PEaCE: A Chemistry-Oriented Dataset for Optical Character Recognition on Scientific Documents

Nan Zhang[*♣], **Connor Heaton**[*♣], **Sean Timothy Okonsky**[†], **Prasenjit Mitra**[♣♠], **Hilal Ezgi Toraman**[†‡◇]

[♣]College of Information Sciences and Technology, The Pennsylvania State University, USA
[†]Department of Chemical Engineering, The Pennsylvania State University, USA
[♠]L3S Research Center, Leibniz University Hannover, Germany
[‡]Department of Energy and Mineral Engineering, The Pennsylvania State University, USA
[◇]Institutes of Energy and the Environment, The Pennsylvania State University, USA
{njz5124, czh5372, sto5087, pmitra, hzt5148}@psu.edu

Abstract

Optical Character Recognition (OCR) is an established task with the objective of identifying the text present in an image. While many off-the-shelf OCR models exist, they are often trained for either scientific (*e.g.*, formulae) or generic printed English text. Extracting text from chemistry publications requires an OCR model that is capable in both realms. Nougat, a recent tool, exhibits strong ability to parse academic documents, but is unable to parse tables in PubMed articles, which comprises a significant part of the academic community and is the focus of this work. To mitigate this gap, we present the **P**rinted **E**nglish **a**nd **C**hemical **E**quations (PEaCE) dataset, containing both synthetic and real-world records, and evaluate the efficacy of transformer-based OCR models when trained on this resource. Given that real-world records contain artifacts not present in synthetic records, we propose transformations that mimic such qualities. We perform a suite of experiments to explore the impact of patch size, multi-domain training, and our proposed transformations, ultimately finding that models with a small patch size trained on multiple domains using the proposed transformations yield the best performance. Our dataset and code is available at https://github.com/ZN1010/PEaCE.

Keywords: Optical Character Recognition (OCR), Chemistry-Oriented Document Analysis, OCR Dataset, Image to Text

1. Introduction

For documents that are available only as scans, before text processing and natural language processing can be applied, we must convert the images that represent the text in these documents into digital characters before they can be processed to understand their content. In order to do so, Optical Character Recognition (OCR) is widely used to extract texts from images in various real-world applications (Memon et al., 2020; Ye et al., 2018) and can complement other data extraction pipelines (Zhang et al., 2022; Wang et al., 2020). Extracting text from images of both scientific texts (*e.g.*, math and physics formulae) and generic printed English plays a vital role in data extraction of scientific articles. A model that is capable in both realms is necessary. Important information in scientific documents is often presented in the form of tables, making data extraction even more difficult.

However, existing open-source OCR models and datasets tend to focus on either scientific texts or generic printed English. Thus, their performance on the documents that contain both is suboptimal. For example, Pix2tex (Blecher, 2020), a pre-trained model that achieves competitive performance on images of math formulae, is less capable on images containing printed English texts. Tesseract is commonly used to extract vanilla printed English, but it cannot be fine-tuned directly on images of scientific texts, because it outputs plain text strings without formatting for contents such as superscript and subscript (Smith, 2007).

Furthermore, simply combining a vanilla printed English and scientific training corpora is unlikely to yield strong performance on records that are a hybrid of the domains for two reasons. First, there will likely be inconsistencies in the way labels from each corpus are presented - records from the scietific corpus may contain LaTeX formatting not present in the vanilla printed English corpus. While this may be able to be rectified, a second issue still remains - the model will be presented with records containing each type of text separately, but will never be presented with records containing *both* types of text. That is, a model trained on such a corpus will never see scientific text interspersed with vanilla printed English.

In this work, we seek to address the inability of existing tools to format text including super/subscripts and other special characters in academic and scientific papers. Such text is predominantly printed in English, but often have specially-formatted and im-

[*]Equal contribution.

LREC-COLING 2024, pages 12679–12689
20-25 May, 2024.

portant characters. For example, a document may contain mentions of chemical compounds such as Na_2CO_3. An OCR model needs to recognize both subscripts, and also discern the values in the subscripts as they denote important physical properties of the compound.

Although one may assume it reasonable to segment the text by plain English/special characters and apply domain-specific models on each resulting group, we believe doing so is not the ideal solution. First, doing so would introduce redundant computation into the pipeline. That is, each segment of the image would have to be analyzed twice - once in order to classify the textual content, and again to synthesize it. Furthermore, in doing so, a model would not be able to leverage the temporal dependency in the text. For example, the text "Na_2CO_3 $(Sodium\ carbonate)$" would get segmented into "Na", "$_2$", "CO", "$_3$", and "$(Sodium\ carbonate)$", and the model would need to synthesize each segment in isolation. We believe a stronger model can be learned processing the entire record at once, leveraging the temporal dependency between the chemical compound, Na_2CO_3, and its name in English, $Sodium\ carbonate$.

