

# New Paths in Document Data Augmentation Using Templates and Language Models

Lucas Wojcik<sup>1</sup><sup>a</sup>, Luiz Coelho<sup>2</sup><sup>b</sup>, Roger Granada<sup>2</sup><sup>c</sup> and David Menotti<sup>1</sup><sup>d</sup>

<sup>1</sup>*Department of Informatics, Federal University of Paraná, Curitiba, Brazil*

<sup>2</sup>*unico idTech, Brazil*

{lmlwojcik, menotti}@inf.ufpr.br; {luiz.coelho, roger.granada}@unico.io

**Keywords:** Document Recognition, Data Augmentation, Natural Language Processing.

**Abstract:** Document Recognition has been tackled with a state of the art (SOTA) mostly composed of multi-modal transformers. Usually, these are trained in an unsupervised pre-training phase followed by a supervised fine-tuning phase where real-world tasks are solved, meaning both model and training procedures are borrowed from NLP research. However, there is a lack of available data with rich annotations for some of these downstream tasks, balanced by the copious amounts of pre-training data available. We can also solve this problem through data augmentation. We present two novel data augmentation methods for documents, each one used in different scopes. The first is based on simple structured graph objects that encode a document's layout, called templates, used to augment the EPHOIE and NBID datasets. The other one uses a Large Language Model (LLM) to provide alternative versions of the document's texts, used to augment the FUNSD dataset. These methods create instances by augmenting layout and text together (imageless), and so we use LiLT, a model that deals only with text and layout for validation. We show that our augmentation procedure significantly improves the model's baseline, opening up many possibilities for future research.

## 1 INTRODUCTION


Document recognition has come a long way since LayoutLM (Xu et al., 2019), which cemented a tradition of using multi-modal transformers in this field ever since its release. Since innovations in transformer pre-training recipes and attention modeling have shown to have the biggest impact in pushing the state of the art (SOTA) forward, most of the recent research focuses on these things to tackle new challenges. As a result, research in document-based data augmentation has become scarce over the last few years.


However, there is a lot to be gained from data augmentation, as many domains, including ones of commercial interest such as official document parsing, suffer from a lack of high-quality data and annotations. The pre-training phase with vast amounts of data that are a staple in document recognition today yields models with powerful few-shot capabilities,


which enhances performance in scenarios with low data availability. But it is also possible to further boost performance by expanding the datasets with data augmentation.


For instance, we tackle EPHOIE (Wang et al., 2021), a dataset composed of scanned examination paper headers from various Chinese schools. This document domain is particularly tricky to deal with, since these documents may contain sensitive information from real people (the authors of EPHOIE had to erase and re-synthesize some fields such as names and schools to make the dataset public). This makes it hard and, at times, impossible to gather and annotate more document instances, making data augmentation an even more relevant tool to use. This is also an issue with NBID (Wojcik et al., 2023), a synthetic ID card dataset.

We also tackle FUNSD (Guillaume Jaume, 2019), a subset of IIT-CDIP (Soboroff, 2022). IIT-CDIP is a dataset composed of noisy scanned documents from lawsuits against the tobacco industry in the nineties. These documents were made public as a result of the lawsuit's settlement, but still represent another sensitive document domain, namely the legal document one. These are also documents that contain confiden-

<sup>a</sup> <https://orcid.org/0009-0006-3825-5959>

<sup>b</sup> <https://orcid.org/0009-0004-8330-5199>

<sup>c</sup> <https://orcid.org/0000-0001-5908-9247>

<sup>d</sup> <https://orcid.org/0000-0003-2430-2030>

tial information, and so are difficult to acquire and annotate.

With these issues in mind, the present work pioneers two novel data augmentation methods aimed at real-world problems (fine-tuning stages), where the annotations are usually more scarce. Our approach includes layout and text augmentation, borrowing techniques from pure NLP research, and experimenting with the latent knowledge from the documents themselves.

The first method is called the *LLM* method, where we use an LLM to produce new instances of the same document by rewriting the texts of every entity. This is in line with recent advancements from NLP (Guo et al., 2023; Ye et al., 2024) where this idea was shown to improve performance in downstream tasks consistently.

The second method is called the *template* method, where we reduce each document to a fully connected digraph where each vertex corresponds to an entity and the edges represent directions between entities. Augmentation is performed by producing a template repository from the available documents and then creating new documents by sampling one template at a time and filling the vertices with new text.

Our methods are crafted for complex (LLM) and simple (template) document domains. We use FUNSD, EPHOIE and NBID for validation of these methods, as these datasets serve as examples for the domains tackled by them. FUNSD is augmented with the LLM approach and the other two are augmented with the template approach. Both methods are crafted for *imageless learning*. The reason for this is to develop faster and more versatile augmentation methods, as well as being easier to implement.

We use LiLT (Wang et al., 2022), fine-tuning it in downstream tasks from the FUNSD (Guillaume Jaume, 2019) and EPHOIE (Wang et al., 2021) datasets, augmented using the LLM and template strategies respectively. We show that our methods improve the model’s performance significantly across every training scenario. Furthermore, our augmented datasets will be made publicly available.

The remainder of this work is structured as follows. Section 2 presents an overview of the document recognition SOTA and some document augmentation techniques, as well as situating our contributions to them. Section 3 presents our two augmentation methods, based on templates and LLMs. We also detail the datasets used and produced by and with each method. Section 4 details our experiments and results, as well as a discussion of the results. Finally, Section 5 presents our conclusions and discusses the directions for future research.

