Short Paper Booklet

15th IAPR International Workshop on Document Analysis Systems

22 — 25 May 2022

DAS 2022 - La Rochelle

La Rochelle Université

L3i Informatique Image Interaction La Rochelle Univ.

IAPR

FOREWORD

We are very happy to welcome you to DAS 2022, the 15th IAPR International Workshop on Document Analysis Systems, held in La Rochelle, France, for the first time. Organizing an international workshop of such significant size after the COVID pandemic, aiming to welcome most of the participants on-site, is a challenge we are very happy to have taken on. Defining best-practice in organizing large hybrid events remains an on-going effort for the scientific community and we hope to have ensured a pleasant experience both for on- and off-site participants.

At the time of writing, over 70% of the registrations are for on-site participation. We are looking forward to hosting our friends and colleagues of the DAS community 4 years after we could last meet face-to-face. We are especially pleased to provide this opportunity to young researchers, some of whom will attend their first ever in-person scientific event. We supported their participation with considerably reduced registration fees for students and a financial assistance program.

We hope you will enjoy our city of La Rochelle. Located on the Atlantic coast of France, La Rochelle has recently been ranked among the most livable cities in France, in particular for students. The city has a rich historical fabric, with its old harbor and towers as its most well-known landmarks.

We will treat you the best way possible with a welcome cocktail in a splendid 18th century cloister (*Cloître des Dames Blanches*), part of the city hall, and a gala dinner in the old harbor, preceded by a sea tour to the picturesque *Fort Boyard* (as seen on TV in 70 countries).

The workshop will be hosted on-campus by La Rochelle Université, using state-of-the-art broadcasting equipment. The campus and all its workshop venues are located within walking distance of the conference hotels, the historic center and the *Minimes* beach.

Finally, we want to thank the numerous and deeply committed volunteers of the local organization team. Without them, and the support of this year's workshop sponsors, this 15th IAPR International Workshop on Document Analysis Systems would not be possible. Last but not least, we want to thank you, the participants, our friends and colleagues, for giving us the pleasure of your attendance, whether online or offline.

Welcome to La Rochelle!

May 2022

Jean-Marc OGIER,
Jean-Christophe BURIE
Mickaël COUSTATY
Antoine DOUCET

PREFACE

Welcome to the 15th IAPR International Workshop on Document Analysis Systems (DAS 2022). DAS 2022 was held in La Rochelle, France, during May 22–25, 2022, and brought together many researchers from Europe and abroad.

With the new remote access facilities, the workshop was not confined to a specific location. In a sense, this was truly a worldwide edition of DAS, taking place around the world in a coordinated fashion, employing a schedule we designed to support participation across a wide range of time zones. Of course, this came with some challenges, but also with interesting opportunities that caused us to rethink the way of fostering social and scientific interaction in this new medium. It also allowed us to organize an environmentally friendly event, extend the reach of the workshop, and facilitate participation from literally anywhere in the world for those with an interest in our field and an Internet connection. We truly hope we managed to make the most out of a difficult situation.

DAS 2022 continued the long tradition of bringing together researchers, academics, and practitioners in the research field of document analysis systems. In doing so, we built upon the previous workshops held over the years in Kaiserslautern, Germany (1994); Malvern, PA, USA (1996); Nagano, Japan (1998); Rio de Janeiro, Brazil (2000); Princeton, NJ, USA (2002); Florence, Italy (2004); Nelson, New Zealand (2006); Nara, Japan (2008); Boston, MA, USA (2010); Gold Coast, Australia (2012); Tours, France (2014); Santorini, Greece (2016); Wien, Austria (2018); and Wuhan, China (2020).

As with previous editions, DAS 2022 was a rigorously peer-reviewed and 100% participation single-track workshop focusing on issues and approaches in document analysis and recognition. The workshop comprised presentations by invited speakers, oral and poster sessions, and a pre-workshop tutorial, as well as distinctive DAS discussion groups.

This year we received 94 submissions in total, 78 of which were in the regular paper track and 16 in the short paper track. All regular paper submissions underwent a rigorous single-blind review process where the vast majority of papers received three reviews. The reviewers were selected from the 80 members of the Program Committee, judging the originality of work, the relevance to document analysis systems, the quality of the research or analysis, and the overall presentation. Of the 78 regular submissions received, 52 were accepted for presentation at the workshop (67%). Of these, 31 papers were designated for oral presentation (40%) and 21 for poster presentation (27%). All short paper submissions were reviewed by all three program co-chairs. Of the 16 short papers received, all 16 were accepted for poster presentation at the workshop (100%). The accepted regular papers are published in this proceedings volume in the Springer Lecture Notes in Computer Science series. Short papers appear in PDF form on the DAS conference website.

The final program included six oral sessions, two poster sessions, and the discussion group sessions. There were also two awards announced at the conclusion of the workshop: the IAPR Best Student Paper Award and the IAPR Nakano Best Paper Award. We offer our deepest thanks to all who contributed their time and effort to make DAS 2022 a first-rate event for the community.

In addition to the contributed papers, the program also includes two invited keynote presentations by distinguished members of the research community: Andreas Dengel from the German Research Center for Artificial Intelligence (DFKI, Germany) and Adam Jatowt from the University of Innsbruck (Austria).

We furthermore would like to express our sincere thanks to the tutorial organizer, Himanshu Sharad Bhatt from American Express AI Labs, for sharing his valuable scientific and technological insights. Special thanks are also due to our sponsors IAPR, the L3i Laboratory, AriadNext, Esker, IMDS, GoodNotes, Yooz, MyScript, ITESOFT, TEKLIA, VIALINK, and the Région Nouvelle Aquitaine and Communauté d'Agglomération de La Rochelle, whose support, especially during challenging times, was integral to the success of DAS 2022.

The workshop program represented the efforts of many people. We want to express our gratitude, especially to the members of the Program Committee for their hard work in reviewing submissions. The publicity chairs, Richard Zanibi (USA) and Joseph Chazalon (France), helped us in many ways, for which we are grateful. We also thank the discussion group chairs, Michael Blumenstein (Australia) and Umapada Pal (India), for organizing the discussion groups, and the tutorial chairs, Rafael Dueire Lins (Brazil) and Alicia Fornes (Spain), for organizing the tutorial. A special thank you goes to the publication chair, Cheng-Lin Liu (China), who was responsible for the proceedings at hand. We are also grateful to the local organizing committee who made great efforts in arranging the program, maintaining the web page, and setting up the meeting platform with support for remote attendance. The workshop would not have happened without the great support from the hosting organization, La Rochelle University.

Finally, the workshop would have not been possible without the excellent papers contributed by authors. We thank all the authors for their contributions and their participation in DAS 2022! We hope that this program will further stimulate research and provide practitioners with better techniques, algorithms, and tools. We feel honored and privileged to share the best recent developments in the field of document analysis systems with you in these proceedings.

April 2022

Seiichi Uchida
Elisa Barney Smith
Véronique Eglin

PROGRAM COMMITTEE

General Chair
Jean-Marc Ogier, La Rochelle, France

Conference Chair
Jean-Christophe Burie, La Rochelle, France
Mickaël Coustaty, La Rochelle, France
Antoine Doucet, La Rochelle, France

Conference Committee
Program Chairs
Seiichi Uchida, Fukuoka, Japan
Elisa Barney-Smith, Boise, USA
Véronique Eglin, Lyon, France

Industrial Chairs
Vincent Poulain d'Andecy, Aimargues, France
Robin Mélinand, Nantes, France

Tutorial Chairs
Rafael Dueire Lins, Recife – Pernambuco, Brazil
Alicia Fornes, Barcelona, Spain

Discussion Group Chairs
Michael Blumenstein, Sydney, Australia
Umapada Pal, Kolkata, India

Publication Chair
Cheng-Lin Liu, Beijing, China

Publicity Chairs
Richard Zanibi, New York, USA
Joseph Chazalon, Le Kremlin-Bicêtre, France

ORGANIZING LOCAL COMMITTEE

Beatriz Martínez Tornés
Carlos Gonzalez
Damien Mondou
Dominique Limousin
Emanuela Boros
Guillaume Bernard
Ibrahim Souleiman Mahamoud
Kais Rouis
Lady Viviana Beltran Beltran
Latifa Bouchekif
Marina Dehez–Clementi
Mélanie Malinaud
Muhammad Muzzamil Luqman
Musab Al-Ghadi
Nathalie Renaudin-Blanchard
Nicholas Journet
Nicolas Sidère
Souhail Bakkali
Théo Taburet
Zuheng Ming

TABLE OF CONTENT

Leveraging Guides to Empower Open Data Research

Christina Christodoulakis, Moshe Gabel, and Angela Demke Brown

Department of Computer Science, University of Toronto, Canada
{christina, mgabel, demke}@cs.toronto.edu

Abstract. Data packages in Open Data repositories often contain data guides: supplementary materials with information supporting interpretation and consumption of contents of files containing tabular data. This short paper describes the design of a system that discovers, unifies and links metadata from guide files to Open tabular data. Enriching tabular data will facilitate tasks like table search, interpretation, and integration for Open Data users such as scientists and journalists.

Keywords: Open Data · Metadata Discovery · Tabular Data.

1 Introduction

Governments and industry are embracing Open Data as they recognize the impact on scientific, economic, social, and environmental development of communities [2]. While this data is freely available, discoverability, accessibility and reusability remain significant barriers from a stakeholder perspective [5,6].

Automatically annotated header lines of tabular data found in Open Documents, such as in [3], provide some information about the contents of the files, usually encoded in attribute names, that can be used for search and integration [4]. Packages in Open Data repositories often contain *data guides*, which are supplementary materials often in tabular format with information supporting interpretation and consumption of contents of files containing tabular data and are associated with a specific set of tables in data files. Metadata described in guides may include extended semantics and contextual information for tables, attributes, and attribute values, as well as other metadata such as languages used, data types, formatting, units, scales, etc.

Users wanting to efficiently search, interpret, and combine tabular data from Open Documents cannot easily benefit from them as is currently no automated way of leveraging guides. Some of the main challenges to solving this include diverse contents, structures, and naming conventions. Figure 1 shows an example adapted from a set of real Open Data files. It shows a data table extracted from the file `electricity.csv`, and guide annotations extracted from a supplementary file `dictionary.csv`.[1]

[1] CSV is a popular Open Data format widely used in a variety of domains for its simplicity and effectiveness in storing and disseminating data, and is frequently used to describe data guides.

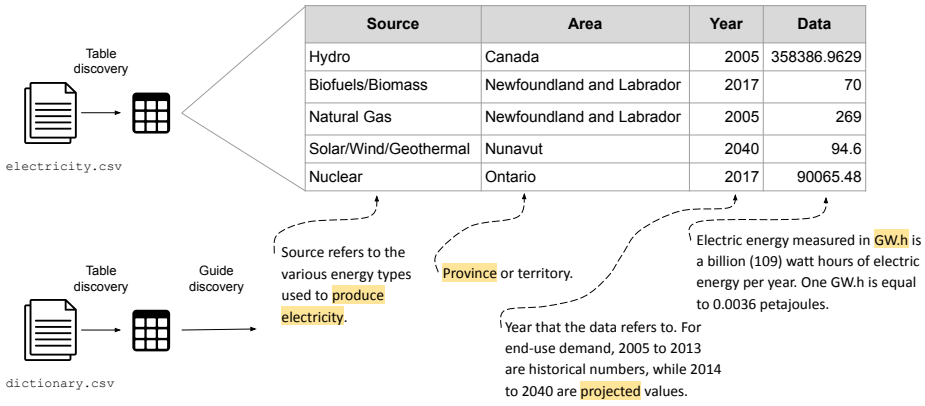


Fig. 1. An example Open Data table extracted from file `electricity.csv` is matched to attribute guides extracted from guide file `dictionary.csv`.

We use the example in Figure 1 to demonstrate several benefits of automatic guide discovery, unification, and linking:

- **Table Search:** Given only the data table in Figure 1 and the query `"projected electricity generation per Canadian province"` an information retrieval engine is unlikely to return this table in top ranked results. However, if `dictionary.csv` is identified as a guide file, and the metadata it describes is properly extracted and combined with the data table annotations, the user can now discover this table successfully, as the description of the attribute `Year` (identified and extracted from a guide file) informs us that there is a range of years for which the data is historical and another range for which it is **projected**, and the description of the attribute `Area` informs us that the values are **provinces** or territories.
- **Interpretation:** A user consuming the data in Figure 1 must know that for this particular table, data recorded up to 2013 is historical, while the rest is projected. Value guides can indicate aggregation rows (e.g., `Area` = `Canada`), and explain value semantics in greater detail.
- **Integration:** Consider a user that wants to build a data set of electricity production across North America, and discovers a data set for US electricity production across states, with electricity production measured in KWh (kilowatt-hour), however, their seed data set as seen in Figure 1 records data in GWh (gigawatt-hour). Knowing the units and scale of the data will add a much needed step of data conversion before integration.

We focus on discovering, unifying, and linking guides from CSV files to tables annotated in CSV files such that approaches supporting tasks like search and integration may benefit from previously unused rich metadata. This requires designing an end to end system that addresses table discovery, guide discovery and linking of the two. Such a system could be a great asset in connecting open data sleuths such as scientists and journalists with open data tables.

2 System Design and Implementation

We are designing a system which will scan CSV files crawled from an Open Data repository to discover and annotate tables. The system will process the annotated tables to discover guides, which it will extract and unify to a common schema. Following that, the system will discover the links between unified guides and annotated tables. Finally, the system will present users with an interface for table and guide annotation, browsing, and review.

Guide discovery and extraction: While some Open Data repositories support and encourage annotating published resources as guides, more often than not such files are not explicitly annotated, and do not follow a single naming convention. Furthermore, formatting of guide files is not uniform, making automatic extraction of guide elements challenging. We studied a random sample of 100 Open Data packages with CSV guide resources crawled from Open Data repositories and observed guide files with guide information related to tables, table attributes, and attribute values. Ideally their respective guides are each well structured tables, with each row corresponding to guides of a table, table attribute, or attribute value respectively. In practice, we frequently observe element guide information presented as merged tables (e.g., attribute and attribute value guides combined in a single guide table, guides for attributes of multiple tables combined in a single guide table, etc.), rotated, or even nested.