Nougat, a newly released model, can perform OCR on entire pages of academic documents, including parsing tables, but struggles *significantly* in parsing tables from documents published in PubMed[1] (Blecher et al., 2023). As the authors describe, PubMed papers often present tables as embedded images. So they could not access the ground-truth text present therein without an expensive annotation process. As such, the model often fails to recognize tables in such documents, and when tables are recognized, they are rarely parsed correctly. PubMed hosts a large portion of papers in the life-sciences and biomedical domains, leaving a significant portion of the academic community unaddressed or under-addressed.

We aim to address the above shortcomings by proposing a new data resource, since there does not exist an OCR dataset that contains images of both scientific texts and printed English to the best of our knowledge. Thus, we introduce PEACE (**P**rinted **E**nglish **a**nd **C**hemical **E**quations) dataset, containing synthetic and real-world images of text from academic articles, with a particular focus on chemistry papers. Each record in PEACE is intended to resemble a cell in a table that may appear in an academic document (a handful of words across two or three lines), and the code we release exposes parameters that make it easy for researchers to generate records of any length and format they desire. A model trained on PEACE could then be combined with a state-of-the-art (SOTA) table parsing model such as Multi-Type-TD-TSR (Fischer et al., 2021) to parse the content of tables identified in scientific documents, addressing the market Nougat cannot.

PEACE has two parts: 1) synthetic records and 2) real-world records. The synthetic portion of the dataset contains 1M images of printed English text, 100k images of numerical artifacts, and 100k images of (pseudo-)chemical equations, subset in mutually exclusive training/dev/test splits. The real-world test set comprises 319 carefully curated records and assesses the performance of OCR models on text from actual chemistry scholarly papers. Figure 1 shows a data instance from the real-world test set of PEACE. All labels in PEACE are LaTeX[2] markup as it is a versatile typesetting tool that can express the multitude of non-alphanumeric characters often found in academic papers such as "$\sum$", "$+$", and "$\mathbb{R}$", and can format super/subscripts. Given that the real-world records contain corruption not found in synthetic records, we propose three transformations - pixelation, bolding, and whitespace padding - to mimic these artifacts.

Figure 1: A data instance of PEACE. The source image is outlined in blue, the LaTeX label in green.

We explore how two different versions of the Vision Transformer (ViT) perform when trained on PEACE, and our findings can be summarized in three folds. First, although all models perform well on the synthetic tests sets, they exhibit a sharp decline in performance on the real-world test set. This reinforces our motivation for proposing PEACE, as existing datasets to not reveal this shortcoming. Then, we observe that the patch size parameter has a significant impact on resulting performance, with smaller patches leading to better performance. Further, we see that models trained in a multi-domain setting, *i.e.* both PEACE and im2latex-100k, yield better performance in each domain than a model trained on a single domain. Finally, we also observe that our proposed image transformations improves performance in two of our three test datasets.

This paper mainly contributes the following:

1. We propose a novel dataset that contains images of both scientific texts and printed English for training and testing OCR models on articles from the hard sciences, with an emphasis on Chemistry.

2. We demonstrate how models that perform well on related datasets perform significantly worse

[1] https://www.ncbi.nlm.nih.gov/pmc/

[2] https://www.latex-project.org/

on our PEACE real-world dataset, highlighting the value of this new resource.

3. We present a set of quantitative evaluations to show the effect of patch size, multi-domain training, and image transformations.

2. Related Work

2.1. "Hybrid" Dataset for OCR

We did not find any relevant datasets that contain images of both scientific texts and printed English for OCR models on scientific documents. Therefore, we list the recent efforts here that are closest to this paper. Zharikov et al. (2020) proposed DDI-100, a dataset of distorted document images. Since the labels of DDI-100 are text strings with corresponding locations, this work concentrates vanilla printed English (*i.e.* no formatting for super/subscripts) and is not as effective on scientific documents. Furthermore, this dataset contains images of entire documents, but we are primarily concerned with recovering text from smaller scope images without figures (*e.g.*, individual table cells).

Deng et al. (2019) proposed a large table recognition dataset, which they dubbed TABLE2LATEX-450K from scientific documents. Its images contain complete tables whereas ours is focused on cells in tables in images. It is infeasible to match cell contents (in LaTeX format) with their corresponding pixels in the table images. Thus, we cannot create a dataset of images that contain scientific text. In addition, work has been done to collect photographs of random academic papers under factors such as non-uniform lighting, strong noise, sharpening, skew, and blur (Kišš et al., 2019). For our purposes, artifacts such as sub-optimal lighting are not overly important as the models will be presented with images from PDF, or sometimes *scanned* documents, which may incur their own distinct set of artifacts.

2.2. Vision Transformer

Transformers have been used in a variety of disciplines, including in computer vision (CV), where the variant was simply dubbed the Vision Transformer (ViT) (Dosovitskiy et al., 2020). The ViT retains many of the features of the original transformer designed for machine translation but processes a sequence of image patches instead of token embeddings. As described in Figure 2, images are first segmented in to $P * P$ patches of non-overlapping pixels before each patch is projected $\mathbb{R}^{P*P} \Rightarrow \mathbb{R}^{D_M}$ where D_M is the internal dimension of the model.

