Context-Relevant Denoising for Unsupervised Domain-Adapted Sentence Embeddings

Michael Lowe*, Joseph D. Prusa*, Joffrey L. Leevy*, and Taghi M. Khoshgoftaar*
*Florida Atlantic University
Email: mlowe2020@fau.edu, josephdprusa@gmail.com, jleevy2017@fau.edu, khoshgof@fau.edu

***Abstract*—In closed-system domains, such as healthcare databases, record scarcity and data quality often act as barriers to applying state-of-the-art language processing techniques. Addressing these challenges requires the adjustment of both domain and task to effectively deliver meaningful value. A common approach for adapting domains with limited and poorly annotated data is data augmentation. *Transformers and Sequential Denoising Auto-Encoder*s (TSDAEs) offer an inductive, unsupervised pre-training method that efficiently leverages unlabeled data by learning from many-to-one corrupted training samples. This approach reduces the need for extensive manual data annotation typically associated with domain adaptation. We advance this method by using transduction-based noise generation, which simulates the kind of noise commonly encountered in text generation within targeted domains. Our study investigates the effects of corruption and contextual noise introduced by this augmentation, thus enhancing the practical ability of domain-adapted models in specialized fields.**

***Index Terms*—optical character recognition, sequence-to-sequence models, text-to-text transformers, data augmentation, noise correction**

I. Introduction

With the growing adoption of generative *Small Language Model*s (SLMs) and *Large Language Model*s (LLMs), many industries and domains have recently expanded their use of generative language frameworks. Although these domains present unique challenges, several universal issues arise in optimizing domain adaptation processes. One significant challenge is concept drift [1], which involves monitoring the effects of evolving vocabulary and task sets on the definition of a domain. This issue is frequently encountered when altering the domain of a language model, presenting characteristics that can negatively impact comprehension. Such phenomena, along with other challenges, are often observed when large samples of domain-specific data are not available. In these instances, methods such as weak supervision and data augmentation are implemented to enrich the context of the targeted domain further. Frameworks like the Sentence Transformers library utilize state-of-the-art language model implementations to adapt domains using various learning tasks [2].

The *Transformers and Sequential Denoising Auto-Encoder* (TSDAE) framework is a pre-training technique focused on denoising as its learning objective [3]. We chose this method for its potential as a transduction-based learner, which refers to the ability of a model to directly learn mappings from specific inputs to outputs without needing a separate explicit training phase. This is particularly important for applications where model adaptability to new, unseen data is critical. The TSDAE framework uses the distillation capabilities of auto-encoders to map corrupted (augmented) data samples to their ground-truth counterparts. These samples are generated through an arbitrary noise function, introducing this mapping as a formal learning task. The noise function implemented within the framework utilizes a token deletion algorithm, where the ratio of deletions can be adjusted as a tunable parameter.

Building upon the reliance on generalized noise functions, we investigate the application of noise from systems that generate data in specific domains, which often face performance issues with out-of-the-box pre-trained models. For instance, the fields of healthcare and information sciences encounter difficulties in applying language models to their data, owing to the nature of data extraction. This constraint leads to heightened scrutiny over data reliability and quality, delaying access for institutions and public forums. Utilizing data from these sources necessitates improved data quality and annotation. Consequently, we examine whether data quality can serve as an annotation medium at the character level, aiming to shift from inductive token-level reasoning to transduction in the pre-training phase. Specifically, we question whether character-level errors can be utilized as a standalone augmentation technique and whether this constitutes a feasible objective for training a language model.

To address this question, we compare the standard implementation of denoising with generated corruption. We introduce *Optical Character Recognition to Sequence* (OCR2Seq) as a method to recreate the document extraction process in a controlled environment for sequence generation. By employing a combination of synthetic and runtime corrupted errors, we examine the impact of context-based learning using token-based and tokenless models, namely Google/T5, as our learners.

The remainder of this paper is organized as follows: Section II covers related work; Section III describes the methodology; Section IV presents our findings and discusses their significance; and Section V condenses the main points of this paper, with suggestions for future work.