Data Model: Following our observations on Open CSV guides, we design a unifying data model for guides as well as software that extracts and maps information from existing guides to the unified model. For the unifying model, we identified a set of guide fields for table attributes (see Table 1) and attribute values. In a preliminary evaluation on a second random sample of 50 guide files, our model captures the information in these files successfully.

Table 1. Guide fields identified for table attributes. OPT indicates optional fields, MLT indicates multilingual.

Field	OPT	MLT	Description
Header	-	-	Short text for identifying the attribute.
Title	-	✓	A version of the header in natural language text.
Description	✓	✓	A detailed definition of the attribute in natural language text.
Note	✓	✓	Natural language text with context on the attribute data.
Unit	✓	-	Attribute value units of measurement (e.g., L, mpH, $, %, etc.).
Scale	✓	-	Scale of the reported attribute values (e.g., billions, 10^{-2}, etc.).
Domain	✓	-	Legal values for an attribute. Defined by a range or dictionary.
Datatype	✓	-	E.g., text, integer, Boolean, date, etc.

Open Document data tables may be published with partially or fully encoded data values. For example, encoded values may be used to represent missing or

redacted information (e.g., `*`, `n/d`, `x`, `NA`, etc.), or an encoding scheme may be used for all data values (e.g., replacing province names with province codes given a value dictionary). Furthermore, guides may contain rich descriptions for non-encoded values. We identify a value guide fields as a subset of fields used for attribute guides, namely, a title, description, and notes.

Prototype implementation: Our prototype has several components; a Web frontend in JavaScript, a Flask-based backend supported by a PostgreSQL relational database implementing the model we name GuideDB, and a Lucene [1] backend for indexing and customized search. We provide an API for writing and reading JSON annotations to and from the database.

Tables in Open CSV (comma separated value) files can pose a significant challenge to identify due to significant variety in structure and formatting. For table discovery in CSV files we use Pytheas, a weighted rule-based table discovery system for CSV files [3].[2] We are currently designing and implementing a hybrid rule-based/learning-based approach to automatically identify guide tables, classify table structure into a set of known designs, and unify guide information into a common schema. We are also designing a customized ranking algorithm based on Lucene and table and attribute similarity distance functions that take into account annotated guides to support table search and integration.

Via the Web interface, users can manually add or generate automated table annotations on CSV files, save automated annotations, or edit or generate their own. Users can annotate multilingual column headers, scales, and units, identify guide tables, extract and normalize guide fields, and match guide fields to a data table, a table attribute, or an attribute value. The interfaces support editing of annotations persisted in GuideDB. We use this interface to annotate the ground truth against which we will evaluate our automatic guide discovery and unification methods.

References

1. Apache Lucene, https://lucene.apache.org/
2. Capgemini Consulting: Creating Value through Open Data: Study on the Impact of Re-use of Public Data Resources (2015), accessed: 2019-09-23
3. Christodoulakis, C., Munson, E., Gabel, M., Brown, A.D., Miller, R.J.: Pytheas: Pattern-based table discovery in CSV files. PVLDB **13**(11), 2075–2089 (2020)
4. Liu, Y., Bai, K., Mitra, P., Giles, C.L.: Tableseer: Automatic table metadata extraction and searching in digital libraries. In: Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries. p. 91–100. JCDL '07, Association for Computing Machinery, New York, NY, USA (2007)
5. Miller, R.J., Nargesian, F., Zhu, E., Christodoulakis, C., Pu, K.Q., Andritsos, P.: Making open data transparent: Data discovery on open data. IEEE Data Eng. Bull. **41**(2), 59–70 (2018)
6. Máchová, R., Hub, M., Lněnička, M.: Usability evaluation of open data portals: Evaluating data discoverability, accessibility, and reusability from a stakeholders' perspective. Aslib Journal of Information Management **70** (05 2018)

[2] https://github.com/cchristodoulaki/Pytheas

CULDILE: Cultural Dimensions of Deep Learning, A Document Analysis System for Historical Documents

B. Gatos[1], G. Sfikas[1], P. Kaddas[1,2] and G. Retsinas[1]

[1]Institute of Informatics and Telecommunications,
National Centre for Scientific Research "Demokritos", GR 153 10, Athens, Greece
{bgat,sfikas,pkaddas, georgeretsi}@iit.demokritos.gr
[2]Department of Informatics and Telecommunications,
University of Athens, GR 157 84, Athens, Greece

Abstract. In this paper, an overview of the Greek National project CULDILE (CULtural DImensions of deep Learning) is presented. It includes a user-friendly software platform to analyze, enhance, index and provide access to a large number of historical document pages. CULDILE platform includes functionality for image pre-processing (image binarization and enhancement, page split etc.), automatic metadata extraction (e.g. detect the existence of handwritten or machine-printed text, tables, seals, signatures etc.), document classification and keyword spotting. The focus of this paper is on the specifications and architecture of CULDILE as well as on relevant general practices and tools.

Keywords: Historical Document Image Processing, Document Image Pre-processing, Document Metadata Extraction, Document Classification, Keyword Spotting.

1 Introduction

The CULDILE project[1] focuses on pioneering research activities in historical document image processing aiming to significantly improve access to historical documents and to take away the barriers that stand in the way of the mass digitization of cultural heritage documents. It includes a user-friendly software platform to analyze, enhance, index and provide access to a large number of historical document pages. A private new dataset from the library of the Piraeus Bank Group Cultural Foundation (PIOP)[2] is used to provide a first proof-of-concept. CULDILE platform includes functionality for image pre-processing (e.g. image binarization and enhancement, page split), automatic metadata extraction (e.g. detect number of columns, the existence of handwritten or machine-printed text, ornamental symbols, seals, signatures), document classification and keyword spotting (search by a keyword marked by the user). In this paper, we give an overview of relevant general practices and tools as well as of CULDILE specifications and architecture.

[1] http://culdile.bookscanner.gr
[2] https://www.piop.gr/el/vivliothiki.aspx

2 General practices and tools

At a first step we recorded all general practices and tools relevant to CULDILE research activities. This includes guidelines for digitization, tools for image annotation, platforms for document image visualization and metadata description schemes.

2.1 Guidelines for Digitization

Recommendations for selecting a particular format or standard for the digitization-related activities can be found in the following sources:

- IMPACT Centre of Competence[3], formats and standards related to master files, metadata, OCR results, delivery files, guidelines for semantic technologies, linguistic resources and tools packaging.
- JISC[4], guidelines for preparation of collection materials, copyright clearance, creation of metadata, scanning, web delivery, digital archiving and preservation.
- National Library of France (BnF)[5], guidelines for storing and processing digital collections, exploring and sharing resources, metadata management and catalogues.
- The National Archives and Records Administration (NARA), USA[6], recommendations for capture, minimum metadata, formats, naming, storage and quality control.

2.2 Tools for Image Annotation

Existing tools that can be used for document image annotation include:

- Aletheia[7], an advanced system for accurate and yet cost-effective analysis, recognition and annotation of scanned documents.
- labelme[8], a graphical image annotation tool using polygons written in Python.
- Computer Vision Annotation Tool (CVAT)[9], an interactive video and image annotation tool for computer vision.

2.3 Platforms for Document Image Visualization

Platforms that provide access to the page images of book and manuscripts include Open Library[10], Google Books[11], Many Books[12] and National Library of Greece, e-Reading Room[13].

---

[3] https://www.digitisation.eu
[4] https://digitisation.jiscinvolve.org/wp/
[5] https://www.bnf.fr/en
[6] https://www.archives.gov
[7] https://www.primaresearch.org/tools/Aletheia
[8] https://github.com/wkentaro/labelme
[9] https://github.com/openvinotoolkit/cvat
[10] https://openlibrary.org
[11] https://books.google.com
[12] https://manybooks.net
[13] https://ereading.nlg.gr/en/

2.4 Metadata Description Schemes

Metadata are usually described following schemes such as:

- Encoded Archival Description (EAD)[14]
- Dublin Core[15]
- Metadata Object Description Schema (MODS)[16]
- Machine Readable Cataloguing (MARC)[17]
- Metadata Encoding & Transmission Standard (METS)[18]
- CIDOC Conceptual Reference Model (CRM)[19]

3 System Specifications

In order to design the CULDILE platform, we took into consideration a long list of specifications that resulted after discussing with all involved partners (people from archives, industry and the research community). The most important are the following:

- The software should be user friendly and permit several levels of access (guest, authorized user, moderator, validator and admin) in a web-based and multi-threaded environment.

- Facilities for document image viewing on page or book/manuscript level should be provided as well as searching based on filters using metadata information.

- A list of pre-defined metadata should be supported (e.g. document category, existence of handwritten or machine-printed text, ornamental symbols, tables, seals, signatures) as well as custom metadata defined by the user. Metadata should be global (concern the whole document page, e.g. number of columns, letter color, existence of tables or images) or local (concern a certain part of the page defined by a polygon, e.g. an area containing handwritten text or a signature, see Fig.1a).

- All metadata should be filled in or edited by the user while all initial entries (values or/and defining polygons) should be automatically calculated by document image processing and deep learning methods that will be implemented and integrated in the platform. Re-training of these methods should be also provided based on selected existing data.

- A list of image pre-processing capabilities (e.g. image enhancement, page split) should be provided.

- Search by a keyword marked by the user should be also supported (query by example keyword spotting).

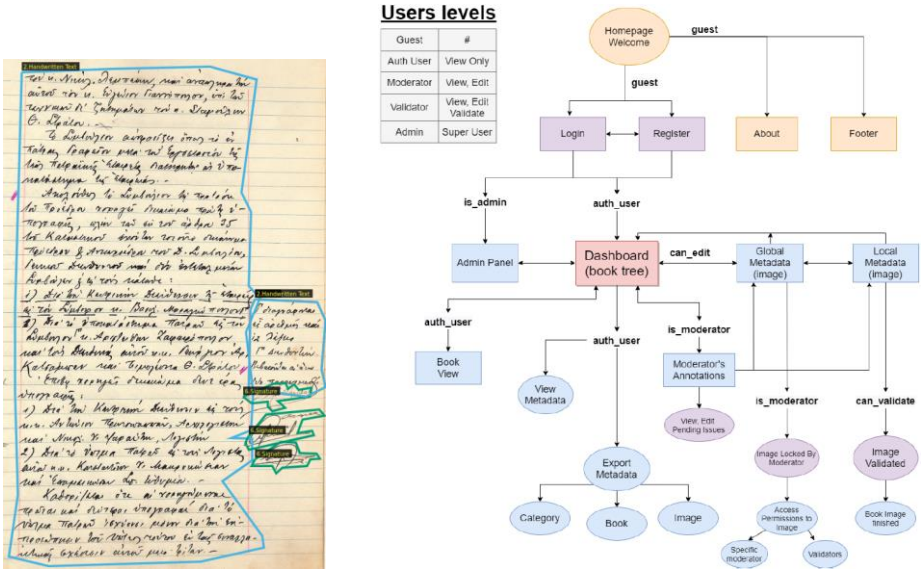
- All metadata should be saved in a convenient JSON format while export capabilities to the most famous metadata description schemes (see 2.4) should be supported.

[14] https://www.loc.gov/ead/
[15] https://dublincore.org
[16] http://www.loc.gov/standards/mods/
[17] https://www.loc.gov/marc/
[18] https://www.loc.gov/standards/mets/
[19] https://www.cidoc-crm.org

4 CULDILE Architecture

The architecture of the CULDILE platform is demonstrated in Fig. 1b. Different levels of access are supported with the following functionality:

- Guest: Access only at general information about the platform
- Authorized User: Read only rights, search and view data through the dashboard tree view, page browsing using thumbnails and book view, metadata view and export facilities.
- Moderator: authorized user with permissions to edit data, check and edit all global and local metadata, keep a record of actions done or pending, lock pages during processing.
- Validator: checks and validates all actions done by moderators, verifies all data that will be provided to all authorized users.
- Admin: Full access to all platform functionality, an admin panel is used to monitor all processes and actions.



(a) (b)

Fig. 1. (a) Example of local metadata defined by polygons. (b) CULDILE architecture overview

Acknowledgment

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH-CREATE-INNOVATE (project code: T1EΔK-03785) as well as by the program of Industrial Scholarships of Stavros Niarchos Foundation.

Exploring Uses of Normalizing Flows for Document Image Processing: Text Super-Resolution and Binarization

Giorgos Sfikas[1,3,4], George Retsinas[2], and Basilis Gatos[3]

[1] CIL/IIT, NCSR "Demokritos", Greece
[2] School of ECE, NTUA, Greece
[3] Dpt. of CS and Engineering, University of Ioannina, Greece
[4] Dpt. of Surveying and Geoinformatics Engineering, Univ. of West Attica, Greece

Abstract. Normalizing flows are powerful models that elegantly combine invertible neural networks with probabilistic modeling. We explore uses of the normalizing flow framework for two document image processing tasks: Text Super-Resolution and Binarization.

Keywords: Normalizing Flows, Text Super-Resolution, Binarization

1 Introduction to Normalizing flows

In the normalizing flow (NF) framework [6], a probability density function $p_X(\cdot)$ is sought to be estimated given a finite set of samples $X = \{x_1, x_2, \cdots, x_N\}$ known to come from that distribution. The core adea is to express the available observed data in terms of a distribution $p_U(\cdot)$, that is termed the "base" distribution and is typically a standard isotropic Gaussian. A diffeomorphism (a smooth, bijective function) $f : \mathbb{R}^D \to \mathbb{R}^D$ is assumed to transform data X into images $\{f_\theta(x_1), f_\theta(x_2), \cdots, f_\theta(x_N)\}$, that are required to follow the (typically) Normal distribution $p_U(\cdot)$, and images and pre-images share the same dimensionality, denoted as D. θ is a set of parameters that define the transformation. The term "normalizing flow" stems from exactly this requirement; f_θ is responsible for creating data that are normally distributed, and in this sense it is "normalizing". Transformation function f_θ is defined as a neural network, and learning the data is performed by finding the optimal network parameters that transform X as required. Concerning notation, in what follows we will write $f_\theta(x)$ or $f(x;\theta)$ or simply f to refer to the same transformation.