After exploring patch sizes of $32 * 32$, $16 * 16$ and $14 * 14$, the authors of ViT ultimately conclude that smaller patch sizes allow for better performing models at the cost of computation. Strudel et al. (2021), reinforce this conclusion, observing "... the performance is better for large models and small patch sizes". They primarily investigated patch sizes of $32 * 32$ and $16 * 16$, only training a single model with the smaller $8 * 8$ patch size. For the $8 * 8$ model that was trained, they found conflicting results: one performance metric shows improvement while the other a degradation.

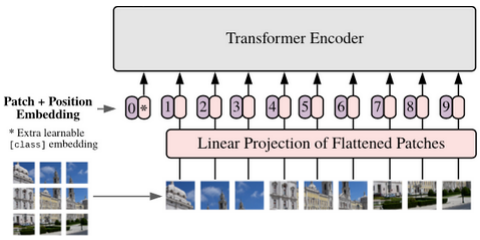


Figure 2: Patch projection in the ViT.

Than et al. (2021) explored patch sizes ranging from $16 * 16$ to $256 * 256$ when classifying chest x-rays by whether or not the patient had COVID-19, with best performance achieved using a patch size of $32 * 32$. In general, the community has seemingly adopted a patch size of 16*16 as the *de facto* standard for the ViT, in line with the originally proposed *ViT-Base*.

2.3. Pix2Tex

Pix2Tex (Blecher, 2020) is a competitive OCR model for scientific text that employs both a ViT encoder and decoder. Its ViT encoder comes with a ResNet backbone (He et al., 2015), which means that several ResNet layers are adopted to extract features from source images that are then fed into the ViT encoder. That is, the ResNet backbone is used in place of a patch-projection module. The training data for the released model checkpoint was a combination of im2latex-100k and math formulae collected from other sources on the web. By training Pix2Tex from stracth on PEACE, we are able to evaluate the capability of this architecture on scientific text, printed English, or both.

2.4. Tesseract

Tesseract (Smith, 2007) is a popular OCR model that was originally trained for generic printed English. By fine-tuning it on PEACE under a multi-domain training setup, Tesseract can possibly detect specicial characters such as super/subscripts on scientific documents by incorporating more vocabulary. The core of Tesseract is an LSTM network (Hochreiter and Schmidhuber, 1997). However, since there was a significant performance gap between Tesseract and Pix2Tex in our preliminary experiment, we do not pursue Tesseract as a baseline in this paper.

2.5. Math OCR

MI2LS is currently the best performing model we are aware of on the im2latex dataset (Wang and Liu, 2021). The model consists of a CNN encoder and RNN decoder, augmented with attention (Bahdanau et al., 2014). Training the model consisted of two phases: token level and sequence level. During token level training, training is based on traditional maximum likelihood estimation (MLE). That is, the model predicts the token most likely to be present at each timestep. During sequence level training, the model is further trained using a reinforcement learning (RL) scheme designed to optimize reward for the emitted sequence as a whole. Here, the reward is an increase in the BLEU score (Papineni et al., 2002). This results in a slightly different training objective than during token-level training. The authors ultimately found that performing sequence level reinforcement learning after token level training did improve the quality of the model, but not to a tremendous extent.

Other efforts towards OCR for math often employ a similar architecture, such as the CNN-LSTM architecture proposed by Mirkazemy et al. (2022) or the U-net architecture proposed by Ohyama et al. (2019). Very recent efforts have began exploring how the transformer architecture can be applied to the OCR task, such as Zhao et al. (2021) exploring how the transformer can be used to synthesize text in handwritten mathematical expressions. MathPix is purported to be proficient at this task, but is a commercial offering with unclear implementation as is thus not considered in this study (Mathpix).

2.6. Nougat

Nougat is a recently proposed model for neural optical understanding for academic documents (Blecher et al., 2023). Trained on 8.2M pages of academic documents, primarily from arXiv, the model takes as input a PDF page and outputs the identified text, including tables, in a markup language. While the training corpus does contain documents from PubMed Central[3] (PMC), such documents typically present tables as images, so Nougat's pre-processing pipeline is unable to identify the text therein when creating a ground-truth record. Accordingly, although the model performs well in the general academic domain, it struggles when parsing tables from PMC, and often fails to recognize them altogether. Seeing as PMC hosts articles for a significant portion of the scientific community, an approach to filling this gap is of value. The introduction of PEACE provides a high-quality training resource for this type of document.