## 2 RELATED WORK

Recent advances in document recognition are often an incorporation of NLP techniques, to the point that this field can be conceptualized as a subfield of NLP. This is seen in the way that the current SOTA for documents is a generalized, multi-modal version of the NLP SOTA. The first instance of this is LayoutLM (Xu et al., 2019), a model that uses a BERT (Devlin et al., 2019) backbone as a baseline and is trained with a very similar recipe of unsupervised pre-training followed by supervised fine-tuning. The translation of the vanilla NLP modeling into a document modeling is by building a multi-modal embedding that encodes layout and vision features (apart from the text) and adapting the pre-training tasks accordingly. In this way, document recognition becomes a simple NLP task with extra dimensions.

This modeling is omnipresent in the current document SOTA, shown in recent models such as LayoutLMv3 (Huang et al., 2022), GraphDoc (Zhang et al., 2022) and ERNIE-Layout (Pend et al., 2022). GraphDoc also incorporates the graph-like attention modeling from StarTransformer (Guo et al., 2019), another NLP approach, while ERNIE-Layout borrows the ERNIE architecture and attention modeling from the corresponding vanilla NLP model (Zhang et al., 2019).

Although most of this SOTA uses a multi-modal learning comprised of vision, text, and layout, recent research has also produced competitive bi-modal models that exclude the requirement for images to be used, even in visual document recognition scenarios such as the ones tackled here. LiLT (Wang et al., 2022) is composed of two independent transformer architectures that connect through the attention mechanism, one for text and one for layout. LayoutMask (Tu et al., 2023) proposes novel pre-training tasks for better cross-modal learning.

There is further influence from NLP in document augmentation. (Márk and Orosz, 2021) presents an extensive survey of augmentation methods for text augmentation. The paper discusses techniques such as synonym replacement, random deletions, and round-trip translations and their applicability in the legal document scenario. It also discusses LLMs (GPT and GPT-2 (Radford et al., 2018; Radford et al., 2019)), to the conclusion that these methods cannot be used as they can’t protect some keywords from the original texts. Finally, the authors do not perform any experiment, limiting the scope of the paper to a discussion.

Practical augmentation methods for documents can be largely summed up in two categories: *intrinsic*

and *extrinsic* methods. Intrinsic methods work by creating a model that learns the semantics of a document instance in order to create new instances. Extrinsic methods work by using sets of data (which may be sets of images, document textures, text samples, etc.) to incrementally build documents in a more handcrafted way. Some works, such as DocBank (Li et al., 2020), present strategies for constructing datasets via new annotation methods. We do not consider these as data augmentation methods, as the datasets are being created and not expanded.

The most prominent example of intrinsic methods is with GANs. Examples of this include DocSynth (Biswas et al., 2021), a GAN trained in the large PubLayNet (Zhong et al., 2019) dataset, comprised of over three hundred thousand pages from scientific papers. The model is trained to recognize the possible layouts and then used to create new ones. Another example is in (Pondenkandath et al., 2019), where the CycleGAN (Zhu et al., 2017) and VGG-19 (Simonyan and Zisserman, 2014) models are used to create aged versions of historical documents.

Another example of an intrinsic method is in (Raman et al., 2021), an annotation-free approach for layout recognition based in a Bayesian network. Documents are defined as a set of primitive elements (paragraphs, lists, titles) and construct documents according to a set of rules through this network. The result is fed into an image manipulation routine to create an augmented document image.

An example of an extrinsic method is SynthDoG (Kim et al., 2022). In the same paper, the authors propose Donut, a Transformer that doesn't use the document's texts as part of the input. SynthDoG works by sampling images from ImageNet (Krizhevsky et al., 2012) to serve as backgrounds, onto which document textures are projected, the text being pasted on top of them. The texts are from many languages and taken from Wikipedia. Such a dataset allows for Donut to be generalized without the need for multilingual text understanding.

NBID (Wojcik et al., 2023) is a synthetic dataset of Brazilian ID cards created from a simple process of inpainting the sensitive information in real document images with a GAN and pasting synthetic text on the anonymized image. It is an example of a simple domain of documents that is hard to deal with due to the sensitive nature of the data. It is further detailed in Section 3.3

An advantage of the implicit methods is the possibility of creating vast amounts of data with very varied layouts, as seen with DocSynth. A drawback is the necessity of lots of annotated data for training the model, meaning they are not well suited for domains

with a general lack of data and/or annotations, as usually is the case with fine-tuning tasks. Implicit methods have the inverse pros and cons: larger amounts of data can be created from smaller datasets (combining five background images, document textures, and text samples, a total of  $5^3$  instances can be created), but these instances may be very similar to each other.

Both of our proposed methods are categorized as extrinsic. The template approach works with basic building blocks in the form of templates and text dictionaries. For the LLM approach, the building blocks correspond to the available documents themselves, as well as their texts. Although we use an off-the-shelf pre-trained model, it is not fine-tuned to learn the document structure, hence why it cannot be categorized as intrinsic.