Formally, we can write [1]:

$$p_X(x) = p_U(f_\theta(x))|det\frac{\partial f_\theta}{\partial x}(x)|, \tag{1}$$

where we use the change-of-variables formula between pdfs, θ are the parameters that define the transformation f, and $\partial f_\theta(x)/\partial x$ is the Jacobian matrix for f_θ. A very important constraint over f_θ is that it needs to be bijective. In practice, network f_θ needs to be structured so as to have both a Jacobian and an inverse

f_θ^{-1} that are easily computable. If network f_θ is defined as a composition $f_\theta(x) = f^K \circ f^{K-1} \circ \cdots \circ f^1(x;\theta)$, training the normalizing flow is tantamount to solving the following maximum likelihood problem:

$$\arg\max_\theta \log \mathcal{N}(f(x;\theta)) + \sum_{k=1}^{K} \log |det \frac{f^k}{z^k}(z^k;\theta)| \tag{2}$$

where we used $z^0 = u$, $z^K = x$, $z^k = f^k(z^{k-1})\ \forall k \in [1, K]$.

The standard formulation of Normalizing flows described above, fits the unsupervised setting of density estimation perfectly. For a supervised learning setting, where we have pairs of source $X = \{x_1, x_2, \cdots, x_N\}$ and target objects or labels $Y = \{y_1, y_2, \cdots, y_N\}$, this standard paradigm can be extended to a formulation of conditional Normalizing flows [6, 4]. Under this setting, transformation f is required to map from $y|x$ to $z|x$, i.e. now targets are mapped to a latent space by means of the normalizing flow, while all are conditioned on the source data x. It is then straightforward to rewrite the density of eq. 1 as a conditional density:

$$p_{Y|X}(y|x) = p_U(f_\theta(y|x))|det \frac{\partial f_\theta}{\partial x}(y|x)|, \tag{3}$$

and the maximum likelihood objective of eq. 2 in its conditional iteration as:

$$\arg\max_\theta \log \mathcal{N}(f(y|x;\theta)) + \sum_{k=1}^{K} \log |det \frac{f^k}{z^k}(z^k|x;\theta)|, \tag{4}$$

where we now set $z^0 = u$, $z^K = y$, $z^k = f^k(z^{k-1}|x)\ \forall k \in [1, K]$. Learning a model on data X, Y can hence be performed by optimizing eq. 4 given the available data and w.r.t. the transformation parameters θ. Transformation f is diffeomorphic thus differentiable by assumption, hence in practice we can choose to use any standard gradient-based optimizer (e.g. SGD, Adam).

Interestingly, flows have been shown to lead to state-of-the-art performance in a number of tasks, using only a Maximum Likelihood criterion to train [3, 4]. Other models often require multiple priors that entail requiring hyperparameters that weight the importance of each prior w.r.t. the likelihood term. These play often a critical role in the success of the architecture in practical applications. Further useful traits of NFs include: efficient and exact density evaluation; potential memory savings; an inherently probabilistic formulation, without many of the difficulties typically associated to probabilistic modeling and other generative models [3].

2 Formulation of Text Super-resolution and Binarization as Normalizing Flows

At a high-level, we follow the way the conditional architecture of SRFlow [4] is built, and we use the same way flow layers are grouped into a cascade of L levels.

Flow level are each related to a spatial resolution, in particular $H/2^l \times W/2^l$, where $H \times W$ stands for the initial resolution. A level can broken down into K groups of flow layers ("flow-steps" [4]). In turn, each flow-step is made up of the following four flow layers: actnorm, 1×1 convolution, affine injector and conditional affine coupling. For our super-resolution application we use a number of levels $L = 3$, and for the binarization application we use a single level $L = 1$, hypothesizing that the binarization problem is less complex / demanding than super-resolution. We use patches sized 160×160 pixels for our experiments. In super-resolution, we sub-sample the training patches to 40×40 to create low-res / high-res pairs. We use a pre-trained RRDB backbone in both cases. Inference is performed as a process of sampling from the learned density, conditioned on the input, i.e. the low-res image or the non-binarized image respectively. In figures 1 we show 2 we show visual results. Regarding the employed datasets for training and testing, we have used the DIBCO binarization competition datasets [7] and the new "PIOP-DAS" dataset [8].

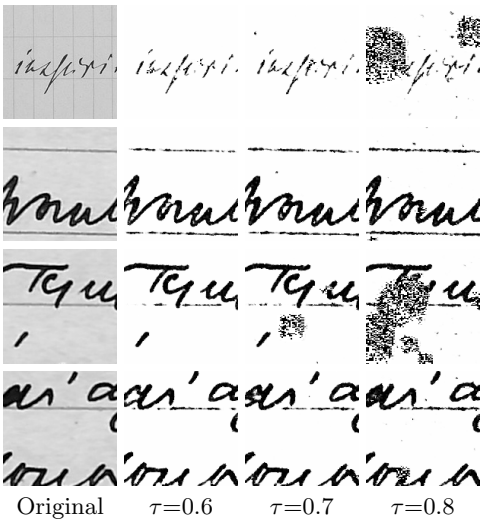


Fig. 1. Binarization results: Original images and binarization results for different "temperature" hyperparameter values τ.

3 Future work

After obtaining the reported first very preliminary though somewhat promising results, we plan to continue our research on NFs along the following axes: First, setup sets of experiments on both considered problems, evaluate numerically the results, and compare to state-of-the-art methods. Concerning super-resolution, consider integrating with a shape-based approach for the prior, leading to an extra loss term (e.g. [2], or the recent [5]). Also, test more challenging SR up-sampling scales. We also envisage using SR combined with binarization, in a

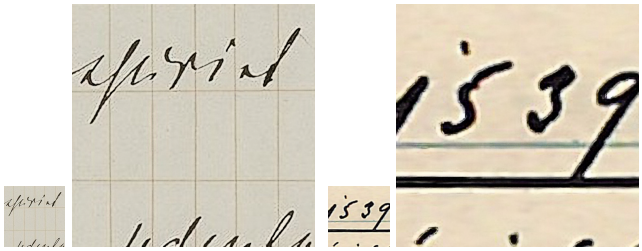


Fig. 2. Super-resolution results: Original images and super-resolved images (τ=0.7).

scenario where a binarization components may aid in avoiding to super-resolve areas that are unimportant (background) or noisy (jpeg artifacts), or aid in properly evaluating the result (by disregarding background from SR result evaluation).

Acknowledgments

This research has been partially co-financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the calls "RESEARCH - CREATE - INNOVATE" (project *Culdile* - code T1EΔK-03785), and "OPEN INNOVATION IN CULTURE" (project *Bessarion* - T6YBΠ-00214).

References

1. Dinh, L., Krueger, D., Bengio, Y.: NICE: non-linear independent components estimation. In: Bengio, Y., LeCun, Y. (eds.) ICLR (2015)
2. Giotis, A.P., Sfikas, G., Nikou, C., Gatos, B.: Shape-based word spotting in handwritten document images. In: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR). pp. 561–565. IEEE (2015)
3. Kingma, D., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: NIPS (2018)
4. Lugmayr, A., Danelljan, M., Van Gool, L., Timofte, R.: SRFlow: Learning the super-resolution space with normalizing flow. In: ECCV. pp. 715–732. Springer (2020)
5. Nakaune, S., Lizuka, S., Fukui, K.: Skeleton-aware text image super-resolution. In: BMVC (2021)
6. Papamakarios, G., Nalisnick, E., Rezende, D., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. JMLR **22**(57) (2021)
7. Pratikakis, I., Zagoris, K., Kaddas, P., Gatos, B.: ICFHR2018 competition on handwritten document image binarization contest. In: ICFHR. pp. 1–1 (2018)
8. Sfikas, G., Retsinas, G., Giotis, A.P., Gatos, B., Nikou, C.: Keyword spotting with quaternionic ResNet: Application to spotting in Greek manuscripts. In: Proceedings of the International Workshop on Document Analysis Systems (DAS) (2022)

Document Intelligence Metrics for Visually Rich Document Evaluation

Jonathan DeGange[1], Swapnil Gupta[2], Zhuoyu Han[1]
Krzysztof Wilkosz[3], and Adam Karwan[3]

[1] Ernst & Young (EY) LLP USA
{jonathan.degange, zhuoyu.han}@ey.com
[2] EY Global Delivery Services India LLP
swapnil.gupta1@gds.ey.com
[3] EY GDS (CS) Poland Sp. z o.o.
{krzysztof.wilkosz, adam.karwan}@gds.ey.com

Abstract. The processing of Visually-Rich Documents (VRDs) is highly important in information extraction tasks associated with Document Intelligence. We introduce *DI-Metrics*, a Python library devoted to VRD model evaluation comprising text-based, geometric-based and hierarchical metrics for information extraction tasks. We apply *DI-Metrics* to evaluate information extraction performance using publicly available *CORD* dataset, comparing performance of three SOTA models and one industry model. The open-source library is available on GitHub[4].

Keywords: Hierarchical Information Extraction · Visually Rich Document · Document Intelligence · Metrics.

1 Introduction

Retrieval of the relevant data is often termed Key Information Extraction (KIE) or Information Extraction (IE). Semi-structured forms and documents with complex layout features are commonly known as Visually-Rich Documents (VRD) [7]. IE from VRDs is a sub-task of document understanding, often termed Document Intelligence[5] (DI), which applies artificial intelligence and machine learning to business documents and processes.

Key Information Extraction from VRDs is a challenging task of active research in the research community [8][9]. Many fields in semi-structured documents such as invoices or receipts are hierarchical (e.g. item description, item count, item total, all roll up to a singular parent line item class), and as previously stated, require two-dimensional processing. Current SOTA approaches are often based on self-supervised pre-training and transfer learning[6]. Models often comprise a multi-modal representation of the page content's text, location (bounding boxes), and other important visual semantic queues.

[4] https://github.com/MetricsDI/DIMetrics
[5] https://sites.google.com/view/di2019
[6] https://docs.microsoft.com/en-us/azure/cognitive-services/form-recognizer

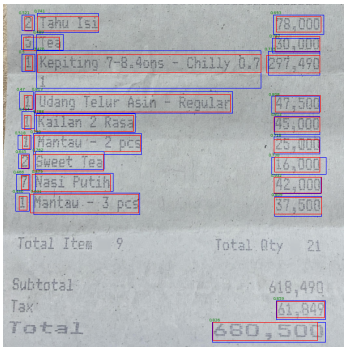
2 Metrics

We provide a library to ease consistent comparison of VRD model performance on IE tasks. The library is a collection of existing and new IE metrics (Table 1) accessible through a Python3 API. Many metrics are dynamic programs based on edit distance, and they are known to be computationally expensive. Our implementations are accelerated by pre-compilation in Cython [1]. We also introduce a novel metric for handling evaluation of hierarchical fields, *Unordered Hierarchical Edit Distance* (UHED).

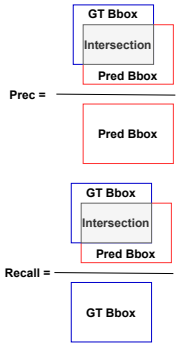
Table 1: Metrics available in the the *DI-Metrics* library

Metric Type	**Metrics Name**	**Range**
Text-Based (Field Level)	Exact Match	*True, False*
	Raw Levenshtein Distance	*0 - min(GT, P)*
	Raw Longest Common Subsequence (LCSeq)	*0 - min(GT, P)*
	Token Classification	*0 - 1*
Geometric-Based (Field Level)	Grouped Bbox by class IoU (IoU_G)	*0 - 1*
	Constituent Bbox by class IoU (IoU_C)	*0 - 1*
Hierarchical (Document Level)	Hierarchical Edit Distance (HED)	*0 - 1*
	Unordered Hierarchical Edit Distance (UHED)	*0 - 1*

Text-based metrics measure the presence of typing or spelling errors and access the convergence of two strings. In **Exact Match (EM)** metric, we simply check whether the entire predicted string *P* is exactly the same as the ground truth string *GT*. **Levenshtein Edit Distance (LED)** between two words is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other. The **Longest Common Subsequence (LCSeq)** is the minimum number of insertions and deletions required to change one string to the other.



(a) Receipt ground truths (red), predictions (blue)



(b) Geometrical explanation of IoU

Fig. 1: Visualization and explanation of geometric-based metrics

Geometric-based metrics consider the ratio between the overlap of the location of the area of the text in ground truth and predicted bounding boxes. Geometric-based metrics are useful especially in IE when the targeted text for extraction coincidentally appears in multiple locations on the same page (i.e. right answer, wrong location), and

also for document layout analysis tasks. Figure 1a presents an example of receipt with labeled visualization of ground truth bounding boxes and predicted fields. When using **Grouped Bbox by class ($\mathbf{IoU_G}$)** approach one computes the overlap of aggregated boxes by calculating a convex-hull minimal spanning box of all constituent bounding boxes surrounding the entire field and thus include any spaces between constituent OCR as well. Similar yet slightly different, **Constituent Bbox by class ($\mathbf{IoU_C}$)** is adapted from the DocBank dataset paper [6], where instead of taking the area of the entire field, we only consider the areas of individual tokens (words).

Hierarchical Metrics are applied when the fields of interest are nested. In [4], edit distances are extended from strings to table cells of strings, using a tree-based edit distance for table cell recognition. **Hierarchical Edit Distance (HED)** was proposed by [2]. This metric also covers information about non-nested and hierarchical fields (line-items), effectively only requiring that the ordering of line-items within a document and words within a field remain the same, while the ordering of fields within a line-item may be permuted without impacting the distance. Our proposed **Unordered Hierarchical Edit Distance (UHED)** relaxes HED, allowing unordered lists of line-items. We apply Hungarian assignment algorithm to find the optimal (GT, P) pairs by minimizing the matrix of input distances for each possible candidate pairs via bipartite matching [5].

3 Experimental Results

To test application of the metrics on models and data, we use CORD Receipts dataset. In Table 2 we present a comparison of HED and UHED metrics for three models: LayoutLM Base V1 [10], DeepCPCFG [2], and Microsoft Form Recognizer pre-built receipt model.

