataset which belongs to a beauty product domain.

FastText. We have experimented with FastText pretrained word vectors [4], trained on 600 billion tokens from the Common Crawl corpus.

The results of training with various word embeddings are shown in Table 5. For all variants of texts, GloVe vectors for word representation trained on Common Crawl corpus (GloVe_{CC}) outperformed or were at par with the others. Therefore, we further used the CNN trained using this particular word vector representation.

Table 5. Choosing among various word vector representations for CNN-based classifier.

Test	$\text{Model}_{\text{train}}$	FastText				GloVe_{CC}				$\text{GloVe}_{\text{Wiki}}$			
		Acc	F1	P	R	Acc	F1	P	R	Acc	F1	P	R
OCR	CNN_{OCR}	0.74	0.66	0.73	0.64	**0.76**	**0.69**	**0.76**	**0.67**	0.70	0.60	0.68	0.58
	CNN_{SC}	0.72	0.63	0.69	0.63	0.74	0.68	0.75	0.65	0.70	0.60	0.64	0.59
	CNN_{HTT}	0.53	0.43	0.57	0.43	0.66	0.62	0.72	0.61	0.40	0.27	0.36	0.31
SC OCR	CNN_{OCR}	0.77	0.66	0.72	0.65	0.77	0.70	0.77	0.68	0.71	0.60	0.67	0.59
	CNN_{SC}	0.76	0.67	0.72	0.66	**0.79**	**0.73**	**0.80**	**0.71**	0.73	0.63	0.67	0.62
	CNN_{HTT}	0.55	0.44	0.55	0.44	0.69	0.64	0.73	0.64	0.42	0.28	0.36	0.32
HTT	CNN_{OCR}	0.17	0.15	0.29	0.14	0.88	0.80	0.82	0.80	0.16	0.14	0.28	0.13
	CNN_{SC}	0.15	0.15	0.27	0.14	0.88	0.81	0.85	0.80	0.17	0.15	0.26	0.14
	CNN_{HTT}	0.98	0.97	0.98	0.97	**0.99**	**0.99**	**0.99**	**0.99**	0.98	0.96	0.96	0.96

4.3 BERT

Bidirectional Encoder Representations from Transformers (BERT) [6] is a very powerful transformer-based [19] deep learning model suitable for a variety of NLP tasks including short text classification [3,8,22].

Unlike the CNN, which we trained from scratch in this paper using the previously trained word embeddings from existing work, BERT model is a contextual representation model that has been pre-trained on large text corpora, BooksCorpus [23] and English Wikipedia, and comes with pre-trained weights.

To use it in our tasks of multi-class classification on short texts, we applied fine-tuning. BERT model is initialized with its pre-trained weights, and new additional classification layers (a linear layer on top of the pooled BERT output) are initialized with random weights. After that, we trained the model for each of our labeled training sets.

BERT model exists in various configurations: (1) as trained on cased or uncased text; (2) in a base size and large size that differ in the number of layers and, consequently, parameters. The larger model has been shown generally to outperform the BERT-base model [6]. Therefore, we perform our experiments on this version.

In our research, we used transformers python package [20], which provides pre-built transformers tokenizers and models. The preprocessing pipelines, which are required for the model are also included. We used these pipelines for the training and testing.

To choose between the cased or uncased versions of BERT, we performed the configuration experiments shown in Table 6. We observed that while the cased version performs better on spell-checked OCRed text as well as Human-typed texts, the OCRed text slightly benefits from using the uncased version as compared to the cased version. Therefore, we use the corresponding best-performing results for the final analysis in Table 7.

Table 6. Experiments with cased and uncased BERT model to choose the optimal settings for each variant of the dataset.

Test set	$\text{Model}_{\text{Train}}$	**Cased**				**Uncased**			
		Acc	F1	P	R	Acc	F1	P	R
OCR	BERT_{OCR}	**0.8021**	**0.7263**	**0.7776**	**0.7097**	**0.8043**	**0.7287**	**0.7948**	**0.7093**
	BERT_{SC}	0.4680	0.3517	0.4211	0.3643	0.1993	0.1891	0.2956	0.1878
	Bert_{HTT}	0.4266	0.3538	0.4090	0.3902	0.2151	0.0936	0.1198	0.1116
SC	BERT_{OCR}	0.6049	0.4640	0.5258	0.4710	0.3265	0.1947	0.2367	0.2100
	BERT_{SC}	**0.8073**	**0.7483**	**0.7968**	**0.7306**	0.7456	0.6965	0.7737	0.6691
	BERT_{HTT}	0.4070	0.3549	0.4182	0.3957	0.179	0.0740	0.1082	0.0795
HTT	BERT_{OCR}	0.5921	0.4265	0.4961	0.4580	0.1316	0.1261	0.3280	0.1050
	BERT_{SC}	0.5665	0.6399	0.6399	0.5277	0.1091	0.1040	0.2886	0.0881
	BERT_{HTT}	**0.9962**	**0.9938**	**0.9953**	**0.9934**	0.4236	0.2675	0.3604	0.2711