[3] https://www.ncbi.nlm.nih.gov/pmc/

3. PEACE Dataset

To mitigate the bottleneck of not having a good OCR dataset that contains images (and corresponding labels) of both scientific texts and printed English, PEACE contains 1M images of printed English text, 100k images of numerical artifacts, and 100k images of pseudo-chemical equations, along with their LaTeX labels. Furthermore, for an understanding of real-world performance, the dataset also contains 319 images from real-world scientific documents, again with LaTeX ground-truth strings, to serve as a real-world test set. A summary of the records in PEACE is presented in Table 1. Scripts to generate the dataset are included in our code release so practitioners can generate different versions of the dataset as they see fit, modifying both the corpus that is sampled to generate the records and the formatting applied to the sampled text.

Printed English	
# Total Characters	33M
# Unique Characters	405
Avg # Characters / Record	32.92
# Records	1M
(Pseudo) Chemical Equations	
# Total Characters	7.9M
# Unique Characters	101
Avg # Characters / Record	78.72
# Records	100k
Numeric Records	
# Total Characters	1.9M
# Unique Characters	51
Avg # Characters / Record	18.83
# Records	100k
Real-world Test	
# Total Characters	5,286
# Unique Characters	101
Avg # Characters / Record	16.57
# Records	319

Table 1: Summary statistics of PEACE.

As stated above, the labels accompanying each record are in the form of LaTeX strings. To be rendered in a LaTeX environment, however, the strings need to be surrounded by "$" characters - *i.e.* in *math mode*. In all cases, the python library `matplotlib` (Hunter, 2007) was used to render text as it has the ability to render LaTeX strings, the format in which all of our ground-truth labels are presented.

PEACE exposes an OCR model to a substantial amount of special characters related to chemistry domain. Artifacts such as benzene rings and cubane cube are not included, since they contain many visual artifacts typically not presented in-line with other text and recognizing them does not fall under the umbrella of OCR. While existing datasets

may contain similar components of our dataset (*e.g.*, printed english, latex characters, and chemical equations), PEACE is the first to integrate these components on a record level.

3.1. Printed English Records

To construct our synthetic printed English records, we repurpose the arXiv[4] and PubMed[3] datasets originally proposed for long-document abstractive summarization (Cohan et al., 2018), and supplement them with a crawl of chemRxiv[5] abstracts via the `paperscraper`[6] python package. In total, the aggregate dataset used to create our synthetic records contains 100M words from 31k papers. This dataset is then used to create (rendered text, LaTeX ground-truth) pairs.

Specifically, we randomly sample an academic document from which to sample text, and then select a sample of up to $w = 10$ consecutive words to serve as the ground truth text. It is uncommon to find *exclusively* vanilla printed English in scientific documents, so we perturb the ground truth text to make it more realistic. First, we randomly add a superscript/subscript to a word in the sampled text with probability of $p_1 = 3.75\%$ and $p_2 = 1.25\%$, respectively. Next, LaTeX characters (ϕ, ∞, $\sum$, $\prod$, *etc.*) are randomly inserted to the sequence with probability of $p_3 = 15\%$. For LaTeX characters that require *arguments* (`\bar{}`, `\dot{}`, *etc.*), English characters are sampled 50% of the time, integers the rest. Finally, up to four carriage returns (line breaks) are inserted into sequence with probability $p_4 = 15\%$. The weight associated with inserting i carriage returns is computed as $\frac{1}{i^2}$.

Once the text is sampled, it is rendered using one of eight randomly selected fonts in one of six randomly selected font sizes. This process is completed $n = 1M$ times to assemble the printed English section of PEACE. The python library `matplotlib` (Hunter, 2007) is used to render the text. We would like to note that all of the parameters mentioned above - w, p_1, p_2, p_3, p_4, and n - are exposed to the end user such that they can easily modify the record generation process. Although not exposed as command line parameters, the potential fonts and font sizes can also be easily modified.

3.2. (Pseudo-) Chemical Equation Records

We also create 100k pseudo-chemical equations such that our model will be exposed to this *form* of language during training. These constructed chemical equations are likely to not abide by the laws of chemistry. However, in order to perform OCR on scientific documents, they are useful for exposing our model to this *structure* of text (*i.e.* sequences of 1-2 characters with subscripts interspersed). We do not pursue existing chemical databases to create chemical compounds and equations, because there does not exist a database that renders different chemical compounds/equations in markup languages (*e.g.*, LaTeX) to the best of our knowledge.

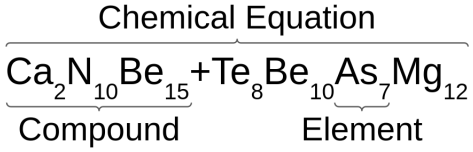


Figure 3: Example pseudo-chemical equation.