II. Background and Related Work

This section reviews the foundational concepts on which our proposed methods are based. We primarily explore data augmentation, text extraction engines, their impact on training language models, and the challenges these methods pose

2835-5776/24/$31.00 ©2024 IEEE
DOI 10.1109/IRI62200.2024.00016

in the precision-health machine learning domain. We begin by reviewing standard and state-of-the-art data augmentation techniques currently implemented to support both weak and unsupervised learning tasks, discussing their primary areas of success and their influence on model robustness to noise. Additionally, we introduce text extraction engines, their role in data generation within digital storage systems, and the challenges in achieving accurate translation. Finally, we apply the aforementioned topics to explore the challenges currently faced in addressing noise-tolerant pre-training tasks for closed domains with scantily annotated data sources. The concept of annotation is then discussed in the context of the challenges posed by language processing and comprehension in precision health.

A. Data Augmentation

Data augmentation creates synthetic instances of samples from concrete datasets with the goal of strengthening the decision boundary of targeted inference tasks [4]. The introduction of augmented samples into a training task serves as a form of implicit regularization [4], [3]. Not only does data augmentation address challenges of overfitting with smaller, close-domain data sources, but it can also enhance the decision boundary of task-specific processes.

Although text augmentation techniques are commonly used in the context of deep learning, the impact of the complexity associated with augmentation is still being actively explored [4]. The process of selecting an augmentation strategy that has a positive impact, as opposed to an adverse one, is usually performed on a case-by-case basis depending on the dataset in question. However, it can also be determined by the specific model architecture in use. One of the most notable examples of tailoring data augmentation is when using token-based language models [5], [6].

1) Data Augmentation and Transformers: Transformer architectures are extremely powerful and robust deep learning models based on encoder-decoder architectures, featuring a built-in multi-head attention mechanism. This mechanism captures sequential, probabilistic, and multi-layer nuances within a specific set of queries in batched input [7]. Most augmentation strategies use this property and are designed around token-based manipulation strategies. Token deletion, substitution, and sentence rephrasing are typical examples of data manipulation. A challenge faced with augmentation is the insertion of out-of-vocabulary instances into a dataset, i.e., unintentional adverse noise. Over-augmentation has been shown to decrease the performance of downstream tasks due to overgeneralization introduced by the high volume of noise [8], [9]. This performance drop-off is attributed to the implementation of implicit regularization at the tokenization level, rather than by the language model itself.

2) Application of Augmentation in Pre-training: A few examples of augmentation-driven adaptation pre-training methods include MPNet [10], the successor to *Masked Language Modeling* (MLM) and *Permuted Language Modeling* (PLM). MPNet is an autoregressive technique that uses masked token-level prediction as its primary training objective. Another example is the TSDAE framework. Finally, *Generative Pseudo-Labeling* (GPL) is a semi-supervised approach that uses generative data augmentation to harvest task-specific annotated samples from general-purpose language models for downstream fine-tuning [11]. These pre-training methods represent only a subset of a wide range but offer insight into the how and why behind the value of data augmentation in sparsely annotated domains. One of the main challenges associated with these techniques, especially TSDAE, is the impact of the built-in noise function across varying domains and the opportunity to extend capability with more domain-specific data generation methods.

B. Text Extraction Engines

Text extraction engines are software systems designed to convert electronically stored documents into character-encoded, human-readable text. These systems facilitate the conversion of various stored data types such as JPEG, TIFF, PNG, WAV, and MP3 into character-encoded transcripts, making the information easily interpretable by data processing systems and, ultimately, machine learning algorithms. For digital archive management, *Optical Character Recognition* (OCR) is widely used for converting warehoused and historical documents into digital assets [12], [13]. Due to the inherent challenges of translating modalities from image to text, models that incorporate data collected from OCR systems often exhibit systemic noise issues. These issues are recognized as common performance pitfalls in closed-domain tasks, where the precision of data extraction and interpretation is critical [14]