4.4 RoBERTa

A Robustly Optimized BERT Pretraining Approach (RoBERTa) [11] is an even better performing modification of BERT trained longer with bigger batch size and a number of other modifications. RoBERTa was shown to achieve superior results in various NLP tasks. In particular, recent works [5,7,13] demonstrate RoBERTa's highest results in text classification. Therefore, we included this model in our comparison to see whether it proves more stable against the spelling errors intrinsic to OCR as opposed to human text typing.

For the experiments, we used the RoBERTa large version of the model.

5 Experiments and Results

We have trained CNN and BERT classifiers on three sets of data: human-typed text, spell-corrected OCR text, and OCR text so that we could observe how our classifiers perform on texts of different quality. Each model we then evaluated on the same three types of texts.

5.1 Evaluation Metrics

For evaluation, we use Accuracy, macro-averaged F1, Precision, and Recall as implemented in scikit-learn metrics package[9]. Macro-average metrics calculate the designated value for each label and find their unweighted mean.

5.2 Experiments

For the purpose of analyzing the impact of the degree of distortion introduced by OCR on the multi-class classification task, we conducted a series of experiments. For each of the classifiers outlined in Sect. 4 apart from the fuzzy search, for each variant of the dataset with varying degrees of distortion introduced by OCR described in Sect. 3, we first trained the classifier on the training part of the dataset variant for those classifiers where training applies and then inferred the classes for the testing parts of all dataset variants.

As fuzzy search is a deterministic classifier and, hence, does not require any training, it was applied directly to the test parts of our dataset.

The results of our experiments are presented in Table 7.

Table 7. Impact of OCRed texts on the performance of various classifiers. The columns correspond to the test parts of the dataset variants with corresponding level of text distortion introduced by OCR. SC stands for spell-corrected OCRed text.

Model$_{train}$	**Test OCR**				**Test SC OCR**				**Test HTT**			
	Acc	F1	P	R	Acc	F1	P	R	Acc	F1	P	R
Fuzzy	0.059	0.013	0.01	0.02	0.060	0.014	0.01	0.022	**0.065**	**0.016**	**0.03**	**0.02**
CNN_{OCR}	**0.76**	**0.69**	**0.76**	**0.67**	0.77	0.70	0.77	0.68	0.88	0.80	0.82	0.80
CNN_{SC}	0.74	0.68	0.75	0.65	**0.79**	**0.73**	**0.80**	**0.71**	0.88	0.81	0.85	0.80
CNN_{HTT}	0.66	0.62	0.72	0.61	0.69	0.64	0.73	0.64	**0.99**	**0.99**	**0.99**	**0.99**
$BERT_{OCR}$	**0.80**	**0.73**	**0.79**	**0.71**	0.47	0.35	0.42	0.36	0.43	0.35	0.41	0.39
$BERT_{SC}$	0.60	0.46	0.53	0.47	**0.81**	**0.75**	**0.8**	**0.73**	0.41	0.35	0.42	0.40
$BERT_{HTT}$	0.59	0.43	0.50	0.46	0.57	0.64	0.64	0.53	**0.99**	**0.99**	**0.99**	**0.99**
$RoBERTa_{OCR}$	**0.82**	**0.76**	**0.81**	**0.75**	0.70	0.58	0.63	0.58	0.58	0.70	0.73	0.72
$RoBERTa_{SC}$	0.59	0.47	0.57	0.47	**0.78**	**0.70**	**0.77**	**0.69**	0.68	0.57	0.67	0.58
$RoBERTa_{HTT}$	0.46	0.43	0.51	0.46	0.37	0.36	0.44	0.40	**0.99**	**0.99**	**0.99**	**0.99**

From Table 7, we observe that while fuzzy search generally performs poorly on the multi-class classification compared to the NN-based methods, it gets least confused by the human-typed text, and its performance increases from OCRed text to HTT.