To create a pseudo-chemical equation, we first randomly sample a number $n_{compound}$ from 1 to 4 to determine the number of compounds that will be used in the equation. Then, for each compound, we randomly sample a number $n_{elements}$ from 1 to 4 to determine the number of chemical elements in said compound. As elements are randomly sampled to construct the compound, a value $n_{quantity}$ ranging from 1 to 500 is randomly sampled as that elements quantity in the compound. Finally, conjoiners "+", "with", "and", and "plus" were randomly sampled to join the constructed compounds. Again, we randomly sample one of eight fonts and one of six font sizes to render the image to increase the variety of characters the model sees during training. An example of the resulting chemical equation record is presented in Figure 3. Similar to printed English records, the key parameters ($n_{compund}$, $n_{elements}$, and $n_{quantity}$) are exposed to the programmer.

3.3. Numeric Records

Numeric records are constructed following a similar process. First, we randomly sample a number $n_{numerals}$ from 1 to 4 to determine the number of numerals that will be created. For each numeral being created, decimal in the range of 0 to $100,000$ is sampled with probability $p_1 = 0.5$. Otherwise, a LaTeX math symbol ("λ", "β", "Ψ", *etc.*) is randomly chosen. Each numeral is then *joined* using characters such as "$+$", "$\pm$", and "$\neq$". These records may not abide by all mathematical laws and properties, but they serve to expose the model to this type of notation. This process is performed 100k times, and records are rendered using one of eight fonts in one of six font sizes. Again, $n_{numerals}$ and p_1 are exposed to the user as parameters, while the conjoining symbols and set of sampled LaTeX math

[4] https://arxiv.org/
[5] https://chemrxiv.org
[6] https://github.com/PhosphorylatedRabbits/paperscraper

symbols can be easily modified within the code. An example record is presented in Figure 4.

$$15,344 \neq \tau - 30,085.6665 - 44,700$$

Figure 4: Example numerical record.

3.4. Real-World Test Set

To obtain records for our real-world test set, we first collect published scholarly papers relating to polymer pyrolysis and identify 21 tables therein. Then, we transform the PDF pages that contain these 21 tables to images using python library `pdf2image`. Finally, we pass the images through the Multi-Type-TD-TSR model (Fischer et al., 2021) which will section the original image *by table cell*. That is, the model will emit images containing the content of each cell in the table.

Once we have images describing single cells, we identify a group of 319 cell images that are representative of the overall set. This group contains two subgroups: normal cells and special cells. Normal cells contain text that is written using only alphanumeric characters while special cells contain text that cannot be written in *only* alphanumeric characters and require special LaTeX symbols to be rendered.

Note that this real-world test set contains nuances in the data not found in the other synthetic parts of the dataset. To start, some of the original polymer pyrolysis papers appear to be scanned and are not *true* PDF documents. Thus, some cells may have artifacts such as pixelation or smearing. Furthermore, our synthetic records crop the images such that the text *always* appears in the top-left portion of the image. The same cannot be said for images processed and cropped by Multi-Type-TD-TSR, however. We find that in many cases, the text begins in the *middle* of the image, not the top-left. In addition, the Multi-Type-TD-TSR model sometimes crops incorrectly, resulting in a significant amount of white space between text. These nuances make it a valuable representation of real-world data, because the tools that crops images are not expected to be error-free.

Finally, we note that 33 images in the real-world test set are over 700 pixels wide and are too big to be processed by the models trained in this study. We include these images in the released version of PEACE so they would be available to other researchers if desired, but they are not reflected in the performance metrics on the real-world test set presented below.

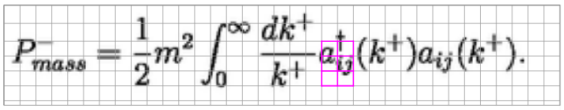
4. Experiments

In this section we describe the experiments that we perform in this study. We first describe our experiments on patch size, identifying nuances of this application perhaps not found in other applications of the ViT, and then our experiments related to multi-domain training for domain-specific applications. We also experiment with different record transformation techniques. We explore the performance of a vision transformer for OCR, dubbed OCR-ViT, as well as Pix2Tex, an OCR-ViT model with a ResNet encoder in place of patch-projection (Section 2.3). Specifically, the OCR-ViT model consists of a bidirectional encoder to process the source image and an autoregressive decoder to synthesize the text.

4.1. Effect of Patch Size

As mentioned above, we are interested in exploring the impact of the patch size parameter on the resulting performance of a transformer-based OCR model. To this end, we explore OCR-ViT models trained with different patch sizes: $10*10$ and $16*16$. Given images of dimensions $160 * 600$, this translates to an effective sequence length of 900 and 375, respectively. To visualize the impact of the patch size in our application, consider Figure 5, where we compare how different patch sizes manifest on the same source image.

a)



b)

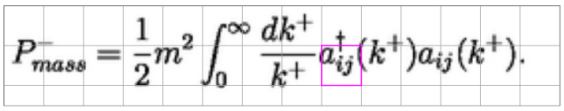


Figure 5: a) Example of how a patch size of 8 manifests on an example image. b) Example of patch size 20 on the same image.