Mitigation strategies to address OCR errors often include manual and keyword corrections, alongside other rule-based methods. The optimization stage of an OCR process can vary, depending on the specific deficiencies identified for a given application [15]. Currently, deep learning approaches, particularly those involving transformer architectures, include strategies like layered object detection for establishing robust extraction boundaries, and sequence-to-sequence transformers aimed at post-OCR correction [16]. These approaches encompass various levels of text recognition and document adjustments such as resolution, skew, and object detection. Text extraction involves implementing more robust segmentation strategies and augmented data. Lastly, post-processing entails applying vocabulary-based adjustments and sequence-to-sequence transformations [9].

C. Domain-Specific Applications

Data quality for language modeling tasks is widely acknowledged as a risk factor when implementing application-specific tasks, but it is often addressed with particular sensitivity in healthcare due to the lack of annotation and variety in subdomain vocabulary. The limited open-source support and restricted access to data have often been definitive blockers in adopting clinical language models.

Advancements in mitigating access as a restriction have been addressed by several publicly available clinical models, datasets, and standardized tasks [17]. Many of these resources are supported by growing communities similar to their open-source counterparts, making contributions a continual process. An example of such ecosystems, and one of the datasets used for this study, is the *'Medical Information Mart for Intensive Care* (MIMIC) family of critical care databases. It is a closed-system healthcare database in a public, redacted data repository for researchers [18], [19]. Like its parent domain, the bulk of MIMIC data consists of plain text records, including lab results, intake forms, provider notes, and discharge summaries, all of which contain clinical shorthand and spelling errors.

Applications within healthcare are generally limited by the volume of training samples due to both proprietary and government regulations. The ability to train clinical models given the shortage of both quality documents and training data is the primary area of improvement this paper aims to address.

III. Methodology

To generate and implement an appropriate noise function, we examine the overall structure of the Sentence Transformers library and the implementation of noise within it. We extend the implementation of this noise function by applying synthetically generated OCR errors to lines of text given to a system. Unlike the token level corruption most commonly associated with augmentation techniques, we explore implementing corruption at the character level to influence sentence structure, an error most commonly encountered in records generated from OCR engines. An illustration of this corruption comparison is depicted in Figure 1. We take corruption at the sequence level, which acts at the token level, and expand corruption to the byte level. This introduces the concept of spelling and extraction error exposure while maintaining the semantic structure of the leveraged sequences.

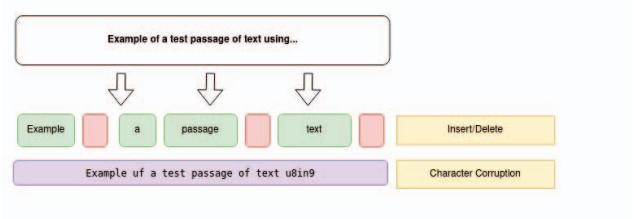

Fig. 1: Upper: Example of a Deletion-based Augmentation Strategy. Lower: Example of a Character-based Augmentation Strategy

A. Dataset

Over the course of this case study, we utilized two information archives as our primary data sources: *CommonCrawl News* (CC-News), a repository of archived worldwide news events in various languages; and MIMIC-III/IV, a publicly available intensive care data mart. To fully grasp the impact of OCR behavior as an objective, we reproduced these text repositories as mocked documents, referred to as MockDocs. The design of our dataset allows us to produce images for OCR consumption, incorporating common cosmetic errors that tend to generate more erroneous output from their data processors.

1) MIMIC-IV: MIMIC is an open-source, de-identified clinical database containing the records of over 40,000 critical care patients from the Beth Israel Deaconess Medical Center, spanning the 2000s decade (2001-2012). The text included in this dataset comprises doctors' and nurses' notes related to patient events, procedures, pharmacy prescription notes and descriptions, and *Internal Classification of Diseases* (ICD) coding with descriptive terminology. This table is a structured yet flattened table combining plain unstructured text with related mappings back to a patient record at several different levels. These observation levels include patient care, medication dosage, diagnostics, intake, and bedside care.