Table 2: Ordered and Unordered Hierarchical Edit Distance metrics.

	HED			UHED		
CORD	**F1**†	**Precision**†	**Recall**†	**F1**†	**Precision**†	**Recall**†
LayoutLM + PSL LI Rules	0.89	0.88	0.91	0.92	0.92	0.94
LayoutLM + Simple LI Rule	0.86	0.85	0.89	0.92	0.96	0.90
DeepCPCFG	**0.96**	**0.97**	**0.97**	**0.97**	**0.98**	**0.97**
MSFT Form Recognizer	0.81	0.91	0.75	0.85	0.96	0.78

†Reported values are the mean of F1, precision and recall for each document's HED scores. F1 is not directly comparable to precision and recall.

LayoutLM is a BERT-like transformer model, where bounding box and WordPiece embeddings are summed together as inputs to the transformer hidden layers. We employ sequence labeling approach with single Softmax classifier after the encoder, and train over approximately 18,000 internal proprietary invoices using cross entropy loss function. To group nested line-item classifications, we use Probabilistic Soft Logic (PSL) [3] to classify parent line item IDs. The PSL rules combine first-order logic with probabilistic graphical model to perform collective classification of line-items using outputs from the LayoutLM token classification Softmax classifier. To assess effectiveness of PSL line item grouping, we also implement Simple LI Rule, a rule-based method for assigning bounding boxes group labels.

DeepCPCFG uses an expert-provided grammar and language model potentials as rules, operating on two-dimensional sequences formed by a directed graph representation of the page structure [2]. Unlike LayoutLM, DeepCPCFG does not require bounding box labels, but uses ground truth key-value pairs as inputs, and latently learns the mapping to bounding boxes on page.

Microsoft Form Recognizer is used as an industry benchmark end-to-end model, accessible via API calls. We benchmark the pre-built receipt model. We do share results of training a custom model on CORD data, due to inability to create custom parent-child predictions with the API.

4 Discussion and Conclusion

We have shared *DI-Metrics*, a library for objective evaluation of IE Document Intelligence Tasks. The library provides a comprehensive set of metrics for use by researchers and industry practitioners to use and transparently benchmark information extraction models. In this paper, we also introduced UHED metric.

Disclaimer: The views reflected in this article are the views of the authors and do not necessarily reflect the views of the global EY organization or its member firms.

Acknowledgement: The authors would like to thank the following colleagues: Freddy Chua, Sunil Tiyyagura, Hamid Motahari and Nigel Duffy for their thoughtful feedback and suggested edits.

References

1. Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D.S., Smith, K.: Cython: The best of both worlds. Computing in Science & Engineering **13**(2), 31–39 (2011)
2. Chua, F.C., Duffy, N.P.: DeepCPCFG: Deep learning and context free grammars for end-to-end information extraction. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021. pp. 838–853. Springer International Publishing, Cham (2021)
3. Duffy, N.P., Puranam, S.A., Dasaratha, S., Phogat, K.S., Tiyyagura, S.R.: DeepPSL: End-to-end perception and reasoning with applications to zero shot learning (2021)
4. Hwang, W., Yim, J., Park, S., Yang, S., Seo, M.: Spatial dependency parsing for 2D document understanding. CoRR **abs/2005.00642** (2020)
5. Jonker, R., Volgenant, T.: Improving the Hungarian assignment algorithm. Operations Research Letters **5**(4), 171–175 (1986)
6. Li, M., Xu, Y., Cui, L., Huang, S., Wei, F., Li, Z., Zhou, M.: DocBank: A benchmark dataset for document layout analysis. CoRR **abs/2006.01038** (2020)
7. Liu, X., Gao, F., Zhang, Q., Zhao, H.: Graph convolution for multimodal information extraction from Visually Rich Documents. arXiv preprint arXiv:1903.11279 (2019)
8. Sarkhel, R., Nandi, A.: Improving information extraction from Visually Rich Documents using visual span representations **14**(5) (2021)
9. Tecuci, D., Palla, R., Nezhad, H., Ahuja, N., Monteiro, A., Ishkhanov, T., Duffy, N.: DICR: AI assisted, adaptive platform for contract review. Proc. of AAAI **34**, 13638–13639 (04 2020)
10. Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., Zhou, L.: LayoutLMv2: Multi-modal pre-training for Visually-Rich Document understanding (2020)

Confidence Score for Unsupervised Foreground Background Separation of Document Images

Soumyadeep Dey and Pratik Jawanpuria

Microsoft, India Development Center
{Soumyadeep.Dey,Pratik.Jawanpuria}@microsoft.com

Abstract. Foreground-background separation is an important problem in document image analysis. Popular unsupervised binarization methods (such as the Sauvola's algorithm) employ adaptive thresholding to classify pixels as foreground or background. In this work, we propose a novel approach for computing confidence scores of the classification in such algorithms. This score provides an insight of the confidence level of the prediction. The computational complexity of the proposed approach is the same as the underlying binarization algorithm. Our experiments illustrate the utility of the proposed scores in various applications like document binarization, document image cleanup, and texture addition.

Keywords: Binarization · Cleanup · Confidence score.

1 Introduction

The technique to classify foreground and the background pixels is known as binarization. Various supervised and unsupervised techniques have been reported in literature for document image binarization. The simplest method to achieve binarization is thresholding gray-scale or color document images. Analytical techniques for document image binarization involve segmenting the foreground pixels and background pixels based on some threshold. For binarization of an image, a global threshold is computed based on the distribution of pixel intensities in [1]. Sauvola and Pietikäinen proposed an adaptive thresholding method for document image binarization in [2]. In this method, a threshold is computed for each pixel based on local mean and variance surrounding the pixel. Lazzara and Geraud proposed proposed a multi-scale version of Sauvola's algorithm in [3] to make it adaptable for low contrast images. The above techniques use pixel intensity based information to perform local/global threshold to obtain binary image. In contrast, Peng *et al.* [4] proposes a Gabor filter based stroke orientation computation technique for document binarization task. A fast Fuzzy C-Means clustering based document binarization technique has been proposed in [5]. A regression based method for background estimation is proposed in [6]. The estimated background is subtracted from the input image and global thresholding is applied for the binarization task. In recent time, document binarization is also explored using supervised techniques like maximum entropy classification [7], multi-resolutional attention model [8], convolutional neural network [9].

There are various downstream applications of foreground segmentation from the background pixels. However, mere binary level segmentation is not enough to achieve these downstream tasks, since perfect segmentation of foreground from background pixels is difficult to achieve and very much data dependent in case of supervised methods. Therefore it is important to have a confidence score for each pixels in an unsupervised manner. In this work, we have proposed an unsupervised scoring function for each pixel of an image to define its confidence to be labeled as background or foreground. The primary contribution of the paper lies in defining the scoring function in an unsupervised manner. We have also shown the application of these scores in various document processing techniques.

2 Computation of scores for each pixel

In the Sauvola's algorithm [2], a threshold is computed for each pixel using the Eq 1, where, for an input image I, $R = \frac{\max(I)-\min(I)}{2}$.

$$T_W(p) = m_W^p \times [1 + k \times (\frac{s_W^p}{R} - 1)] \tag{1}$$

The threshold T is computed for each pixel (p) based on a window W of size $n \times n$ surrounding it, where m_W^p, s_W^p respectively represent mean and standard deviation of W around pixel p, and k lies between $0 \leq k \leq 1$.

Empirically, it has been observed that binarization using the the thresholds obtained from Eq 1 misses foreground pixels in many scenarios. An example failure case is shown in Fig. 1(A)(ii). Such a segmented output may also used in downstream applications such as image clean-up task. An example output of such a setting is provided in Fig. 1(B)(ii). We again observe a substantial loss of foreground information with the segmented output obtained using Eq 1.

To alleviate the above concern, we propose to compute a confidence score for each pixel. The goal of this score is to reflect the confidence about the class prediction. Typically, confidence on prediction should increase if the pixel value lies further away from the computed threshold. Based on this intuition, we define normalized confidence values of background $(C_W^b(p))$ and foreground $(C_W^f(p))$, for each pixel p using Eqs 2 and 3.

$$C_W^b(p) = \begin{cases} \frac{I(p)-T_W(p)}{\max(I)-T_W(p)} & \text{if } I(p) > T_W(p) \\ 1 - \frac{T_W(p)-I(p)}{T_W(p)-\min(I)} & \text{otherwise} \end{cases} \tag{2}$$

$$C_W^f(p) = 1 - C_W^b(p) \tag{3}$$

Here, $\max(I)$ and $\min(I)$ represent maximum and minimum value of any pixel of an input image I, respectively. It should be noted that the confidence score lies in the interval $[0, 1]$. Overall, with the availability of such scores, downstream tasks may take a more suitable decision (for pixels with low confidence scores) to avoid foreground information loss. The proposed confidence scores can be generated with any adaptive thresholding approach. For empirical comparison, we considered Sauvola's thresholding algorithm [2] as the base method.

3 Applications

We discuss four applications of the proposed score values.
Document binarization: We develop a modified version of [2] to handle missing data using the pixel scores. Our code is available at `https://tinyurl.com/scoredbinarization`. Our result for a sample image is in Fig. 1(A)(iii).
Document cleanup: The missing data from [2] results in patchy cleanup. The proposed score function can be used to obtain a non-patchy cleaned up version of the input image (please refer to Fig. 1(B)).
Preprocessing: The proposed score values can also be used as pre-processing step to various algorithms. For instance, its application as preprocessing step to DNN based image cleanup [9] is discussed in Fig. 1(C).
Texture transfer: The goal here is to transfer the content of an input image to a new textured background without any loss of original content of the original document. Here, we used the foreground and background scores to achieve this objective. An example of such texture transfer is shown in Fig. 1(D).

4 Conclusion

We presented an unsupervised method to compute confidence score for each pixel in a document image. We have further shown the utilization of these computed scores for various downstream tasks. More theoretical investigation and experimentation on the discussed applications are interesting future directions.

References

1. N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man, and Cybernatics*, 9(1):62–66, 1979.
2. J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern Recognition*, 33:225–236, 2000.
3. G. Lazzara, and T. Géraud. Efficient multiscale sauvola's binarization. IJDAR, **17**(2), 105–123. 2014.
4. X. Peng, H. Cao, K. Subramanian, R. Prasad, and P. Natarajan. Exploiting stroke orientation for crf based binarization of historical documents. ICDAR 2013.
5. T. Mondal, T. Coustaty, M. Gomez-Krämer, and P. J. Ogier. Learning free document image binarization based on fast fuzzy c-means clustering ICDAR 2019.
6. G. D. Vo and C. Park. Robust regression for image binarization under heavy noise and nonuniform background. *Pattern Recognition*. **81**, 224 – 239. 2018.
7. N. Liu, D. Zhang, X. Xu, W. Liu, D. Ke, L. Guo, S. Shi, H. Liu, and L. Chen. An iterative refinement framework for image document binarization with bhattacharyya similarity measure. ICDAR 2017.
8. X. Peng, C. Wang, and H. Cao Document binarization via multi-resolutional attention model with drd loss. ICDAR 2019.
9. S. Dey and P. Jawanpuria. Light-weight Document Image Cleanup using Perceptual Loss. ICDAR 2021.

Fig. 1. (A,B,C,D)(i) input images; A(ii) binary image using [2]; A(iii) improved binary image using the proposed scores; B(ii) image cleanup using [2]; B(iii) image cleanup using the proposed scores; C(ii) image cleanup using [9]; C(iii) image cleanup using [9] with proposed score based pre-processing; D(ii) texture to be transferred to D(i); D(iii) texture transferred image using proposed scores.

The Human Element in Document Analysis Systems

Daniel Lopresti [0000-0003-2129-4223]

Lehigh University, Bethlehem PA 18015, USA
lopresti@cse.lehigh.edu

Abstract. The Document Analysis Systems (DAS) workshop series began in 1996 and has now reached its 15th instantiation, a notable record of longevity and success. DAS was conceived with an explicit focus on systems, as opposed to the broader range of topics appearing at ICDAR and ICFHR. The term "system" is often defined as "a regularly interacting or interdependent group of items forming a unified whole." Yet, the human element is often excluded from our use of "system," even though humans create the documents we analyze, the documents themselves are designed for human consumption, and humans are the ultimate end users (beneficiaries) of the results of document analysis. In this short position paper, we begin with a brief summary of research that mentions human involvement as reported at DAS and other conferences in the field. We then discuss several concrete examples where excluding the human from the system has serious negative consequences. In certain cases, issues of bias and fairness may be at stake, an important consideration as many applications of artificial intelligence are now receiving critical attention. A more intentional inclusion of the human element would yield better outcomes for those who are impacted by the systems we build, and lead to interesting research questions as well. Our hope is to generate productive discussion at the DAS workshop and beyond.

Keywords: Document Analysis Systems, Human-Document Interaction, Performance Evaluation, Applications, Ethics.

1 Introduction

The Document Analysis Systems (DAS) workshop series began in 1996 and has now reached its 15th instantiation, a notable record of longevity and success. DAS was conceived with an explicit focus on systems, as opposed to the broader range of topics appearing at ICDAR and ICFHR. In particular, the latter often see significant attention aimed at lower-level methods (e.g., basic classification techniques), which form only one part of a larger system when it comes to a real-world implementation. On the other hand, the term "system" is often defined as "a regularly interacting or interdependent group of items forming a unified whole."[1] The notion of multiple components is key, along with their interaction / interdependence, and unified whole. A familiar example is the canonical document processing pipeline consisting of pre-processing, layout analysis, text/graphics recognition, and post-processing. Surprisingly, though, the human

[1] Owing to space constraints on this short format paper, we are unable to include references.

element is often excluded from our use of "system," even though humans create the documents we analyze, the documents themselves are designed for human consumption, and humans are the ultimate end users (beneficiaries) of the results of document analysis. A system that achieves very high accuracy when tested in isolation, but that does a very bad job when human users are added to the mix, has failed in its mission.