In Figure 5, we see that there is not only a tradeoff between patch size and effective sequence length, but also between the patch size and the *amount of text* contained in each patch. For example, consider the patches highlighted in pink, which are the ones required to (mostly) describe the portion of the image containing the text "a_{ij}". When a patch size of 20 is used, all three characters must be described using a single patch embedding. With a patch size of 8, however, six different patches are used to convey the same information.

Recall from section 2.2 that each patch is presented to the model as a D_M-dimensional embedding, where D_M is the internal dimension of the model. In other words, the content of each patch must be described using D_M numbers. When using a $20 * 20$ patch, then, the text a_{ij} must be described using D_M numbers. For a $8 * 8$ patch, however, $6 \times D_M$ numbers can be used to describe the same text - in some sense, lowering the burden

placed on the model. Of course, this comes with a tradeoff of higher computational overhead.

We hypothesize that the impact of patch size is more pronounced in OCR applications than other applications of the ViT. In our application, the precise content of each patch is of utmost importance - the model must be able to differentiate between an "e" and "c" or "i" and "j" for example. In other ViT applications, however, the precise contents of each patch may not be as important - describing the predominant shape and/or color of the content therein, for example. For this reason, we believe exploring the efficacy of a smaller patch size is important.

4.2. Multi-Domain Training

In addition to exploring patch size, we also explore the impact of adding multi-domain training data for domain-specific inference when sufficient domain-specific training data is available. That is, we explore whether training our models on the joint of im2latex-100k *and* PEACE yields better performance on the test set of im2latex-100k (or PEACE) than training on im2latex-100k alone. This kind of experiment can reveal the utility of PEACE.

4.3. Record Transformations

The synthetic portion of PEACE contains **only** high-resolution records, while the real-world test set contains records with various artifacts - pixelation, smudging, *etc*. We also observe that some records taken from particularly old papers tend to have a very dark, thick font which models in some of our preliminary experiments had difficulty processing. Furthermore, we notice that while the Multi-Type-TD-TSR model does a good job of extracting table cells, it often leaves a non-negligible amount of white space around the textual content of each cell.

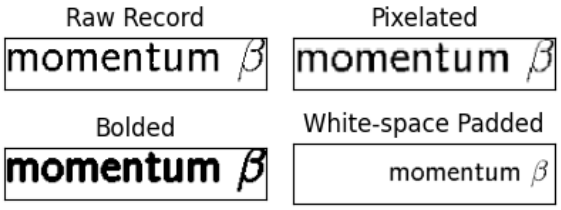


Figure 6: Example transformations.

To address this, we propose three transformations for application on training records: 1) *pixelation*, 2) *bolding*, and 3) *random white-space padding*. An example of these transforms are presented in Figure 6. The *pixelation* filter first compresses the model to a smaller dimension before *expanding* it back to the original dimension, interpolation missing pixels which introduces a small amount of noise. At a high level, the *bolding* filter operates on a pixel-wise basis on a binary, black and white image. If a white pixel has $\geq n$ "hot" pixels in a *n-pixel* radius (horizontally, vertically, diagonally), the white pixel is converted to a black pixel. The *random white-space padding* is very straightforward, adding a random amount of white-space along either side of the image.

5. Results

5.1. Metrics

The reported performance metrics are BLEU-4 score, edit distance, and exact match percentage. BLEU score, a precision-based metric, evaluates the similarity between two pieces of text by counting the number of N-grams in a generated sequence that are present in the ground-truth (Papineni et al., 2002). BLEU-4 analyzes the precision of uni-, bi-, tri-, and quad-grams in the generated sequence with respect to the ground-truth, giving equal weighting to each. Values range from 0 to 100 where higher is better.

Levenshtein distance is the number of single-character edits that must be made to the strings in order for them to match (Levenshtein et al., 1966). Following common practice in the domain (Wang and Liu, 2020, 2021), we present Edit Distance (Edit) between a set of N reference strings R and parallel set of N hypothesis strings H as described in Equation 1. Values range from 0 to 100 with higher values indicating a stronger model.

$$Edit(R,H) = 1 - \frac{TotalLevenshtein(R,H)}{TotalLength(R,H)} \quad (1)$$

where

$$TotalLevenshtein(R,H) = \sum_{i=1}^{N} LevenshteinDistance(r_i, h_i)$$

$$TotalLength(R,H) = \sum_{i=1}^{N} max(len(r_i), len(h_i))$$

The exact match (EM) describes the percent of generated hypotheses that match their corresponding ground-truth string *exactly*. EM essentially describes the record-level accuracy. As with the other metrics, values range from 0 to 100 with higher values indicating a stronger model.