2) CC-News: CC-News is a public data source housing news events from around the world, provided through a series of web crawlers [20]. While this source is commonly used for language document summarization, semantic search, and other vector-intensive tasks, in this study, we use it for document generation. Having text as a ground truth allows us to recreate paper news documents to apply text extraction and synthetic generation techniques. This approach serves as a baseline prior to expanding into a closed-system domain.

B. Document Creation

We developed a pipeline using the Abstract Factory design pattern [21] to preserve the document structure and base assignment logic, incorporating several implementations of behavioral flaws to generate our augmented samples. As a measure for sequence-to-sequence learning, these generated data points serve as autoregressive tasks for learning patterns within the logic behind our imperfections.

Each experiment is designed to address the following processes:

1) **Ingestion**: The first objective of this stage is identifying the origin, structure, and volume of the data source. Other areas of interest include the presence of, or necessity for, metadata required to perform downstream tasks. These tasks can vary widely and include sub/multi-task classification, annotation for generating document summaries, and classic translation. After augmentation, we save these categorizations as metadata.
2) **Transformation**: Depending on the complexity of the data source, a denormalization or aggregation strategy may need to be implemented to generate data in the manner required for document creation. For instance, with the MIMIC datasets, we aggregated admissions data, patient notes, labs, and *Current Procedural Terminology* (CPT) codes around specific events in the patient's stay. This approach helped us summarize distinct documents or care episodes, aligning with the generative documents we aimed to produce. In other cases, like news articles,

the focus is more on document generation and deliberately introducing challenges for text extraction.

3) **Document Generation**: Each document use case comes with unique creation attributes, but we consistently use the *Free Portable Document Format* (FPDF) library [22] for template creation. This open-source document rendering package offers essential features and introduces specific challenges for physical document extraction.

C. Noise Function Generation

We apply our principal objective to produce two types of OCR documents. Systemic documents are generated from cosmetic flaws parameterized at the OCR engine level, and probabilistic documents are generated from varying parameters in the NLPAug data augmentation library [23].

1) NLPAug Noise Function: The initial noise function applied throughout the training process involves text corruption via the NLPAug library. To tailor our noise function, we set the following high-level hyperparameters: a word-level error rate of 0.3 and a character-level error rate of 0.3, ensuring a minimum of 1 word is corrupted and 2 characters are edited within the string. We chose this configuration as the baseline for our experiment to maintain the context of the text. Please refer to Table I for the detailed hyperparameter configuration. The selection of specific values for the different hyperparameters in this table was guided by the default configuration provided in the NLPAug library documentation for generating OCR errors. This choice ensures the creation of OCR synthetic errors while maintaining the structural integrity of the sentences. Preserving sentence structure is important for effective chunking, which is necessary for training sentence transformers.

Parameter	Value
aug_char_p	0.3
aug_char_min	2
aug_char_max	10
aug_word_p	0.3
aug_word_min	1
aug_word_max	10
min_char	1
tokenizer	Word Level nltk

TABLE I: Hyperparameters for Text Corruption

2) Systemic Noise Function: The systemic noise function involves recreating the preprocessing workflow commonly associated with the development of OCR pipelines. We focus on tweaking specific parameters often used for denoising images processed at runtime. In conjunction with PyTesseract [24], we use OpenCV [25] and the FPDF library to modify the structure, encoding, size, and resolution of the documents selected for this study. This process involves taking our sample text and rendering it onto a blank template in an FPDF document, where we apply cosmetic errors and processing noise. Our most common tactics for recreating these errors include malformed character encoding, marks across the text, irregular segmentation strategies, and resolution scaling.

D. Measuring Augmentation Quality

We begin by defining the dynamic components that the Levenshtein edit distance equation [26] considers when determining the difference between two strings. The transformations are as follows:

Insertions: Extra characters inserted into a string in comparison to the original parent string observed in this dynamic comparison.