As discussed, extrinsic methods suffer from a lack of variability. The LLM approach attempts to overcome this limitation by leveraging the knowledge contained in LLMs. Rewriting the text from a document gives the dataset a larger variance in terms of vocabulary, syntax and intonation. Since LLMs can understand a wide variety of scenarios, and rewriting is an easy problem of them, this allows us to augment very complex documents, such as the ones contained in FUNSD (Guillaume Jaume, 2019), which contain highly specialized text samples.

The template approach is destined for domains of simpler documents and layouts, and as such the datasets themselves usually lack in variety. As such, this method aims to reduce the amount of data needed to create a fully representative dataset. For instance, for a given set of official documents such as passports and ID cards, the possible layouts are strictly defined by law, and these can be used to build the template repository for augmentation. A single template can be used to create dozens of different documents through the creation of synthetic data to fill said template, in a process similar to the one presented in NBID (Wojcik et al., 2023).

Therefore, our contributions include the introduction of these two novel augmentation techniques that manage to overcome some of the challenges faced by the literature in data augmentation. These techniques are versatile and can be used for many different scenarios. This is the main advantage of our proposed methods, since those found in the literature are either unfit for our small fine-tuning scenario (intrinsic methods) or handcrafted for different document domains and hardly transferable for the datasets used. For instance, SynthDoG relies on gathering texts from wikis and pasting them onto textures that are projected onto background images. Both the technique of using texts from random domains and the projection

of the document don't make sense for EPHOIE and FUNSD, that have fixed domains and are correctly scanned. This makes it hard to apply these techniques for a direct comparison. Lastly, we will make our augmented datasets publicly available.

### 3 METHODOLOGY

This section presents and details our novel augmentation techniques. These are the LLM and template methods, described in Section 3.1 and 3.2 respectively. Each approach is tailored for a specific scenario: the first for documents with complex texts and layouts, the second for simple texts and layouts. As such, we choose FUNSD (Guillaume Jaume, 2019) and EPHOIE (Wang et al., 2021) for augmentation respectively for each technique, as these datasets follow their domains of applicability. The LLM approach works by using LLMs to rewrite the texts of a given document, while the template approach uses the graph structure of a document to create a template that can be used for augmentation.

While both of our methods work by using the same document structure to paste new text instances, the template method stands out by being more general in its text substitution approach. In the LLM case, we simply rewrite the text of the same document a few times, always using the original text as a guide in order to maintain the overall semantics of the document. This is required by the domain this approach is tailored for, where the documents feature complex inter-entity relationships that could be broken if the substitution approach is careless. This is not the case for the domain tackled by the template approach, where the entities are far simpler, being composed of names of people, schools, grades and subjects (for the EPHOIE dataset), and as such a dictionary swap can be used with no problems for the document's coherency.

Both of our approaches follow a line of producing *imageless* augmentations. The reason for this is to develop faster and more versatile augmentation methods, as well as being easier to implement. Augmenting images, for the document scenario, poses itself as a rather difficult problem, requiring methods such as GANs or other inpainting methods, or gathering more image instances to expand the dataset. In the first case, there is often a semantic gap between the real and synthetic images that may produce unwanted bias for the model, while in the second case gathering more instances might not be possible, for instance when dealing with sensitive document domains such as ID cards or lawsuits (the case for FUNSD and EPHOIE, where the documents were only pub-

licized at the end of the processes and where some information had to be erased and re-synthesized, respectively). Also, although most of the recent Document Recognition models use image features for learning, imageless models such as LayoutMask (Tu et al., 2023), which only uses text and layout cues, remains competitive with the current SOTA at the time of writing, being ranked #1 at the entity labeling task in FUNSD.

As such, our work defines the document augmentation process as being a task relating primarily to textual augmentation. This aligns with the established SOTA for document recognition, which is closely related to NLP methods, as previously discussed. For this work, we define a document as a list of entities, where an entity corresponds to a semantic object within the document. Every entity contains a set of attributes, with required info being the entity's text, its coordinates in the document image, and its class. Some datasets may define other attributes. For instance, FUNSD includes the "linking" attribute, which is the key-value relationship between different entities inside the same document. These relationships may exist between entities belonging to the class of header (key) and question (value), and question (key) and answer (value).

#### 3.1 LLM Augmentation

For FUNSD (Guillaume Jaume, 2019), which features complex templates and texts (the reason why our template augmentation cannot be used for this dataset), we experiment with a technique inspired by recent advances in NLP research dealing with data augmentation (Ye et al., 2024; Guo et al., 2023). The mentioned papers have found that textual augmentation through rewriting, that is, using a model to produce alternative versions of the same text, improved the end result for the Text Recognition model consistently for all models and tasks explored in both papers. Hence, we bring this idea to Document Recognition by using an LLM to provide alternative versions for the text of every entity in a given document. Our approach for FUNSD is illustrated in Figure 1.