In this short position paper, we briefly summarize situations where human involvement has been incorporated in research reported at DAS, ICDAR, and ICFHR. Such cases appear relatively rare. We then discuss several concrete examples where excluding the human from the system has serious negative consequences. A more intentional inclusion of the human element would yield better outcomes for end users and others who are impacted by the systems we build. It would also lead to interesting new research questions for the community to work on.

2 General Considerations

It is impossible to provide a full survey in this short paper given the length constraints. Instead, we sketch out some general areas where human involvement has been explicitly discussed. We begin by noting that nearly every paper uses training data that has been collected and annotated by humans. Normally this involves a relatively small number of experts with specific knowledge (often a single student), and no attempt is made to build systems that are accessible to average end users. Some research has aimed at optimizing the ground-truthing process since it can be extremely tedious. A related concept, "Human-in-the-Loop," is widely quoted across a range of AI applications, including document analysis where it has been employed for accomplishing transcription tasks more efficiently. But none of this work treats users as diverse individuals; they are specialists asked to perform a job and expected to do it well (ideally perfectly).

Where has the non-expert user played a central role? It is interesting to recall that some of the earliest work in OCR was done by Ray Kurzweil in his attempts to build reading machines for the blind. Also in the health-related space, Plamondon and his colleagues have applied the lognormal model for handwriting generation for diagnosing various mental and physical conditions. Toyama, et al. have combined eye tracking with augmented reality and document analysis to build systems to assist with reading. These applications are closer to the spirit we have in mind. Here users are not regarded as experts who succeed at a task or fail; rather, to be successful the system must be designed to accommodate the widest possible range of users. If there is a failure, it is the fault of the system and not the user.

Looking outside these applications, which draw from health or educational goals, there seems to be a lack of consideration of the ways document analysis systems impact people. In the broader world of machine learning and AI, however, there is much recent discussion about potential abuses, bias, and fairness of various technologies. We have seen, for example, facial recognition systems that are clearly biased against individuals with darker skin, which generate understandable outrage and calls for change.

Are there analogous situations in the field of document analysis? Or should we believe our work is immune to such considerations? How do we go about identifying

those who might be impacted adversely? Can we learn anything from our colleagues who work in human-computer interaction and other areas of AI who may be ahead in their thinking? Are there best practices we can adapt to our own field?

3 Three Illustrative Cases

In this section we identify three cases where ignorance of the downstream impacts has negative impacts on users; the human element is left out of the system. Two can be regarded as the result of simplistic performance evaluations that focus on simple accuracy rates (an "average case" analysis). The majority wins out while outliers lose. But when we connect these abstract quantities to real people, the result may be systems that are biased and unfair, just as we have seen in the case of facial recognition.

The third example reflects on the attitude our field takes toward what qualifies as a "publishable" research contribution. While seemingly a pure scientific question, this, too, can have negative impacts on colleagues who work in different parts of the world.

3.1 Reliably Reading Hand-Marked Paper Ballots

It is now widely accepted that one of the most trustworthy voting mechanisms is the use of hand-marked paper ballots. The Optical Mark Reading problem has a long history in the processing of paper forms, with commercial systems available from IBM as early as the 1930's. This problem appears so simple that it has not attracted much attention in the community for decades. It would be easy for a university student to build a system that was nearly 100% accurate reading test ballots prepared by other students.

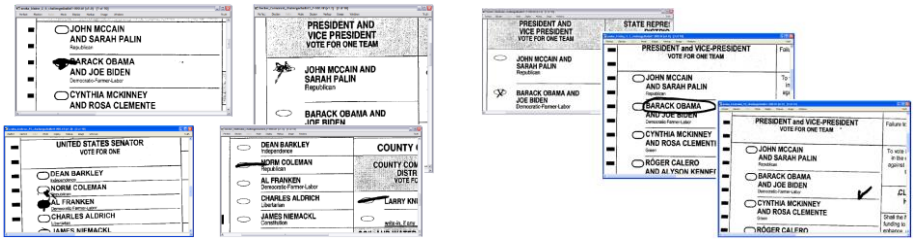


Fig. 1. Sloppy-but-valid marks (left) and non-confirming marking styles (right).

When examining a real population of voters, however, representing all demographics from across society, the problem becomes much more challenging. Voting is a right, and a system that fails to count certain voters' ballots accurately is a serious problem. If those voters exhibit common traits – for example, lower literacy at following instructions, or less facility with the language in use – then there may be bias that cuts across ethnic, racial, or socioeconomic lines. A ballot-reading system that looks like it is doing a good job on average may still be disenfranchising groups of voters.

In earlier work, we collected and characterized a set of challenged paper ballots from the 1998 US Senate Election in Minnesota. Figs. 1-2 show examples from this

collection representing valid votes that may not be read correctly without incorporating human levels of understanding in the system, something not possible today.

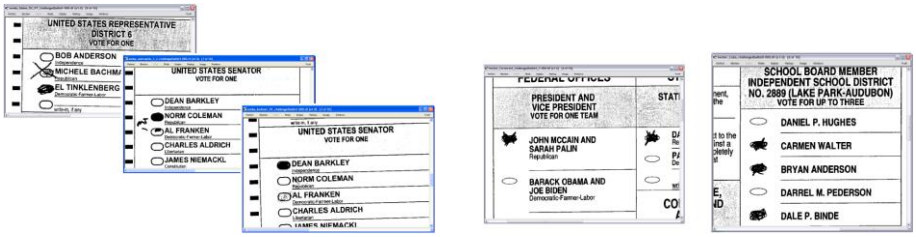


Fig. 2. Attempts to cancel a vote (left) and valid votes that look cancelled (right).

3.2 Robust Signature Verification for Elections

A similar concern arises when voting by mail or by provisional ballot. In such cases, the voter is usually required to sign an outer envelope as proof of identity. This signature is compared to one collected when the voter first registered – sometimes decades earlier. Currently these comparisons are made by election officials who are aware of the life events that could affect someone's signature: a name-change due to marriage, a hand injury, a stroke, or forgetfulness about how one signed so many years ago. Challenges to validity by opposing parties can be a major point of contention that could determine the results of a close election. In theory an automated system would do a better job because it is apolitical, but first we must eliminate inherent biases that disadvantage some voters more than others in the signature matching process.

A better approach in both of these cases would be to explicitly include the human element, incorporating more broadly representative data when training and testing, and conducting error analyses that not only report averaged accuracies but also consider the very real impacts on people and on demographic groups.

3.3 Document Analysis for Under-Resourced Languages

We conclude by raising another issue those active in the community will recognize. There are thousands of languages in the world today, but the vast majority of document analysis research reported at DAS, ICDAR, and ICFHR represents only a small percentage of these. It is often the case that submissions applying known techniques to a new language will be rejected for "lack of contribution." While it is true that many of our systems will function similarly on new inputs given the right training data, it may be too harsh a generalization to claim there is nothing to be learned in studying existing methods applied to a new language (the first language "wins the race"). More importantly, rejecting such papers without consideration erects a wall that excludes worthy colleagues. The result is cultural biases that, again, negatively impact real people.

We might instead ask what would make such work interesting and publishable, and proactively develop public guidelines that help those who wish to join our community know what we are seeking. A collaborative approach that incorporates the human element would advance both research and inclusivity in the field of document analysis.

Robust Extraction of Marked-Up Text Sections from Scientific Document Printouts

Mark-Christoph Müller[0000−0001−5639−7682]

Heidelberg Institute for Theoretical Studies gGmbH
Heidelberg, Germany
mark-christoph.mueller@h-its.org

Abstract. We present a simple tool for extracting text and markup information from printouts of (not only) scientific documents. While the heavy-lifting OCR is done by off-the-shelf TESSERACT, our focus is on detection, extraction, and basic *categorization* of color-highlighted text sections, as well as on providing a framework for downstream processing of extraction results. The tool can be useful for document analysis tasks that must, or benefit from being able to, use printed paper.

Keywords: Document Images · Information Extraction · OCR · Multimodality · Natural Language Processing

1 Introduction

Despite the shift towards PDF and XML, **printed paper** is still crucial for scientific document use.[1] It is the medium of choice for *active reading*, supporting straightforward markup with highlighter pens, which is commonly done during manual excerption from scientific literature. However, as soon as the highlighted text is supposed to undergo further *computational* processing, paper ceases to be practical. Biomedical database curation [1] is a case in point: Here, human domain experts often use paper printouts to mark up relevant sections in scientific documents, but for the subsequent database insertion (often done by other people), the data has to be re-keyed manually, which is both inefficient and error-prone.

We present a simple OCR-based document analysis tool which combines the advantages of working with paper hard-copies and the efficiency of automatic text recognition and extraction. In essence, the tool mainly integrates an OCR component (off-the-shelf TESSERACT, see below), a simple image processing module, and an XML-based multi-level annotation processing framework from natural language processing (NLP). Thus, our focus is on providing robust **core extraction functionality** based on proven state-of-the-art components, rather than on optimizing individual modules. Also, by using an NLP data representation

[1] This work was done as part of the project DeepCurate, which is funded by the German Federal Ministry of Education and Research (BMBF) (No. 031L0204) and the Klaus Tschira Foundation, Heidelberg, Germany.

framework including an API, we establish straightforward technical connectivity between extraction results and downstream processing (see Section 4). Code and data are available at https://github.com/nlpAThits/docimg2mmax.

2 System Overview

The tool significantly extends and improves our previous work in [4], where earlier versions of some of the current functionality were used. Basically, the tool reads a scanned document, consisting of one image per page, recognizes and extracts the text content, then (optionally) analyses the image for color-highlighted sections, and creates special word-level annotations for highlighted content.

We use the MMAX2[2] [5] multi-level annotation processing framework for representation and further processing of OCR and extraction results. MMAX2 supports visualization and manual annotation of the extracted data (see below), and also provides a Python API [3]. In a nut shell, data in MMAX2 is stored in the form of so-called MARKABLES, which aggregate arbitrary attribute-value pairs and associate these with underlying, immutable text data (in this case with the OCR result).

OCR is performed with TESSERACT (tested with version 4.1.1), which is only loosely integrated and called via Python sub-processes. TESSERACT can output its results in hOCR format[3], which includes highly detailed recognition information. The generation of hOCR output is always activated, while other parameters (`--oem`, `--psm`, `--dpi`, and `--tessdata-dir`) are directly passed through. This way, a high degree of transparency and flexibility is maintained. After recognition, the hOCR file is analysed, and the recognized text as well as bounding box and confidence information for line, word, and character elements is stored in MARKABLES on different annotation levels. Optionally, the tool can also create an HTML file with an SVG-based overlay of the original image, which visualizes the extracted marked-up text. Markup detection and extraction works by analyzing the page image, identifying colored areas, and mapping these to previously extracted words, based on the latters' bounding boxes. Highlighting can appear either *horizontally* on the desired text, or, for larger sections that span several lines, *vertically*, e.g. on page margins (see Figure 1). The detection of colored image areas takes advantage of the fact that, in an RGB image, *non-colored* pixels have highly similar values in their three channels, while whenever at least one channel value differs above a certain threshold (we use an absolute value of 10) from the others, the pixel actually has a discernible color.

3 Examples

We demonstrate the tool on a black-and-white printout of an open-access scientific paper [2] which has been marked up using different colors and then scanned

[2] https://github.com/nlpAThits/MMAX2
[3] http://kba.github.io/hocr-spec/1.2/

in 300 dpi. Figure 2 shows two example extraction results. In each example, the left image is a part of the scanned page image, and the right image shows the rendering of the extracted full text in MMAX2. Highlighted words are rendered with a yellow background. Note the OCR accuracy (courtesy of TESSERACT), which at least for standard text is almost perfect. Boxes in the left images are drawn automatically around highlighted words. For each highlighted word, two properties are determined, viz. the *percentage of the word area* that is actually highlighted, and the *dominant highlighting color*. A threshold on the first property is used (here: 10%) to discard words that are only marginally touched by coloring. The second property is intended to capture a kind of highlighting *category* by allowing to cluster words that were highlighted *in the same color*. It is implemented by just selecting, from the colored part of each word's bounding box, the most frequent RGB triple. Table 1 shows the respective properties for one word each from the four colored regions in Figure 2. Visualization of the dominant colors is for illustration only; actual clustering / categorization will have to be done by analysing the ratio of the three color channel values.

Word	*"reaction"*	*"peptide"*	*"substrate"*	*"crowding"*
% HL	36%	69%	89%	85%
Dominant color	245:255:244	253:224:246	241:255:255	203:254:213

Table 1. Highlighted words with automatically extracted dominant color.

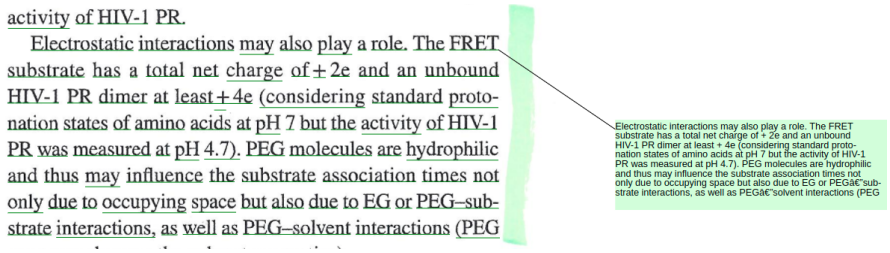


Fig. 1. Detail of HTML file with extracted *vertical* markup.