For all learning-based models, we observe that the best results are achieved for the same distribution as the model was trained on, e.g., the model trained on OCRed text performs best on the OCRed test. Interestingly, CNN shows smaller

[9] https://scikit-learn.org/stable/modules/model_evaluation.html.

variance for a given test set across training on texts with varying OCR distortion level (that is, looking at the results column-wise in Table 7) compared to the BERT-based models. However, both BERT and RoBERTa achieve higher results for the OCR test set. Interestingly, for the SC OCR while BERT is showing the best performance, CNN classifier managed to outperform RoBERTa. The HTT set was trivial to learn from for all the models.

One of the main takeaways of this analysis is that adding spell correction as a pre-processing for the OCRed text tends to improve classification, at least for CNN and BERT. In particular, F1 metric has increased by 4% for CNN and by 2% for BERT.

In general, our analysis suggests that in order to obtain better performance for classification of OCRed texts, it has to be processed to reduce the distortion introduced by OCR.

Acknowledgments. The work was done with partial support from the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, grants 20220852 and 20220859 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

1. Alex, B., Burns, J.: Estimating and rating the quality of optically character recognised text. In: Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage, pp. 97–102 (2014)
2. Alex, B., Grover, C., Klein, E., Tobin, R.: Digitised historical text: does it have to be mediocre? In: KONVENS, pp. 401–409 (2012)
3. Amjad, M., et al.: Urduthreat@ fire2021: shared track on abusive threat identification in Urdu. In: Forum for Information Retrieval Evaluation, pp. 9–11. FIRE 2021, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3503162.3505241
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
5. Briskilal, J., Subalalitha, C.: An ensemble model for classifying idioms and literal texts using BERT and RoBERTa. Inf. Process. Manage. **59**(1), 102756 (2022)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, June 2019. https://doi.org/10.18653/v1/N19-1423
7. Guo, Y., Dong, X., Al-Garadi, M.A., Sarker, A., Paris, C., Aliod, D.M.: Benchmarking of transformer-based pre-trained models on social media text classification datasets. In: Proceedings of the The 18th Annual Workshop of the Australasian Language Technology Association, pp. 86–91. Australasian Language Technology Association, Virtual Workshop, December 2020. https://aclanthology.org/2020.alta-1.10

8. Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Zhao, T.: SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 2177–2190. Association for Computational Linguistics, July 2020. https://doi.org/10.18653/v1/2020.acl-main.197, https://aclanthology.org/2020.acl-main.197
9. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
10. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, 25–29 October 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1746–1751 (2014). https://aclweb.org/anthology/D/D14/D14-1181.pdf
11. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach (2019). arxiv.org/abs/1907.11692
12. Mieskes, M., Schmunk, S.: OCR quality and NLP preprocessing. In: WNLP@ ACL, pp. 102–105 (2019)
13. Murarka, A., Radhakrishnan, B., Ravichandran, S.: Classification of mental illnesses on social media using RoBERTa. In: Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis, pp. 59–68. Association for Computational Linguistics, April 2021. https://aclanthology.org/2021.louhi-1.7
14. Murata, M., Busagala, L.S.P., Ohyama, W., Wakabayashi, T., Kimura, F.: The impact of OCR accuracy and feature transformation on automatic text classification. In: Bunke, H., Spitz, A.L. (eds.) DAS 2006. LNCS, vol. 3872, pp. 506–517. Springer, Heidelberg (2006). https://doi.org/10.1007/11669487_45
15. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
16. Smith, D.A., Cordell, R.: A research agenda for historical and multilingual optical character recognition, p. 36. NULab, Northeastern University (2018). https://ocr.northeastern.edu/report
17. Stein, S.S., Argamon, S., Frieder, O.: The effect of OCR errors on stylistic text classification. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 701–702 (2006)
18. Van Strien, D., Beelen, K., Ardanuy, M.C., Hosseini, K., McGillivray, B., Colavizza, G.: Assessing the impact of OCR quality on downstream NLP tasks (2020)
19. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017). https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
20. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. Association for Computational Linguistics, October 2020. https://doi.org/10.18653/v1/2020.emnlp-demos.6, https://aclanthology.org/2020.emnlp-demos.6
21. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820 (2015)

22. Zhao, Z., Zhang, Z., Hopfgartner, F.: SS-BERT: mitigating identity terms bias in toxic comment classification by utilising the notion of "subjectivity" and "identity terms". CoRR abs/2109.02691 (2021). arxiv.org/abs:2109.02691
23. Zhu, Y., et al.: Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 19–27 (2015). https://doi.org/10.1109/ICCV.2015.11