Since FUNSD has a wide variety of text types, we found no one-size-fits-all augmentation technique that can be used. We separated each entity into four classes according to the text type. These classes correspond to complex sentences, simple questions (entities with texts such as "name:", "R&D" and such), simple answers (names, dates, measures, and such), and none (such as empty strings and one or two character strings). For each one of these classes, a different augmentation process was used. We highlight

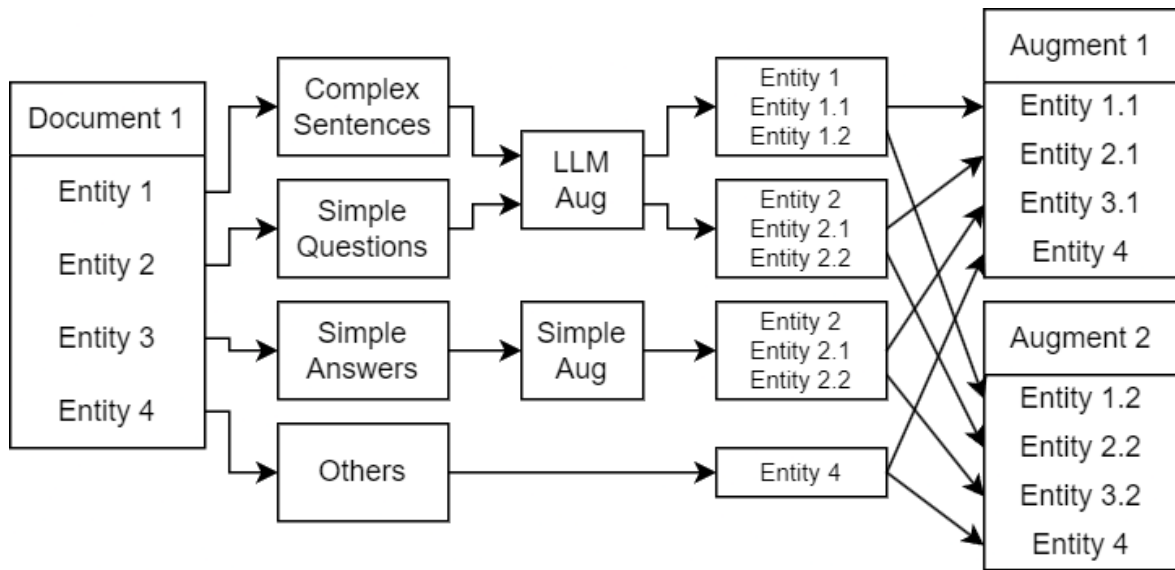


Figure 1: Diagram representing the LLM augmentation process.

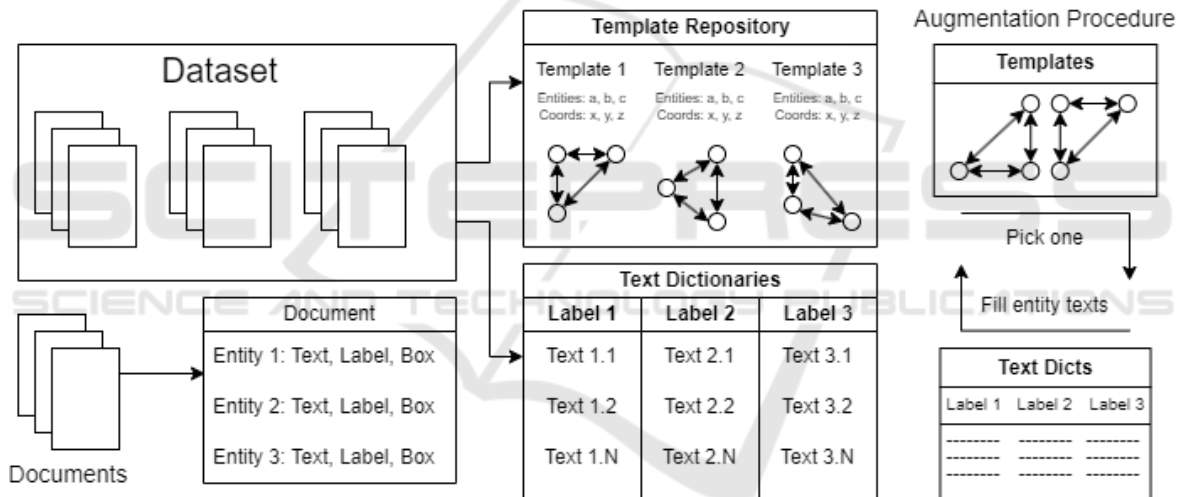


Figure 2: Diagram representing the template augmentation process.

that the labels defined here do not correspond to the classes defined by the dataset itself.

When considering the complex sentences, we ask a pre-trained LLM to rewrite the text up to five times through prompt engineering. The model is encouraged to modify the vocabulary and syntax used in the original texts. This method provides the dataset with a more significant textual variety according to these two areas. When considering the simple questions, the LLM is asked to provide a list of synonyms. The amount of synonyms varies according to each entity, due to both vocabulary and context limitations.

For both cases, we use an off-the-shelf LLM model with no fine-tuning, with the augmentations being done only through prompt engineering. We cu-

rate the output of the LLM manually in order to both remove the extra text padding that the model adds to each generation<sup>1</sup> and to ensure that the generations make sense within the context of the original text.

Simple answers are augmented through a few simple techniques. Names are replaced using a dictionary. Names with initials are expanded using random names from the dictionary with the same initial, while full names are retracted into the initials. The names also change format: from "Surname, Name" to "Name Surname", etc. New dates are generated by changing the date format. For example: MM-DD-

<sup>1</sup>Some examples are: "Certainly! Here's a list of synonyms:" and "Thank you for asking! Here are some synonyms:".