5.2. Effect of Patch Size

The performance of training our OCR-ViT model with two different patch sizes, 10 and 16 are included in Table 2. We see that a lower patch size yields stronger performance in all cases, but

Model	Patch Size	im2latex			PEACE (Synthetic)			PEACE (Real-World)		
		BLEU	Edit	EM	BLEU	Edit	EM	BLEU	Edit	EM
MI2LS-MLE	-	89.08	91.09	79.39	-	-	-	-	-	-
MI2LS-RL	-	**90.28**	**92.28**	**82.33**	-	-	-	-	-	-
OCR-ViT	10	**84.53**	**87.45**	**36.92**	**99.53**	**99.64**	**98.31**	**81.24**	**86.09**	**51.05**
OCR-ViT	16	72.11	77.11	19.48	98.76	99.20	94.85	72.52	77.96	41.96
OCR-ViT (w/o bolding, pixelation, padding)	16	65.59	71.57	20.91	98.85	99.25	99.13	55.27	58.73	12.24
OCR-ViT (im2latex **only**)	16	31.11	39.06	0.83	0.42	2.45	0.01	0.68	3.38	0.00
OCR-ViT (PEACE **only**)	16	1.60	13.42	0.00	98.89	99.28	95.09	67.79	74.60	37.06
Pix2Tex (im2latex **only**)	8	**90.24**	**91.78**	**39.24**	0.97	4.03	0.03	2.90	8.84	0.35
Pix2Tex (**im2latex + 10% PEACE**, w/o bolding, pixelation, padding)	8	87.95	89.44	33.24	**99.19**	**99.38**	**94.93**	**68.85**	**74.74**	**31.82**

Table 2: Performance of various model architectures trained using different patch sizes on im2latex, PEACE, and PEACE Real-World test sets. For all three of our performance metrics, a higher value indicates a better model. Unless stated otherwise, models were trained on *both* im2latex and PEACE.

these performance improvements are most pronounced on im2latex and the PEACE real-world test set. Considering the discussion in Section 4.1, this makes sense. The records in im2latex and PEACE real-world are more *complex* than vanilla printed English, which comprises the majority of the synthetic PEACE test set. As a consequence, each *patch* is burdened with expressing relatively *more* information in these settings, and lowering the patch size lowers thus burden. This advantage of lower patch size we find on OCR-ViT facilitates us to pursue a patch size of 8 on Pix2Tex.

5.3. Multi-Domain Training

In this section, we present results from our experiments exploring the impact of multi-domain training for domain-specific inference. That is, we explore the effect of training on the union of im2latex-100k and PEACE versus only on im2latex-100k. Performance is evaluated on three test sets: im2latex-100k, PEACE synthetic test set, and PEACE real-world test set.

For the OCR-ViT models, performance on PEACE synthetic test set is relatively unchanged as long as PEACE is included in the training corpus - in isolation *or* with im2latex-100k. Performance on im2latex-100k and PEACE real-world test set is greatly improved when a model is trained on both corpora than either one in isolation. We see that multi-domain training yields an average improvement of 825.39% on im2latex (primarily in EM) and 8.23% on PEACE real-world.

For Pix2Tex, comparing the last two rows of Table 2, we see that adding PEACE into training yields substantial performance boost on PEACE synthetic and real-world test set. Scores on im2latex-100k are slightly lower than training on im2latex-100k alone, but it is clear that multi-domain training yields the best overall performance on Pix2Tex model. This intuitively makes sense, as our target domain is a hybrid of printed English and scientific text. The advantage of multi-domain training reconfirms the value of proposing PEACE, since constructing a hybrid dataset like PEACE is not straightforward using existing resources due to reasons such as inconsistency of datasets across various domains. Note that we use 10% of PEACE without bolding, pixelation, and padding to reduce training cost, as it would take roughly three months to train on the entirety of PEACE using the script provided by Pix2Tex.

5.4. Significance of Real-World Test Set

In Table 2, it is clear to see that all the models yield significantly worse performance when they are tested on PEACE real-world test set compared to when they are tested on other data. Since this real-world test set is comprised of artifacts from *real* scientific documents, we conclude that it helps to show the actual capability of each OCR model on real-world chemistry scholarly papers. This novel test set clearly reveals the weakness of each selected model, so its value is demonstrated.

5.5. Impact of Record Transformations

In comparing rows 4 and 5 of Table 2, we see that our proposed bolding, pixelation, and padding transformations yield performance improvements on the im2latex and PEACE real-world test sets,

with most pronounced improvements on the PEACE real-world test set. The transformations yield an *average* improvement of 3.61% and 102.25% on im2latex test set and PEACE real-world test set, respectively, in BLEU, edit distance, and exact match. Performance on PEACE synthetic test set is largely unchanged with and without the transformations.

Intuitively, these results make sense. The impact of the transformations is most pronounced on the type of records they were designed to mimic, but they increase the model's performance in other applications also. These transformations lead to a slight performance decrease on the PEACE synthetic test set, but an overall stronger model.