4 Summary & Outlook

The presented text extraction and markup detection tool is deliberately designed to be simple and reduced to core functionality. Nevertheless, our rather superficial evaluation showed that both out-of-the-box OCR and markup extraction quality is very good, provided that 1) the image quality is good (clean black-and-white printout) and 2) appropriate highlighter colors are used. In an actual application scenario, these factors can easily be controlled for. Next, we are going to evaluate the applicability of the tool in a literature-based biomedical database curation scenario. Database curation from documents should be able to benefit strongly from powerful and flexible text search, including e.g. handling of synonyms. Once a document has been processed with our tool, these functionalities

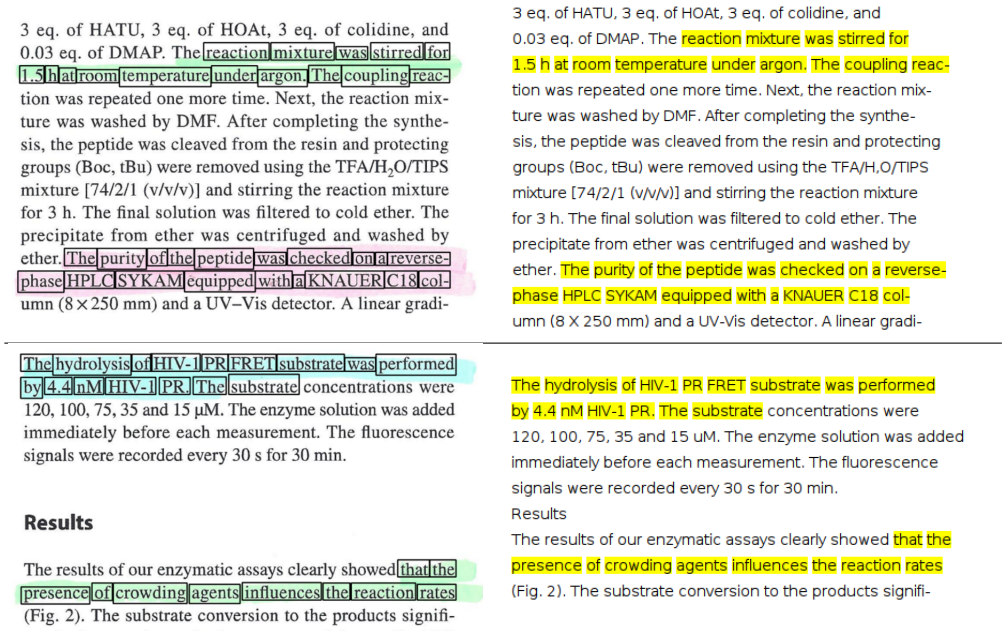


Fig. 2. Images with color highlighting (left) and extracted text (right).

are available with little extra effort on the basis of the MMAX2 format and the Python API.

References

1. International Society for Biocuration: Biocuration: Distilling data into knowledge. PLOS Biology **16**(4), 1–8 (2018). https://doi.org/10.1371/journal.pbio.2002846
2. Maximova, K., Wojtczak, J., Trylska, J.: Enzymatic activity of human immunodeficiency virus type 1 protease in crowded solutions. European Biophysics Journal **48**(7), 685–689 (2019). https://doi.org/10.1007/s00249-019-01392-1
3. Müller, M.C.: pyMMAX2: Deep access to MMAX2 projects from python. In: Proceedings of the 14th Linguistic Annotation Workshop. pp. 167–173. Association for Computational Linguistics, Barcelona, Spain (Dec 2020), https://aclanthology.org/2020.law-1.16
4. Müller, M.C., Ghosh, S., Wittig, U., Rey, M.: Word-level alignment of paper documents with their electronic full-text counterparts. In: Demner-Fushman, D., Cohen, K.B., Ananiadou, S., Tsujii, J. (eds.) Proceedings of the 20th Workshop on Biomedical Language Processing, BioNLP@NAACL-HLT 2021, Online, June 11, 2021. pp. 168–179. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.bionlp-1.19
5. Müller, M.C., Strube, M.: Multi-level annotation of linguistic data with MMAX2. In: Braun, S., Kohn, K., Mukherjee, J. (eds.) Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods, pp. 197–214. Peter Lang, Frankfurt a.M., Germany (2006)

Out-of-Distribution Performance in Document Image Classification: Initial Findings

Stefan Larson[1], Gordon Lim[2], Yutong Ai[2], and Brian Chen[2]

[1] DryvIQ, Ann Arbor MI, USA
[2] University of Michigan, Ann Arbor MI, USA

Abstract. The RVL-CDIP corpus [1] is the *de facto* standard benchmark for document classification, yet to our knowledge all studies that use this corpus do not include evaluation on *out-of-distribution* documents. This paper reports on a work-in-progress evaluation of document classifiers trained on RVL-CDIP and tested on a new set of over 3000 out-of-distribution documents. Based on initial experiments, we find that standard image-based classifiers appear to struggle at predicting out-of-distribution inputs.

Keywords: Document Classification · Out-of-Distribution Detection · Image Classification · Datasets

1 Introduction

The task of automated document classification has wide-ranging use cases, especially in industry where it can be used to apply labels to massive amounts of documents. In many applications a desirable document classification system must be able to both (1) classify documents with high accuracy and (2) distinguish between documents that are within the training label set and those that are outside of the label set. In this paper we investigate model performance on two types of *out-of-distribution* (OOD) documents: (a) those that fall outside of the target label set's scope (i.e., not an RVL-CDIP category); (b) those that are in-domain yet are from a different distribution than RVL-CDIP. A common way of distinguishing the former type is to use a decision threshold on the confidence scores obtained from the logits of a model. If l is a vector of logits, and $p = \text{softmax}(l)$ a vector of confidence scores, then a threshold t can be used such that

$$\text{decision rule} = \begin{cases} \text{in-domain}, & \text{if } \max(p) \geq t \\ \text{out-of-domain}, & \text{if } \max(p) < t. \end{cases}$$

Prior work has shown that even if classifiers perform well on *in-distribution* inputs, they may struggle on the task of out-of-domain prediction (e.g., [2] for short-text classifiers). Moreover, few studies have investigated out-of-distribution performance for document classifiers.

This extended abstract begins to fill this gap by reporting on work towards a new evaluation corpus targeted at the out-of-distribution problem for document classifiers. Our new dataset consists of 3161 documents that are out-of-distribution vis-à-vis the RVL-CDIP document classification corpus. We train several image-based document classifiers on the full RVL-CDIP dataset, and then evaluate these models on our new out-of-distribution dataset.

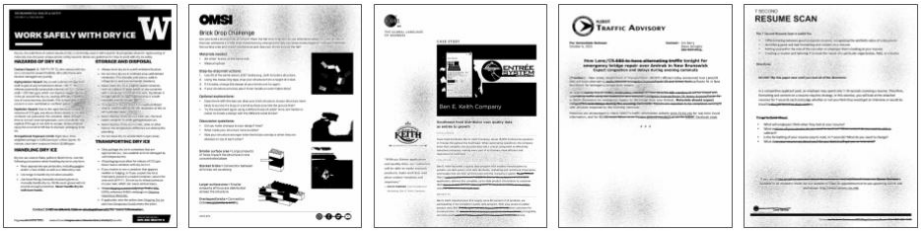


Fig. 1. Example out-of-distribution document images with added scanner-like noise.

2 Datasets

The RVL-CDIP corpus consists of grayscale images of scanned documents from the IIT-CDIP collection, a large repository of publicly-available documents that were released as part of litigation against several tobacco-related companies. As such, all documents in the RVL-CDIP corpus are tobacco-related. The corpus consists of 16 categories. Each category has 20,000 training samples (320,000 total training samples). There are 40,000 validation and 40,000 test images. All documents from this dataset are from the year 2006 or earlier, with 2006 being the year that the IIT-CDIP collection was released.

Our new out-of-distribution dataset consists of two subsets of data: document images that (a) do not belong to any of the 16 in-domain RVL-CDIP categories (we call this subset OOD-a); (b) belong to one of the 16 RVL-CDIP categories yet are not from IIT-CDIP or tobacco-related (OOD-b). Our out-of-distribution documents were collected from three internet sources: (1) Google and Bing web searches; (2) the public DocumentCloud repository; and (3) the scraped PDFs from the Common Crawl. The DocumentCloud repository contains a large number of government, legal, or public service related documents that were made available through serviced Freedom of Information Act (FOIA) requests. Currently, we have collected 3,161 out-of-distribution documents, and have converted the first page of each document to a grayscale image. In contrast to documents from DocumentCloud, the majority of documents from web searches and Common Crawl are "born digital" (i.e., they are not scanned versions of physical documents). For this reason we also use the Augraphy tool [3] to add scanner-like noise to our out-of-distribution set. Examples of the out-of-distribution dataset post-Augraphy are shown in Figure 1.

Our new dataset is out-of-distribution with respect to the RVL-CDIP corpus in several ways: (1) a substantial portion of our new data is "born-digital",

whereas a large majority of RVL-CDIP is scanned physical documents; (2) a substantial portion of our data was created post-2006, with a large amount having been created within the past 10 years; (3) documents from our dataset are almost exclusively from industries and topics other than tobacco-related ones.

3 Experiments

We evaluate out-of-distribution performance by training several classifiers on the full RVL-CDIP training set and evaluating the confidence scores of the in-distribution test data versus the confidence scores obtained on the out-of-distribution set (in the case of OOD-*a*) and evaluating in-domain accuracy (in the case of OOD-*b*).

MODELS. We trained several image-based classifiers on the full RVL-CDIP training set. These models are VGG-16, ResNet-50, GoogLeNet, AlexNet, and LayoutLMv2. The accuracy scores that we achieved on the RVL-CDIP test set are shown in the second column of Table 1.

METRICS. For the OOD-*a* set, we use AUC to measure the seperability between confidence scores for in- and out-of-domain inputs. An AUC of 1.0 would mean perfect separation, and that classifiers are able to completely distinguish between in- and out-of-domain inputs based on prediction confidence. An AUC of 0.5 (indicating the two distributions are roughly overlap) would mean that classifiers are unable to distinguish between the two types of inputs. In contrast, the OOD-*b* set consists of data that do belong to the RVL-CDIP target categories, and so we evaluate performance on this set by measuring accuracy.

RESULTS. Table 1 charts in-distribution accuracy of each image classifier on the RVL-CDIP test set. Most models come very close to reported results in prior work. The AUC scores for OOD-*a* are relatively high, indicating models do reasonably well at discriminating between in- and out-of-domain data. When we use Augraphy to add scanner-like noise to our OOD data, the AUC score drops for all models except CLIP. Importantly, we observe a severe drop in accuracy on the OOD-*b* set.

Table 1. In-distribution accuracy compared OOD performance for each model.

Model	**ID Acc. (reported)**	**ID Acc. (achieved)**	**OOD-*a* AUC**	**OOD-*a* (Aug.)**	**OOD-*b* Acc.**
VGG-16 [4]	0.910	0.905	0.885	0.870↓	0.683
ResNet-50 [4]	0.911	0.900	0.874	0.865↓	0.575
GoogLeNet [4]	0.884	0.871	0.852	0.849↓	0.633
AlexNet [4]	0.900	0.885	0.874	0.869↓	0.607
LayoutLMv2 [5]	0.953	0.887	0.843	0.832↓	0.533

4 Discussion, Future Work, and Conclusion

The AUC scores on OOD-a indicate that the models are able to distinguish between in- and out-of-domain documents reasonably well. Adding scanner-like noise to the out-of-distribution test set pushes the AUC scores down for all of the supervised models, which seems to indicate that adding the noise to the out-of-distribution documents makes them more similar to the in-distribution RVL-CDIP documents. While the AUC scores are reasonable, we inspected the confidence scores returned by the models and found that most of the in-distribution test documents have max confidence scores of near 1.0 (e.g., 0.99 and even 1.0). Upon inspection, many of the out-of-distribution test documents are also near 1.0, but typically slightly lower (e.g., 0.985). This tells us that the models still predict the out-of-distribution with high in-distribution confidence.

The accuracy scores on OOD-b are substantially worse than the in-domain (ID Acc. in Table 1) counterparts (dropping by 28 points, on average), indicating a possible combination of the following factors: (1) RVL-CDIP is not diverse enough to endow trained models with the ability to generalize to new input distributions, (2) in order to achieve high in-domain and in-distribution accuracy, models need to overfit to RVL-CDIP. Nevertheless these findings highlight the importance of considering out-of-distribution inputs when evaluating document classifiers. Future work should investigate confidence calibration and regularization techniques in order to improve performance on the out-of-distribution documents. Additionally, it is worth investigating whether text-based classifiers are any more performant at the out-of-distribution detection problem.

In conclusion, the out-of-distribution detection problem is an important yet overlooked problem in the document classification field. This work describes work in progress toward developing and analyzing a companion evaluation dataset for the popular RVL-CDIP corpus.

References

1. Harley, A. W., Ufkes, A., Derpanis, K. G.: Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval. In: International Conference on Document Analysis and Recognition (ICDAR) (2015).
2. Larson, S., Mahendran, A., Peper, J., Clarke, C., Lee, A., Hill, P., Kummerfeld, J. K., Leach, K., Laurenzano, M. A., Tang, L., Mars, J.: An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In: Empirical Methods in Natural Language Processing (EMNLP) (2019).
3. The Augraphy Project. Augraphy: an augmentation pipeline for rendering synthetic paper printing, faxing, scanning and copy machine processes.
4. Afzal, M. Z., Kölsch, A. Ahmed, S. Liwicki, M.: Cutting the Error by Half: Investigation of Very Deep CNN and Advanced Training Strategies for Document Image Classification (ICDAR) (2017).
5. Xu, W., Xu, Y. Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., Zhou, L.: LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. In Proceedings of the 59th Annual Meeting of the Association for Computationsl Linguistics (ACL) (2021).

Text Classification Models for Form Entity Linking*

María Villota[1 [0000−0002−1457−4270]], César Domínguez[1 [0000−0002−2081−7523]], Jónathan Heras[1 [0000−0003−4775−1306]], Eloy Mata[1 [0000−0003−0538−4579]], and Vico Pascual[1 [0000−0003−3576−0889]]

Department of Mathematics and Computer Science, University of La Rioja, Spain
{maria.villota, cesar.dominguez, jonathan.heras, eloy.mata, vico.pascual}@unirioja.es

Abstract. Forms are a widespread type of template-based document used in a great variety of fields. The automatic extraction of the information included in these documents is greatly demanded due to the increasing volume of forms that are generated in a daily basis. However, this is not a straightforward task when working with scanned forms because of the great diversity of templates with different location of form entities, and the quality of the scanned documents. In this context, there is a feature that is shared by all forms: they contain a collection of interlinked entities built as key-value (or label-value) pairs, together with other entities such as headers or images. In this work, we have tackled the problem of entity linking in forms by combining image processing techniques and a text classification model based on the BERT architecture. This approach achieves state-of-the-art results with a F1-score of 0.80 on the FUNSD dataset, a 5% improvement regarding the best previous method.