YY can be changed to "Month DD, YYYY" and vice versa. Measures and numbers are augmented by generating OCR noise on a few digits: a small amount of digits is randomly selected and replaced by other random digits. The rest of the entities, belonging to the label of none, are left untouched.

Since we vary the number of augmentations per entity according to both the LLM output and the type of the text in the simple answer scenario, it becomes necessary to choose how we create the new documents, given the new texts available. To solve this problem, we order the new text variants for every entity and choose the next unused text variant when creating the next augmented document, as seen in Figure 3.1: the first augmented document uses the first augment of each entity, the second augmented document uses the second augment of each entity, etc. Once all entity augments are used up, the next augmented document chooses one text from the entire list (included the original one) at random, uniformly. Also, since the text may vary in length, we paste each entity's augmented text into the document (sampling the font size and line limit from the original entity's bounding box) to generate a new, more precise bounding box.

The text rewrite technique, as we call it, is the first attempt in the literature to use the few-shot power of LLMs for data augmentation in tasks involving the FUNSD dataset (to the best of our knowledge). Apart from aiding our model in reaching higher levels of accuracy, it also opens up new paths for future research. With recent developments in image generation (Dhariwal and Nichol, 2021), this technique can be used to recreate document images as well. As such, it could be used for improving the performance of tri-modal (text, layout, and image) document models (Huang et al., 2022) as well.

This technique stands apart from the template approach the text substitution method here is focused on maintaining the scope of the document itself. Rather than using the entity structure to fill in random texts, we keep the same syntax and semantics, only rewriting the existing texts. Values, grammatical structures and names are changed, but the root document stays the same. This technique is also set apart from other techniques in the literature by the fact that it deals with visual documents directly.

### 3.2 Template Augmentation

Some types of documents are predefined by a reduced set of *templates*, such that each document corresponds to a specific arrangement of entities within the image. For instance, a given region may have a set

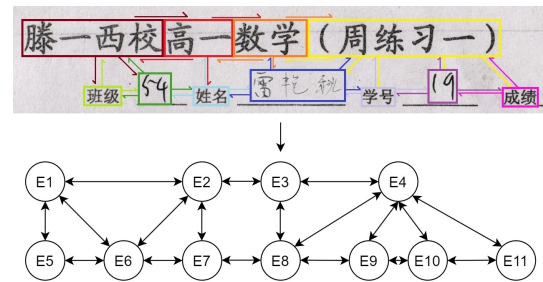


Figure 3: Template example from an EPHOIE instance.

of accepted identity card formats that are predefined by law. In this case, the templates correspond to the document backgrounds where the information of the holder will be displayed.

For our purposes, we use a simpler definition of template that does not use the image. We define a template as a fully connected directed graph where each entity is a vertex, having the entity type and the  $x$  and  $y$  coordinates as attributes of the corresponding vertex. Each edge connecting two vertices has an attribute corresponding to one of eight possible directions (vertical, horizontal, and diagonal), indicating the relative position between the two entities connected by the edge.

The presented definition will be used for our augmentation approach, and is enough to define the ID card domain presented previously. Figure 3 illustrates our definition. We highlight that the graph is complete, but we suppress most of the edges in the illustrated document for the sake of readability.

Given all of the possible templates at hand, it is possible to generate an arbitrary number of new document instances by picking a template at random, filling the entity vertices with appropriate texts and generating new bounding boxes by writing into a blank image. However, we need a way to generate new texts for the entities in each template. There are many ways to approach this: we could use a fine-tuned LLM to generate new texts given the entity's class name (which should be semantically meaningful for its type of texts - an entity of the class 'name' should have proper names as its transcripts) or create a dictionary of possible texts for each entity class.

For the latter, there is still the question of how to build such dictionaries. We can use the document domain in question to narrow our scope, for example in identity cards of a given country we can find all of the authorities that can issue them to build the "issuing organization" dictionary, and we can search for lists of common names in that country for the "name" dictionary, and so on. For NBID, this is the method used in its original paper, and also the one we use for our augmentations, since it is publicly available on the

dataset’s official github. A simpler approach, and the one we use for EPHOIE, is to gather all of the texts that appear in the available dataset samples.

In this way, Figure 2 presents our approach for the proposed augmentation pipeline. It can also be summed up in the following steps:

1. Sample the dataset in order to create a repository of texts. For each entity class, create a list of all the texts of every entity corresponding to the same label across every document present in the dataset.
2. Extract a repository of templates. Extract the template of every document in the dataset from their annotations and, if there are any, identical templates (same vertices and directions on every edge) are merged together.
3. Randomly pick a template and fill in the entity texts by randomly sampling from the entity text dictionaries.

Apart from improving the model performance (see Section 4), this strategy is also interesting because it makes it possible to generate a great variety of instances from very few examples, given many templates are contemplated. This applies to other domains such as the aforementioned ID cards. Also, we can further relax our assumption that the dataset has a reduced number of templates overall, so long as we have enough instances to accurately represent the diversity in the given document domain. For EPHOIE, from 1183 training documents we have extracted 1046 unique templates, and our results show that these were well enough to construct meaningful augmented partitions.

### 3.3 Dataset Description

Table 1 presents the number of documents and entities for FUNSD and EPHOIE. The real partition (and what we will subsequently call real documents) is the training set defined by the dataset itself. For EPHOIE, we extracted 1046 unique templates from the set of 1183 real training documents. We incrementally build three sets of synthetic documents using this repository of 1046 templates, adding 1200 new instances each time. We removed 12 synthetic instances because they ended up malformed at the end of the generation process. These failures were due to a few instances that had texts way too long for the templates, which ended up overflowing out of the page.