6. Conclusion

In this paper we introduce the PEACE dataset for OCR, containing more than 1M (rendered text, LaTeX ground-truth) pairs of printed English and chemical equation text. The dataset contains three subsections: printed English, pseudo chemical equations, and images of text extracted from real-world scientific documents. This dataset helps bridge the gap between OCR models and datasets designed for either vanilla printed English or scientific text (*e.g.*, math and physics formulae), but not both.

Additionally, we survey a variety of architectures when applied to the PEACE dataset. We find that a traditional ViT with small patch size $(10*10)$, trained in a multi-domain setting using our proposed pixelation, bolding, and padding transformations yields the best *overall* performance. However, a Pix2Tex model (*i.e.* ViT + CNN encoder) yields competitive performance when trained on only 10% of PEACE *without* our proposed transformations suggesting a promising path forward for future OCR models.

7. Acknowledgements

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office Award Number DE-EE0007897 awarded to the REMADE Institute, a division of Sustainable Manufacturing Innovation Alliance Corp.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

8. Bibliographical References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Lukas Blecher. 2020. GitHub - lukas-blecher/LaTeX-OCR: pix2tex: Using a ViT to convert images of equations into LaTeX code. — github.com. https://github.com/lukas-blecher/LaTeX-OCR. [Accessed 06-Oct-2022].

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, and Alexander M Rush. 2017. Image-to-markup generation with coarse-to-fine attention. In *International Conference on Machine Learning*, pages 980–989. PMLR.

Yuntian Deng, David Rosenberg, and Gideon Mann. 2019. Challenges in end-to-end neural scientific table recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 894–901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words:

Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Pascal Fischer, Alen Smajic, Alexander Mehler, and Giuseppe Abrami. 2021. Multi-type-td-tsr – extracting tables from document images using a multi-stage pipeline for table detection and table structure recognition: from ocr to structured table representations.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Martin Kišš, Michal Hradiš, and Oldřich Kodym. 2019. Brno mobile ocr dataset. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1352–1357. IEEE.

Martin Kišš, Michal Hradiš, and Oldřich Kodym. 2019. Brno mobile ocr dataset. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1352–1357.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Mathpix. Mathpix OCR — mathpix.com. https://mathpix.com/ocr. [Accessed 06-Oct-2022].

Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin. 2020. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, 8:142642–142668.

Abolfazl Mirkazemy, Peyman Adibi, Seyed Mohhamad Saied Ehsani, Alireza Darvishy, and Hans-Peter Hutter. 2022. Mathematical expression recognition using a new deep neural model. *Available at SSRN 4245142*.

Wataru Ohyama, Masakazu Suzuki, and Seiichi Uchida. 2019. Detecting mathematical expressions in scientific document images using a u-net trained on a diverse dataset. *IEEE Access*, 7:144030–144042.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Shruti Patil, Vijayakumar Varadarajan, Supriya Mahadevkar, Rohan Athawade, Lakhan Maheshwari, Shrushti Kumbhare, Yash Garg, Deepak Dharrao, Pooja Kamat, and Ketan Kotecha. 2022. Enhancing optical character recognition on images with mixed text using semantic segmentation. *Journal of Sensor and Actuator Networks*, 11(4).

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

R. Smith. 2007. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. 2021. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272.

Joel CM Than, Pun Liang Thon, Omar Mohd Rijal, Rosminah M Kassim, Ashari Yunus, Norliza M Noor, and Patrick Then. 2021. Preliminary study on patch sizes in vision transformers (vit) for covid-19 and diseased lungs classification. In *2021 IEEE National Biomedical Engineering Conference (NBEC)*, pages 146–150. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. 2020. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM conference on health, inference, and learning*, pages 222–235.

Zelun Wang and Jyh-Charn Liu. 2020. Pdf2latex: A deep learning system to convert mathematical documents from pdf to latex. In *Proceedings of the ACM Symposium on Document Engineering 2020*, pages 1–10.

Zelun Wang and Jyh-Charn Liu. 2021. Translating math formula images to latex sequences using deep neural networks with sequence-level training. *International Journal on Document Analysis and Recognition (IJDAR)*, 24(1):63–75.

Yibin Ye, Shenggao Zhu, Jing Wang, Qi Du, Yezhang Yang, Dandan Tu, Lanjun Wang, and Jiebo Luo. 2018. A unified scheme of text localization and structured data extraction for joint

ocr and data mining. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2373–2382. IEEE.

Nan Zhang, Shomir Wilson, and Prasenjit Mitra. 2022. STAPI: An automatic scraper for extracting iterative title-text structure from web documents. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3461–3470, Marseille, France. European Language Resources Association.

Wenqi Zhao, Liangcai Gao, Zuoyu Yan, Shuai Peng, Lin Du, and Ziyin Zhang. 2021. Handwritten mathematical expression recognition with bidirectionally trained transformer. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16*, pages 570–584. Springer.

Ilia Zharikov, Philipp Nikitin, Ilia Vasiliev, and Vladimir Dokholyan. 2020. Ddi-100: dataset for text detection and recognition. In *Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control*, pages 1–5.