Keywords: Entity Linking· Text Classification · Deep learning

1 Introduction

Forms are template-based documents that contain a collection of interlinked entities built as key-value (also known as label-value or question-answer) pairs [10], together with other entities such as headers or images. These documents are used as a convenient way to collect and communicate data in lots of fields, including administration, medicine, finance, or insurance. In these contexts, there is an enormous demand in digitising forms and extracting the data included in them [10]; the latter is a task known as form understanding [7]. The form understanding task is especially challenging when working with scanned documents due to the diversity of templates, structures, layouts, and formats that can greatly vary among forms; the different quality of the scanned document images; and, the scarcity of publicly annotated datasets [7].

* This work was partially supported by Grant RTC-2017-6640-7; and by MCIN/AEI/10.13039/501100011033, under Grant PID2020-115225RB-I00.

Form understanding consists of two steps: *form entity recognition* and *form entity linking* [7]. In the former, the spatial layout and written information of forms are analysed to localise the position of form entities and to identify them as questions (keys), answers (values), or other entities present in the form. In the latter step, the extracted entities are interlinked to understand their relationships. Several approaches have been published in the literature in order to solve both tasks. Usually, they try to take advantage of both semantic text features and layout information of the forms by combining different methods [1–3,6,8,10]. In this work, we have focused on the problem of entity linking in forms using a new method that combines computer vision and natural language processing techniques. Namely, we have proposed a new method for the task of entity linking in forms that combines image processing techniques and a text classification model based on a transformer architecture. For the text classification model, we have tested different architectures using transfer learning. The best model was obtained using the BERT architecture [4], which achieved a F1-score of 0.80 on the FUNSD dataset [7]; a 5% improvement regarding the best previous method. Finally, we have publicly released all the code and models developed in this work `https://github.com/mavillot/FUNSD-Entity-Linking`.

2 Methods

A summary of our method for form entity linking is provided in Figure 1. For each answer that is found on a given form, we identify a set of candidate questions based on their distance to the answer; and, subsequently, we concatenate the text of each candidate question with the text of the answer, and use a text classification model to determine if that combination of question and answer makes sense. Finally, if multiple questions are valid for the given answer, we take the one that is closer to the answer. For our text classification models, we have fine-tuned several transformer-based language architectures [9]; namely, BERT, DistilBert, Roberta, DistilRoberta, and LayoutLM. For fine-tuning the models, we replaced the head of each language model (that is, the last layer of the model), with a new head adapted to the binary classification task. Then, we trained the models for 6 epochs on the FUNSD dataset [7]. All the networks used in our experiments were implemented in Pytorch, and have been trained thanks to the functionality of the libraries Hugging Face, FastAI and Blur using the GPUs provided by the Google Colab environment.

3 Results

In this section, we analyse the results achieved with our method. We start by exploring the performance of the studied text classification model, see Table 1. The best model for all the evaluated metrics is obtained using the BERT architecture. This model clearly overcomes the rest by a large margin, it achieves a F1-score of 0.80; whereas, the rest of the models obtain values lower than 0.70. We additionally compare our proposed method with the existing algorithms

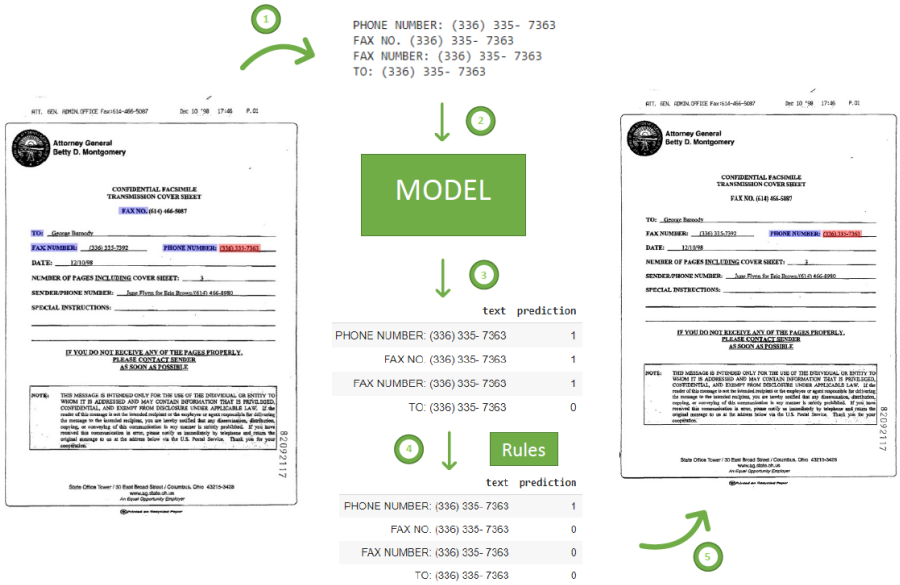


Fig. 1. Pipeline of the proposed method. (1) From an answer, a set of candidate questions are identified. (2) Each combination of candidate question-answer is fed to a text classification model that (3) identifies the valid combinations of question-answer. (4) If more than one combination is valid, the closest question is taken. (5) Finally, the results are returned.

available in the literature, see Table 1. From such a comparison, we find that the performance of our method using the BERT model improves all the existing approaches. In addition, we can notice that our method, independently of the employed text classification model, obtains a better mAP and mRank than the algorithms available in the literature. This proves the effectiveness of combining image processing techniques and deep learning models in this context.

4 Conclusion and Further work

In this paper, we have proposed a method for form entity linking based on the combination of image processing techniques and text classification models. This approach has achieved state-of-the-art results for form entity linking in the FUNSD dataset, and shows the benefits of combining deep learning models with algorithms based on the existing knowledge about documents when working in contexts where annotated data is scarce. As further work, we are interested in applying our method to more recent documents since the FUNSD dataset is formed by old documents, and also adapting our approach to work with documents written on different languages.

	mAP	mRank	F1-score
BROS [5]	-	-	0.67
Carbonell et al. [1]	-	-	0.39
FUDGE [3]	-	-	0.62
FUNSD paper [7]	0.23	11.68	0.04
DocStruct Model [10]	0.72	2.89	-
LayoutLM Word Level [8]	0.47	7.11	-
MSAU-PAF [2]	-	-	0.75
MTL-FoUn [8]	0.71	1.32	0.65
Sequential Model [8]	0.65	1.45	0.61
SPADE [6]	-	-	0.41
Ours-BERT	**0.87**	**0.49**	**0.80**
Ours-DistilBERT	0.79	0.79	0.68
Ours-DistilRoBerta	0.76	0.95	0.65
Ours-LayoutLM	0.79	0.81	0.69
Ours-RoBerta	0.77	0.94	0.66

Table 1. Comparison of our approach with existing methods for entity linking. In bold face the best results.

References

1. Carbonell, M., et al.: Named entity recognition and relation extraction with graph neural networks in semi structured documents. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 9622–9627 (2021)
2. Dang, T.A.N., et al.: End-to-end hierarchical relation extraction for generic form understanding. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 5238–5245. IEEE (2021)
3. Davis, B., et al.: Visual FUDGE: Form understanding via dynamic graph editing. In: Document Analysis and Recognition (ICDAR 2021). pp. 416–431. Cham (2021)
4. Devlin, J., et al.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics. pp. 4171–4186 (2019)
5. Hong, T., et al.: BROS: A pre-trained language model for understanding texts in document (2021), https://openreview.net/forum?id=punMXQEsPr0
6. Hwang, W., et al.: Spatial dependency parsing for semi-structured document information extraction. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 330–343 (2021)
7. Jaume, G., Ekenel, H.K., Thiran, J.P.: FUNSD: A dataset for form understanding in noisy scanned documents. In: Proceedings of the 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW). vol. 2 (2019)
8. Prabhu, N., et al.: MTL-FoUn: A multi-task learning approach to form understanding. In: Document Analysis and Recognition – ICDAR 2021 Workshops. pp. 377–388 (2021)
9. Razavian, A.S., et al.: CNN features off-the-shelf: An astounding baseline for recognition. In: CVPRW'14. pp. 512–519 (2014)
10. Wang, Z., et al.: Docstruct: A multimodal method to extract hierarchy structure in document for general form understanding. In: Proceedings of the Findings of the Association for Computational Linguistics (EMNLP 2020), pp. 898–908 (2020)

Augraphy: Data Augmentation for Document Images*

Alexander Groleau[1], Kok Wei Chee, and Stefan Larson[2]

[1] Sparkfish, Addison TX, USA
agroleau@sparkfish.com
[2] DryvIQ, Ann Arbor MI, USA

Abstract. This short paper introduces *Augraphy*, a Python package for data augmentation pipelines for document image analysis. Augraphy uses many different augmentation strategies to produce augmented versions of clean document images that appear as if they have been distorted due to noisy paper printing, faxing, scanning, or copy machine processes.

Keywords: Document Analysis · Denoising · Data Augmentation.

1 Introduction

Data augmentation is a widely used strategy in various areas of machine learning, including computer vision, image processing, natural language processing, and audio applications. Data augmentation can be used to generate new training samples data by applying transformations, rotations, noise, and other modifications to training data. Alternatively, data augmentation can be used to create noisy or challenging evaluation data from clean data, in which case it can be used for robustness testing or image denoising.

This paper introduces Augraphy,[3] a Python library for document image data augmentation. Augraphy uses highly-configurable pipelines to apply adjustments to document images to create augmented versions that appear old or noisy, as if they had been printed on dirty laser or inkjet printers, scanned by dirty or low-quality office scanners, or otherwise mistreated by real-world paper handling office equipment. This paper highlights some of the features of Augraphy, and demonstrates how it can be used effectively to produce challenging synthetic document denoising data.

2 Augraphy

Related Work. Several data augmentation libraries exist for image tasks. General purpose image augmentation libraries include Albumentations [3], Augmentor [1], Augly [2], and imgaug [5]. Augmentation techniques from these gen-

* Supported by Sparkfish LLC.
[3] https://github.com/sparkfish/augraphy

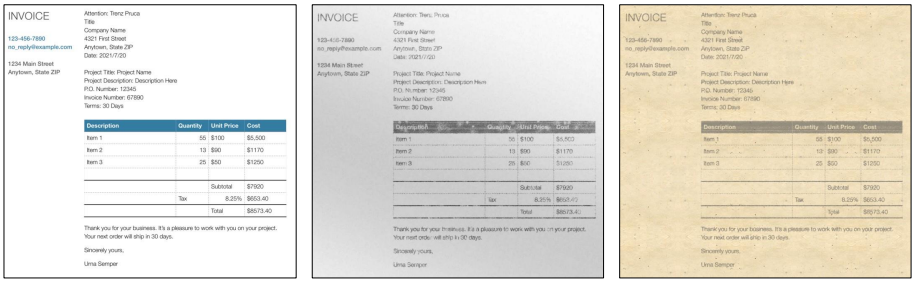


Fig. 1. Example input image (left) with two augmented versions (middle and right).

eral purpose libraries include rotations, translations, warps, and color transformations, yet none of these libraries provide augmentations targeted at imitating the types of transformations seen in document analysis corpora.

A notable exception is DocCreator [4], which is a document synthesizing tool that provides several transformation strategies as part of its synthesis pipeline. DocCreator's augmentations target imitating artefacts seen in historical (e.g., ancient or medieval) manuscripts, and hence do not address more modern causes of noise, such as noise introduced by document scanners. DocCreator is written in C++ and is meant to be used as a what-you-see-is-what-you-get tool; with no scripting or API interface, it is not easily amenable to being used in broader machine learning model development pipelines. In contrast, Augraphy is written in Python and has a simple interface to allow for seamless use with other Python libraries and data pipelines.

The Augraphy Package. Augraphy is a lightweight Python package. It is registered on the Python Package Index (PyPI) and can be installed using `pip install augraphy` . Augraphy requires only a few other commonly-used Python scientific computing or image handling packages in order to run, such as NumPy and Pillow. Augraphy has been tested on Windows, Linux, and Mac computing environments. Listing 1 shows how easy it is to get Augraphy up and running to create a straightforward augmentation pipeline and apply it to an image.

```
import augraphy; import cv2
pipeline = augraphy.default_augraphy_pipeline()
img = cv2.imread("image.png")
data = pipeline.augment(img)
augmented = data["output"]
```

Listing 1.1. Transforming an image with Augraphy.

Examples of output generated by Augraphy can be seen in Figure 1, which shows augmentations mimicking low printer ink and fuzzy, low-resolution text (middle image), and other paper surfaces (right image). We also show several of Augraphy's individual augmentation features in Figure 2. Importantly, these in-

dividual augmentation strategies can be composed together in an augmentation pipeline to create even more realistic looking, noisy output.

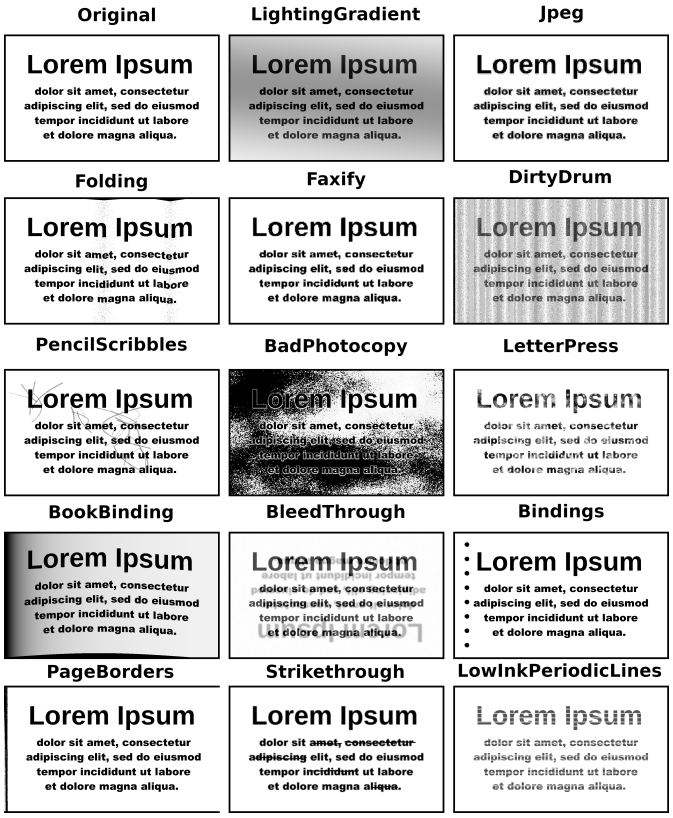


Fig. 2. Various individual augmentation types available in Augraphy. These individual augmentations can also be composed together.