In the original NBID dataset, the 1000 training documents were generated from 200 root documents, where each real document had its sensitive information erased, and new data was synthesized and pasted on the empty documents. Each real document was

used to generate five instances, totaling the 1000 documents present in the dataset. For NBID, there were 54 unique templates from the 1000 real training images, a much smaller ratio of documents per templates. Considering each five clones of the same root dataset are bound to share the same template, this means 54 unique templates from 200 documents, approximately 3.7 documents per template. This means NBID has far less varied templates than EPHOIE, where this ratio is approximately 1.15.

For FUNSD, we generate up to five augmented versions of every document from the 149 instance training partition, for a total of 745 synthetic generations. Each synthetic partition contains  $N$  augmentations of every real document, where  $N$  is the order of the partition. This means the 1 augment partition is comprised of one generation of each of the 149 real documents, the 2 augments of two generations, and so on.

Table 2 presents the number of entities in the real training set of EPHOIE, and Table 3 presents the entities in the real training set of FUNSD. As we can see, both datasets contain a very unbalanced number of entities between one class and the other. In Table 3 (FUNSD), we also present the number of entities we manually classified according to the type of text they contain, according to our definitions in Section 3.1. Furthermore, we also present how many new texts were generated for each entity, that is, for how many entities we generated one, two, three, or four more new text variations.

The number of augments per entity varies because a few entities don’t have enough meaningful text variations. Some texts consisting only of simple nouns, for example, can be replaced by synonyms, but the list of synonyms is limited by the dictionary and by the context. FUNSD consists of official forms and the style of language must adhere to this context, limiting the acceptable vocabulary. Furthermore, the text types, as we classified them, are also unbalanced among each other, most of them belonging to the type of simple question for which the synonym augmentation procedure was used.

For FUNSD, we use the pre-trained Llama-2-7b-hf (Touvron et al., 2023) for the text augmentation. The main reason for choosing Llama2 is the fact that it is freely available to use, also being faster and more efficient than other models such as GPT-3 (Brown et al., 2020).

The entities for which we could not produce meaningful augments correspond to simple entities of three characters or less (such as tickboxes) as well as entities with no text (where we assume that the OCR mechanism used for the semi-manual annotation procedure described in the FUNSD paper failed)

Table 1: Number of instances in FUNSD, EPHOIE and NBID partitions.

Partition	FUNSD		EPHOIE		NBID	
	Documents	Entities	Documents	Entities	Documents	Entities
Real Train	149	7411	1183	12411	1000	7515
Testing	50	2332	311	3343	110	785
1 Augment	149	7411	1200	12921	10000	103513
2 Augments	298	14822	2400	25850	-	-
3 Augments	447	22233	3588	38656	-	-
4 Augments	696	29644	-	-	-	-
5 Augments	745	37055	-	-	-	-

Table 2: Number of entities in EPHOIE by class.

Entity type	Amount
Other	5679
Exam Number	128
Score	377
Name	2365
Student Number	422
School	1358
Grade	441
Seat Number	184
Class	1625
Subject	376
Candidate Number	467
Test Time	79

or entities to which the LLM failed to produce satisfying augmentations. Examples of the latter include some entities from the “other” label consisting of long codes of letters and numbers, acronyms the model did not recognize, and chemical compounds.

## 4 EXPERIMENTS

In this section, we present the model we use to validate our approach and its training scenarios, as well as the results for each one, both the baseline and our augmented results. We use the publicly available implementation of the model for our fine-tuning, and compare the results to the ones reported by the paper of the baseline model.

### 4.1 Model and Protocols

To validate our approach, we use LiLT (Wang et al., 2022), a bi-modal, dual transformer (Vaswani et al., 2017) model. It features two different transformers with independent weights, each one corresponding to one embedding: text or layout. The models communicate through a specially designed “bi-directional attention complementation mechanism” that replaces the vanilla attention of the original transformer. This is better detailed in the original LiLT paper.

This design allows LiLT to be coupled with different transformer models. This is an important advantage when dealing with multilingual scenarios, allowing a base layout-only pre-trained transformer (LiLT base) to be coupled with different models from the literature (even trained in different languages). LiLT is fine-tuned on both FUNSD and XFUND (Xu et al., 2022), which contains seven different languages. The base LiLT model is coupled with an English RoBERTa (Liu et al., 2019) for FUNSD and InfoXLM (Chi et al., 2021) for XFUND. The authors of LiLT also evaluate EPHOIE using both InfoXLM and a Chinese RoBERTa model (Cui et al., 2020). No experiments are performed with NBID, which was released more recently. We fine-tune the pre-trained LiLT on the base NBID for a baseline.

We opted not to train LiLT on XFUND because the authors can’t validate the output of the LLM in some of the languages of this dataset, such as Chinese and Japanese. We focus on FUNSD and EPHOIE, using LiLT-RoBERTa-EN and LiLT-InfoXLM, which were made publicly available by the authors. These were previously pre-trained on large datasets of documents, following the common practice in the literature. Our experiments with FUNSD use both the public LiLT-RoBERTa-EN and a new model with a Portuguese RoBERTa model (LiLT-RoBERTa-PT) we created by coupling the RoBERTa-PT model with the base LiLT (layout only) from the official LiLT repository.