Deletions: Characters missing from a string in comparison to the observed parent.

Substitutions: A string may retain the same length as the observed parent string but have different characters, typically referred to as substitutions.

These transformations are expressed within our error rate equation as follows:

- S: number of character substitutions
- D: number of character deletions
- I: number of character insertions
- Z: any arbitrary text transformation for observation
- N: number of samples

Each sample point and the level of granularity are defined by the task assigned, such as character error rate, word error rate, or line error rate.

$$ERR = \frac{\sum_{i=1}^{I} T_i}{N}; \quad T = \{S, D, I, \ldots, Z\} \tag{1}$$

In our error rate equation, Z is included alongside other transformations (substitutions, deletions, and insertions) to capture any additional text changes that do not fall neatly into the other categories. The Z transformation is determined at the sentence level for OCR errors. It represents an arbitrary text transformation observed in our measurements of augmentation quality. By incorporating Z, we aim to account for all potential variations introduced during OCR processing, thus providing a comprehensive measure of augmentation quality.

Character Error Rate: Serves as a dynamic character mapping to gauge how accurately each character is translated during document extraction. This metric offers the most granular level of statistical analysis achievable with our tokenless transformer-based approach, broken down word by word at the evaluation level.

Word Error Rate: Acts as a sequence-to-sequence measurement of the accuracy with which each word in a line of text is represented during document extraction. At the word level, which serves as the primary granularity level for general use and extension into sequence prediction in language modeling tasks, considerations include word insertion, masked language modeling, and sequence translation.

Our objective is to statistically analyze the overall effectiveness of the technique and determine if each augmentation strategy is statistically distinct from the others, thereby validating a robust augmentation strategy. Similar to our aim with augmentation strategies, we intend to increase the variance of our training set without compromising the integrity of our original dataset. By applying the three transformations

and examining their distribution and probability density, we aim to identify each technique as a meaningful, independent augmentation strategy. Given the data distribution presented later, we chose the Kolmogorov-Smirnov test [27] to compare each independent sample distribution against one another rather than against a normal distribution, due to the segmented nature of page chunks during document extraction.

E. Measuring Comprehension

To initiate our setup, we choose the pooling architecture as detailed in the TSDAE case study [3]. Following this, we evaluate it alongside Byt5 [9], a model known for its effectiveness in processing data at the byte level. Byt5 is part of a broader group of models designed for natural language processing tasks, and its unique approach to tokenization makes it particularly suitable for tasks involving denoising.

Domain adaptation typically involves a pre-training component along with task-specific fine-tuning once the domain has shifted. Given the nature of our dataset, we opt for the common method of scoring weakly supervised or unsupervised processes: language perplexity. Perplexity is one of the most commonly used statistical metrics for evaluating the quality of a language model, measuring how well a language model predicts an unseen sequence of text.

Perplexity is usually calculated during model training when a cross-entropy loss function is implemented. We define our loss function as follows:

$$H(p,q) = -\sum_{x}^{X} p(x)\log_2(q(x)) \tag{2}$$

Perplexity is defined as the exponential of our cross-entropy loss function. This metric was selected as a means to measure how well our model comprehends language from denoising our augmented samples.

$$Perplexity = 2^{H(p,q)} \tag{3}$$

IV. Results and Discussion

We analyze the distributions of each augmentation technique in relation to one another, given our previously established error rates. We document the following in terms of the average rate of corruption along with their respective distributions, as shown in Tables II, III, and IV, and Figures 2 and 3.

With regard to Table II, the noise function used by TSDAE introduces the largest edit distance but the smallest set disjoint of erroneous vocabulary due to the simplicity of noise injection, as indicated in Table II. Noise is introduced at the sentence level by either adding or removing a token. Although edit distance scores can be drastically influenced by removals, the impact depends on the tokenization strategy leveraged by a language model, affecting either the entire string or only a small part of it. Our proposed method of generative augmentation provided the most significant shift in edit distance distribution and erroneous token cardinality. We observed more distinct errors, introducing segmented words, misencoding of characters, and skipped words. The NLPAug data augmentation technique gains a more granular level of control over noise injection at the byte/character level. This approach shows improvements in the outlying variance, but our default setting for the task only allows for a slight increase in noise, which drastically influences our edit distance. When analyzing all three techniques, we assess whether there is truly a distinction in terms of statistical independence that can provide insight into the quality of our augmentation strategy.