### 4.2 Results

We fine-tune the pre-trained LiLT models using our augmented datasets, detailed in Section 3.3. In our experiments, we combine the real partition with our augmented partitions in joint training. These results are compared to the ones reported by LiLT in the corresponding monolingual dataset plus task scenario, which we use as a baseline.

For both all datasets, we fine-tune in the Semantic Entity Recognition (SER) task, the problem of assigning the correct class to every entity inside the docu-



Table 3: Number of entities on FUNSD, by classes and augments.

Entities by Class		Number of Augmentations per Entity		Augmentations by Type	
Header	411	1	978	Complex	647
Question	3266	2	1580	Synonym	4106
Answer	2802	3	1560	Simple	375
Other	902	4 or More	1010	None	2283
Total	7411	Augments	14611	Augmented	5128

Table 4: Results for the SER (RoBERTa-EN) and RE (InfoXLM) tasks on FUNSD.

Train Partition	SER	RE
Real only (Reported)	88.41	62.76
Real + 1 Augment	88.82	64.4
Real + 2 Augments	<b>89.76</b>	68.44
Real + 3 Augments	89.04	<b>70.52</b>
Real + 4 Augments	<u>89.72</u>	<u>70.35</u>
Real + 5 Augments	89.02	69.55

ment. For FUNSD, we also fine-tune in the Relation Extraction (RE) task, the problem of extracting the relationships between headers and questions, and questions and answers. For each task, adapting layers are added to the output of the classification model. A full explanation of the network adaptations for each task can be found in LiLT. The implementation of these extra layers is found in the official LiLT GitHub, and we use it in our experiments.

Our results for FUNSD are presented in Table 4. We report the micro-averaged F1-score. We highlight that LiLT presents a better result for the monolingual SER task with RoBERTa-EN (the InfoXLM result is 85.86), but this model is not used in the RE task. For the consistency of our comparisons, we train SER with LiLT-RoBERTa-EN and RE with LiLT-InfoXLM. Both tasks are in the monolingual scenario, with only the vanilla FUNSD training set plus our augmentations.

These results show that our augmentation does improve the training set, as the baseline is improved in every case. For SER, we improve the error margin by 1.35 in the best case, and by 7.76 in RE. This better improvement for the RE task might be explained by the fact that we expand InfoXLM’s knowledge of the English language with our augmentations. InfoXLM is multilingual, and our fine-tuning with more examples of English texts seems to aid the model in this scenario. Finally, our results show that the augmentations have consistently improved the baseline in every training scenario.

Our results for EPHOIE are shown in Table 5. Again, we report the micro-averaged F1-score, using the results reported by LiLT as a baseline. Here, we use LiLT-InfoXLM for fine-tuning. We don’t use LiLT-RoBERTa-ZH because this model was not made

Table 5: Results for the SER task on EPHOIE.

Train Partition	Test F1-score
Reported - RoBERTa-ZH	97.97
Reported - InfoXLM	97.59
Real + 1 Augment	<b>99.2</b>
Real + 2 Augments	<u>99.19</u>
Real + 3 Augments	99.13

Table 6: Results for the SER task on NBID.

Train Partition	LiLT-EN	LiLT-PT
Real only	<b>99.54</b>	99.54
Augmented only	98.74	<b>1</b>
Real + Augment	99.4	99.4

available by the authors, but even with InfoXLM we can beat the baseline with a comfortable margin. The best result reported by LiLT is the LiLT-RoBERTa-ZH 97.97 score, which we beat with a best score of 99.2, an improvement of 1.23 out of a possible 2.03 margin.

Lastly, we present our results for NBID in Table 6. While the joint training does not benefit the model’s performance, unlike the previous results, in this case the augmented partition shows to be fully representative of the dataset’s domain. In a simple setting such as the NBID dataset (which features a very reduced set of templates), our template augmentation manages to create a representative clone of the original dataset.

### 4.3 Discussion

The results shown in the previous section show that our augmentation methods manage to improve the baseline model’s performance. We tailor each method for each dataset respectively, and as such some extra fine-tuning would be needed if the domains are to change. This is the main drawback of our LLM technique, apart from also requiring extra annotation in the way of a new label for each entity. Improvements on this can be found by also leveraging the LLM ability of understanding instructions, and so these limitations can be overcome by fine-tuning the model via prompt engineering.

The template approach finds in its need for a text generator its main limitation. The proposed approach

of building dictionaries from the dataset itself has the drawback of limiting variability in the augmented instances. As previously discussed, some LLMs may lack specialized domain knowledge for some documents and as such may not be a reliable way of generating text. These generators can be built from text dictionaries found online and other methods of random generation (such as generating random numbers to compose dates), and so may need to be defined on a case by case basis, which was the case for NBID.

Finally, as presented, both methods have limited scalability. There are only so many ways to rewrite a sentence and add meaningful variations to the dataset, and the template generations are tied to the number of templates and the available texts for filling these. As described in Section 3.2, the template method works well for domains with simple templates, especially when these templates are fully known. This was the case for NBID, where most of the templates in the training section also appeared in testing. However, these techniques are not suit for endless augmentation, and can only take the model performance so far in domains that are more complex, such as EPHOIE. Nonetheless, there is still room for improvement, as shown by our results.