With respect to the MIMIC documents, we first notice the change in distribution behavior compared to its plain text counterpart generated by the CC News write-ups, as illustrated in Figures 2 and 3. This current distribution follows closer to a bimodal distribution once applying our three augmentation strategies to our data. Due to this constraint, we chose to use a Kolmogorov-Smirnov test, concluding that the analysis is still possible due to the closely aligned means and probability density associated with each transformation, as detailed in Table III.

The advantage of creating augmented samples and their respective sources affords the opportunity for our dataset to be used as a denoising task. We observe the three augmentation types and their impact on domain comprehension by calculating the language perplexity of each implemented model, as seen in Table IV. We note that TSDAE, as an augmentation type, had the highest perplexity given our standard learning rate of 2e-5. This makes the TSDAE noise function-based augmentation approach the least performant in terms of predicting sequences within a domain's given probability distribution. Character-level augmentation styles, such as NLPAug's synthetic OCR simulation method and our direct OCR of MockDocs, yielded results showcasing more optimal perplexity scores due to the permutation of sentence structure. This potentially changes the semantics of the sequence as opposed to insertion or deletion.

		tsdae_edit	ocr_edit	nlp_aug_edit
CC News	**mean**	6.98	4.20	4.27
	std	1.14	2.26	0.82
	min	4.09	0.69	2.64
MIMIC	**max**	9.58	10.07	7.26
	mean	6.10	2.73	4.07
	std	0.59	0.50	0.32
	min	4.04	0.69	2.89
	max	7.43	3.97	4.88

TABLE II: Statistics for Data Augmentation Strategies

	Group 1	Group 2	KS Distance	P-value
MIMIC	tsdae_edit	ocr_edit	0.939	0
	ocr_edit	nlep_aug_edit	0.825	0
	nlp_aug_edit	tsdae_edit	0.946	0
CC News	tsdae_edit	ocr_edit	0.471	0
	ocr_edit	nlp_aug_edit	0.972	0
	nlp_aug_edit	tsdae_edit	0.852	0

TABLE III: Kolmogorov-Smirnov Analysis

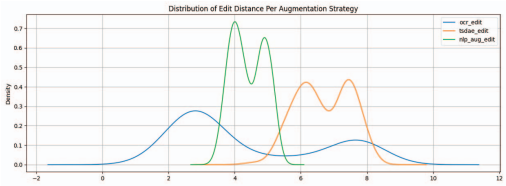

Fig. 2: Edit Distance Probability Density of MIMIC-IV

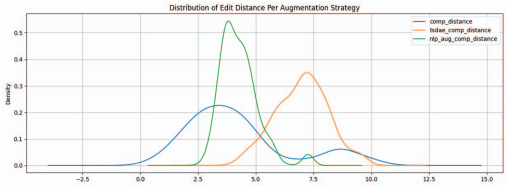

Fig. 3: Edit Distance Probability Density for CC-News

	Model	Perplexity Score
CC News	Byt5_nlp-aug	1.342
	Byt5_tsdae_aug	5.126
	Byt5_ocr	1.453
MIMIC	Byt5_nlp-aug	1.198
	Byt5_tsdae_aug	3.627
	Byt5_ocr	1.346

TABLE IV: Language Perplexity Scores

V. Conclusion

This study successfully validates the use of synthetic and reproduced OCR errors as an effective data augmentation strategy for sequential denoising, emphasizing its applicability in domain-specific tasks characterized by underrepresented data samples and scarce annotated resources. Through meticulous comparison of error distributions among different augmentation strategies, namely TSDAE, NLPAug, and our proposed OCR error simulation, the results affirm that our approach not only maintains statistical independence but also enhances the robustness of language models to noisy environments.