## 5 CONCLUSIONS

In this work, we presented two new data augmentation strategies for documents, aiming at both complex and simple domains. We have discussed their strengths and weaknesses in relation to other methods. Finally, we show that these methods manage to improve the baseline model's performance. In future work, we aim to use these same methods in other datasets, showing their applicability in other domains.

## ACKNOWLEDGEMENTS

The authors would like to thank UNICO for all the support in the making of this research project and also NVIDIA Corporation for the generous donation of the Quadro RTX 8000 GPU that made our experiments possible. The authors also thank PROEX CAPES for the financing, and David Menotti thanks CNPq (# 315409/2023-1).

## REFERENCES

Biswas, S., Riba, P., Lladós, J., and Pal, U. (2021). Doc-synth: A layout guided approach for controllable doc-

ument image synthesis. In *Int. Conf. on Document Analysis and Recognition (ICDAR)*.

- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Chi, Z., Dong, L., Wei, F., Yang, N., Singhal, S., Wang, W., Song, X., Mao, X.-L., Huang, H., and Zhou, M. (2021). InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Human Language Technologies*, pages 3576–3588. Association for Computational Linguistics.
- Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., and Hu, G. (2020). Revisiting pre-trained models for Chinese natural language processing. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668. Online. Association for Computational Linguistics.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conf. of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.
- Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc.
- Guillaume Jaume, Hazim Kemal Ekenel, J.-P. T. (2019). Funsd: A dataset for form understanding in noisy scanned documents. In *Accepted to ICDAR-OST*.
- Guo, Q., Qiu, X., Liu, P., Shao, Y., Xue, X., and Zhang, Z. (2019). Star-transformer. In *Conf. of the North American Chapter of the Association for Computational Linguistics*.
- Guo, Z., Wang, P., Wang, Y., and Yu, S. (2023). Improving small language models on pubmedqa via generative data augmentation.
- Huang, Y., Lv, T., Cui, L., Lu, Y., and Wei, F. (2022). Layoutlmv3: Pre-training for document ai with unified text and image masking. *CoRR/arXiv*, abs/2204.08387.
- Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., and Park, S. (2022). Ocr-free document understanding transformer. In *European Conf. on Computer Vision (ECCV)*.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Int. Conf. on Neural Information Processing Systems (NeurIPS)*, pages 1097–1105.
- Li, M., Xu, Y., Cui, L., Huang, S., Wei, F., Li, Z., and Zhou,

- M. (2020). Docbank: A benchmark dataset for document layout analysis.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pre-training approach. cite arxiv:1907.11692.
- Márk, C. and Orosz, T. (2021). Comparison of data augmentation methods for legal document classification. *Acta Technica Jaurinensis*, 15.
- Pend, Q. et al. (2022). ERNIE-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3744–3756, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Pondenkandath, V., Alberti, M., Diatta, M., Ingold, R., and Liwicki, M. (2019). Historical document synthesis with generative adversarial networks. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 5, pages 146–151.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.
- Raman, N., Shah, S., and Veloso, M. (2021). Synthetic document generator for annotation-free layout recognition. *CoRR*, abs/2111.06016.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Soboroff, I. (2022). Complex document information processing (CDIP) dataset.
- Touvron, H. et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.
- Tu, Y., Guo, Y., Chen, H., and Tang, J. (2023). Layout-mask: Enhance text-layout interaction in multi-modal pre-training for document understanding.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.
- Wang, J., Jin, L., and Ding, K. (2022). LiLT: A simple yet effective language-independent layout transformer for structured document understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7747–7757. Association for Computational Linguistics.
- Wang, J., Liu, C., Jin, L., Tang, G., Zhang, J., Zhang, S., Wang, Q., Wu, Y., and Cai, M. (2021). Towards robust visual information extraction in real world: New dataset and novel solution. In *Proceedings of the AAAI Conf. on Artificial Intelligence*.
- Wojcik, L., Coelho, L., Granada, R., Führ, G., and Menotti, D. (2023). Nbid dataset: Towards robust information extraction in official documents. In *Anais da XXXVI Conference on Graphics, Patterns and Images*, pages 145–150, Porto Alegre, RS, Brasil. SBC.
- Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., and Zhou, M. (2019). Layoutlm: Pre-training of text and layout for document image understanding. *CoRR/arXiv*, abs/1912.13318.
- Xu, Y., Lv, T., Cui, L., Wang, G., Lu, Y., Florencio, D., Zhang, C., and Wei, F. (2022). XFUND: A benchmark dataset for multilingual visually rich form understanding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3214–3224, Dublin, Ireland. Association for Computational Linguistics.
- Ye, J., Xu, N., Wang, Y., Zhou, J., Zhang, Q., Gui, T., and Huang, X. (2024). Llm-da: Data augmentation via large language models for few-shot named entity recognition.
- Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. (2019). ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.
- Zhang, Z., Ma, J., Du, J., Wang, L., and Zhang, J. (2022). Multimodal pre-training based on graph attention network for document understanding. *CoRR/arXiv*, abs/2203.13530.
- Zhong, X., Tang, J., and Yepes, A. J. (2019). Publaynet: largest dataset ever for document layout analysis. In *2019 Int. Conf. on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE.
- Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*.