Future work to expand our technique includes exploring how this approach affects downstream tasks, particularly classification, to test robustness against the noise introduced to shifted domains.

References

[1] C. Ding, J. Zhao, and S. Sun, "Concept drift adaptation for time series anomaly detection via transformer," *Neural Processing Letters*, vol. 55, no. 3, pp. 2081–2101, 2023.

[2] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: http://arxiv.org/abs/1908.10084

[3] K. Wang, N. Reimers, and I. Gurevych, "Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning," *arXiv preprint arXiv:2104.06979*, 2021.

[4] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *Journal of big Data*, vol. 8, pp. 1–34, 2021.

[5] E. Shushkevich, M. Alexandrov, and J. Cardiff, "Bert-based classifiers for fake news detection on short and long texts with noisy data: A comparative analysis," in *International Conference on Text, Speech, and Dialogue*. Springer, 2022, pp. 263–274.

[6] A. Srivastava, P. Makhija, and A. Gupta, "Noisy text data: Achilles' heel of bert," in *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, 2020, pp. 16–21.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[9] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, "Byt5: Towards a token-free future with pre-trained byte-to-byte models," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 291–306, 2022.

[10] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mpnet: Masked and permuted pre-training for language understanding," *Advances in neural information processing systems*, vol. 33, pp. 16 857–16 867, 2020.

[11] K. Wang, N. Thakur, N. Reimers, and I. Gurevych, "Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval," *arXiv preprint arXiv:2112.07577*, 2021.

[12] M. L. Jockers and T. Underwood, "Text-mining the humanities," *A new companion to digital humanities*, pp. 291–306, 2015.

[13] F. Lihui and T. Underwood, "The core issues and latest progress of current digital humanities research: An interview with ted underwood," *Foreign Literature Studies*, vol. 43, no. 6, p. 1, 2021.

[14] M. Jiang, Y. Hu, G. Worthey, R. C. Dubnicek, T. Underwood, and J. S. Downie, "Evaluating bert's encoding of intrinsic semantic features of ocr'd digital library collections," in *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, 2021, pp. 308–309.

[15] R. Smith, "An overview of the tesseract ocr engine," in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, vol. 2. IEEE, 2007, pp. 629–633.

[16] A. Hemmer, J. Brachat, M. Coustaty, and J.-M. Ogier, "Estimating post-ocr denoising complexity on numerical texts," *arXiv preprint arXiv:2307.01020*, 2023.

[17] E. Lehman and A. Johnson, "Clinical-t5: Large language models built using mimic clinical text," 2023.

[18] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi, and R. Mark, "Mimic-iv," *PhysioNet. Available online at: https://physionet.org/content/mimiciv/1.0/(accessed August 23, 2021)*, 2020.

[19] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[20] F. Hamborg, N. Meuschke, C. Breitinger, and B. Gipp, "news-please: A generic news crawler and extractor," in *Proceedings of the 15th International Symposium of Information Science*, March 2017, pp. 218–223.

[21] N. Nahar and K. Sakib, "Automatic recommendation of software design patterns using anti-patterns in the design phase: A case study on abstract factory." in *QuASoQ/WAWSE/CMCE@ APSEC*, 2015, pp. 9–16.

[22] O. Plathey, "Fpdf," http://www.fpdf.org/, 2024.

[23] E. Ma, "Nlp augmentation," https://github.com/makcedward/nlpaug, 2019.

[24] S. Saoji, A. Eqbal, and B. Vidyapeeth, "Text recognition and detection from images using pytesseract," *J Interdiscip Cycle Res*, vol. 13, pp. 1674–1679, 2021.

[25] G. Bradski, "The opencv library." *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[26] L. Yujian and L. Bo, "A normalized levenshtein distance metric," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.

[27] V. W. Berger and Y. Zhou, "Kolmogorov–smirnov test: Overview," *Wiley statsref: Statistics reference online*, 2014.