Qualitative Case Study: Document Denoising. In this section we highlight the effectiveness of Augraphy by creating a new evaluation set for the task of document denoising. Document denoising is the task of removing noisy artifacts from a document image, and one recent dataset that has emerged for this task is the NoisyOffice dataset [6], which itself generated noisy versions of clean documents by applying several augmentations. However, both the original documents and the augmentations in NoisyOffice are quite limited, so it is natural to wonder if a model trained on NoisyOffice data can generalize to more diverse data inputs for the denoising task.

In Figure 3 we show example test inputs (left) to a convolutional autoencoder, which we trained on the NoisyOffice dataset. The model's outputs are

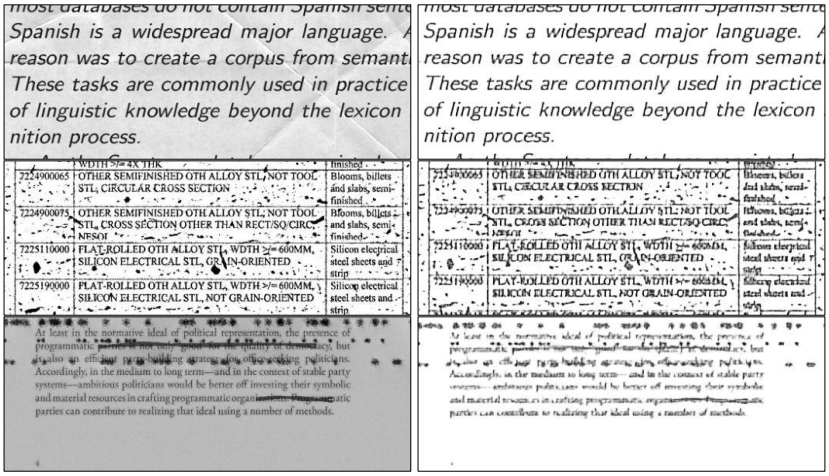


Fig. 3. Inputs (left) and outputs (right) to a denoising model. A NoisyOffice [6] sample is shown in the top row. Augraphy samples are shown in the bottom two rows.

shown on the right side of Figure 3. We see that the model does well on the NoisyOffice input (top row), but underperforms on data that was augmented by Augraphy (bottom two rows), showing that Augraphy's augmentations are effective at producing challenging testing data for analyzing the robustness of denoising models.

3 Conclusion

This paper introduces Augraphy, a new data augmentation package for document analysis tasks.

References

1. Bloice, M., Roth, P., Holzinger, A. Biomedical image augmentation using Augmentor. *Bioinformatics*, 35(21):4522-4524, 2019.
2. Papakipos, Z., Bitton, J. AugLy: Data Augmentations for Robustness. *arXiv preprint*, 2022.
3. Buslaev, A., Iglovikov, V., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020.
4. Journet, N., Visani, M., Mansencal, B., Van-Cuong, K., Billy, A. Doccreator: A new software for creating synthetic ground-truthed document images. *Journal of Imaging*, 3(4):62, 2017.
5. Jung, A., et al. Imgaug. 2020.
6. Castro-Bleda, M., España-Boquera, S., Pastor-Pellicer, J., Zamora-Martínez, F. The NoisyOffice Database: a corpus to train supervised machine learning filters for image processing. *The Computer Journal*, 63(11):1658-1667, 2020.

ShabbyPages, a Robust Corpus for Training Document Image Models*

Alexander Groleau[1], Stefan Larson[2], and Kok Wei Chee[3]

[1] Sparkfish, Addison TX, USA
[2] DryvIQ, Ann Arbor MI, USA
[3] ck91wei@gmail.com

Abstract. This paper presents the 'ShabbyPages' dataset, consisting of images of documents with realistic noise properties that result from standard office operations, such as printing, scanning, and faxing through old or dirty machines, degradation of ink over time, and handwritten markings. We designed this corpus to help train and evaluate machine learning methods – denoising, character recognition, and so on – with text documents. The dataset and scripts to reproduce it are available on GitHub. The dataset construction process is described, with attention paid to the decisions made.

Keywords: optical character recognition · image processing · denoising

1 Introduction

Inspired by the NoisyOffice dataset [1], we produced ShabbyPages as a way to help train, test, and calibrate computer vision machine learning algorithms designed for working with documents. We observed several limitations to the NoisyOffice dataset (namely, lack of diversity in font sizes and noise augmentations, as well as a lack of tables, graphics, form lines, etc.). Therefore, we were particularly interested in producing a dataset more appropriate for training general denoising models, so we built and leveraged the Augraphy [2] document image augmentation tool to produce noise for these images. ShabbyPages consists of clean-noisy document image pairs.

2 Creation

Development of the dataset occurred in stages. A team of researchers scoured the open internet for "born-digital" documents - PDFs that were created entirely electronically, rather than scans of existing printed documents. 600 documents were collected for review, representing categories such as government press releases, corporate financial communiques, informational brochures, and many others.

* Supported by Sparkfish LLC.

The *pdftoppm* tool was used to separate each document into its constituent pages, converting these to PNGs in the process, using GNU Parallel [4] to distribute the job across all available cores. `parallel pdftoppm pdfs/ pages/ -png -r 150` took 2 minutes to process 6202 pages on a Ryzen 9 5950x.

From there, we used Python's *cv2* library [3] to convert each document's color channels to grayscale, and once color data was removed, we generated and applied an Augraphy pipeline. After augmenting the images, we fit each to a standard 8.5"x11" Letter document at various DPI levels, cropping to fit where necessary. There were several possible resizing methods, but we opted to use one which simulates using a document scanner to capture an image. Code for all of these processes is available on GitHub [5].

3 ShabbyPages - An Augraphy Project

In our investigations, we came across precious few sources of ground-truthed document images. To aid the research community in making more, we're releasing this corpus and the code we used to produce it. Training denoising models requires a large quantity of noisy data and the original clean sources, and producing this is exactly what the Augraphy library was designed to facilitate. The Augraphy team has been hard at work for several months, improving the reliability, performance, and flexibility of the project, and we're proud that it's now mature enough to produce useful datasets.

Building the augmentation pipeline was straightforward: we took the default Augraphy pipeline and parametrized all the augmentations within it, keeping the parameters at the top of a file for easy adjustment. Supporting scripts were written to coordinate passing images through the Augraphy pipeline and saving them to new locations, dealing with different image resolutions, and so on. It was then possible to reproduce the noising process, so a cycle of testing was conducted where we generated noised images from the pipeline, determined properties of the output we didn't want to include in a published set, and accordingly tweaked the pipeline to no longer produce those effects. We wanted the ShabbyPages corpus to serve multiple functions – denoising, OCR, and so on – which constrained the degree of variation tolerated in the generating code.

Two primary classes of modifications to this code were considered, corresponding to a change in input constants and a restriction on certain combinations of augmentations. The Augraphy API enabled a tight feedback loop here, and made it possible to iteratively narrow in on a pipeline that generated the data we wanted. Driving Augraphy is largely a matter of developing some heuristics for unacceptable data and then telling Augraphy how to avoid producing that. This enables a workflow familiar to practitioners and researchers in this field, where edits are made to a build script and the rendering job is run, in a cycle.

4 Tuning the Pipeline

The default augmentation pipeline has already been tuned to produce realistic output, but the parameters required some careful tweaking to achieve our goals. Even so, there wasn't a great deal of work to do here beyond fiddling with constants to add or remove sources of variation, and to reduce the probability of certain augmentations being applied together.

Each augmentation in the Augraphy library was designed to reliably produce a specific effect on a document image, but care must be taken to ensure the effect is appropriate to the intended use. We intend for the first release of the ShabbyPages set to see use in training denoising models, so augmentation effects that make text impossible to read were rejected. Over a period of several weeks, we adjusted inputs to bring augmentations into the Goldilocks zone: not too heavy, not too noisy, just right. As an example, here's part of the diff between pipelines built on successive days:

```
← bleedthrough_intensity_range=(0.1, 0.2)
→ bleedthrough_intensity_range=(0.05, 0.15)

← bleedthrough_color_range=(0, 224)
→ bleedthrough_color_range=(32, 224)

← bleedthrough_alpha=random.uniform(0.1,0.2)
→ bleedthrough_alpha=random.uniform(0.05,0.1)
```

Augmentations in the Augraphy project are designed to produce changes mimicking those resulting from real-world processing of documents. A diverse array of effects are possible, including the incomplete deposit of ink on paper by a stamp or typeblock, staining on the page from a dirty print drum, the reduction in quality and artifacts introduced by faxing, and the shadow produced when scanning a book of a page curling away towards the binding. Several of these transformations are pairwise mutually incompatible, depending on the intended outcomes of the produced documents, and determining these pairs is a large part of the process when using Augraphy.

Here are a few bad pairings we found:

- BadPhotoCopy turns transparent clusters of noise into opaque black patches, and cannot be used with DirtyDrum.
- Bleedthrough can produce large dark regions which Letterpress interprets as text, applying unnatural blobs of noise.
- The combination of dithering, thresholding, and Gaussian noise makes BadPhotoCopy and Faxify destructively interact to produce unreadable text.

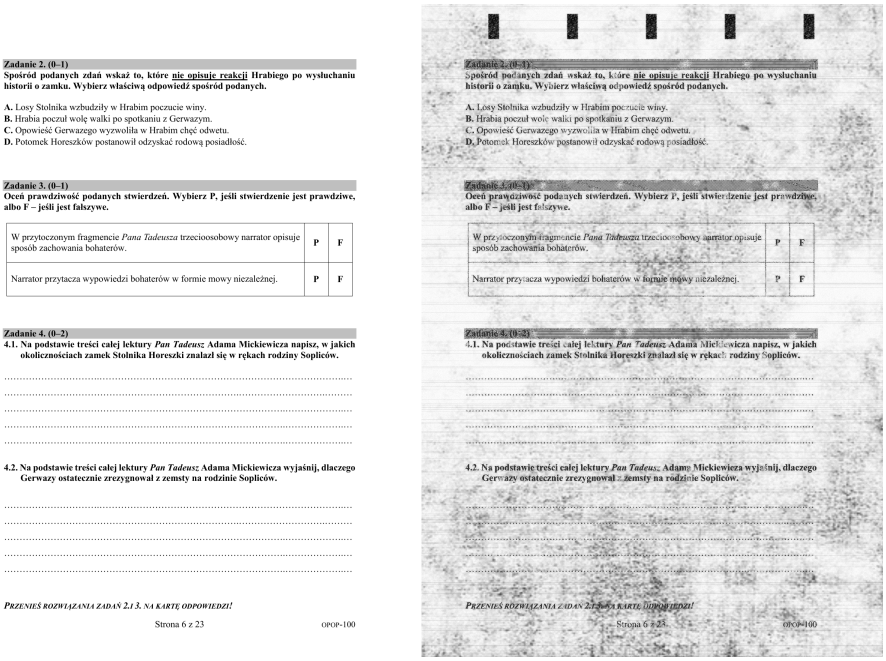


Fig. 1. Example input image (left) and Augraphy-augmented output (right).

5 Conclusion

Compilation and production of a collection of realistically noised document images with the Augraphy tool was straightforward. The resulting dataset contains a much broader variety of noise types, font sizes, document formats, and languages than the NoisyOffice set, and the initial release is publicly available for use in denoising, recognition, and classification tasks. Scripts to facilitate dataset production with Augraphy are published on GitHub, and both these scripts and the Augraphy project itself are being actively developed.

References

1. UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/NoisyOffice
2. Augraphy, An augmentation pipeline for rendering synthetic paper printing, faxing, scanning and copy machine processes, https://github.com/sparkfish/augraphy
3. OpenCV, https://opencv.org
4. Tange, O. (2021, July 22). GNU Parallel 20210722 ('Blue Unity'). Zenodo. https://doi.org/10.5281/zenodo.5123056
5. ShabbyPages, https://github.com/sparkfish/shabby-pages

Face detection in identity documents: an efficient security feature under challenging constraints

Lara Younes and Ahmad Montaser Awal

Research departement - ARIADNEXT, Rennes, France
lara.younes@ariadnext.com, montaser.awal@ariadnext.com

Abstract. Identity documents are usually highly protected with security features. The ID's holder photo is one of the basic security features as it certifies that the ID belongs to the right person. While face detection has been largely studied in the literature, there is still place for improvement specially in the context of mobile captured images of an ID document. This work tackles the face detection task in ID documents captured in the wild as an efficient security feature for identity verification. It presents a comparison of the mainly used face detectors in the literature. It highlights the remaining effort to be done for detecting faces in documents and suggests a framework for training a new face detector of better accuracy. It also gives recommendations of optimal data augmentation strategies to reach high precision for the detection of low effort photo-related frauds in documents. The result is a face detector for which the effectiveness is also reported over the MIDV-2020 dataset.

Keywords: Face detection · Face verification · Security feature · identity document · Deep CNNs · Data Augumentation · MIDV-2020

1 Introduction

Remote identity verification involves automatic analysis and verification of an identity document as well as its holder's identity. One fundamental task is the face localization both in identity documents and self portraits. It is thus important to use an accurate face detector for both challenging constraints and high resolution images. The face detection problem is not completely solved in non-studied contexts such as in identity documents, as shown in [1]. The security elements overlapping face zone or dark skinned faces captured in difficult conditions (low resolution, low contrast, $\cdots$) remain the main challenges. This work addresses the study of a face detector in documents. The generalization of the new detector to self-portrait is also taken into account to ensure a full secure remote identity verification system. In addition, the detector encompasses capacities allowing to detect identity fraud attempts.

2 Face Detection Framework

Training framework. A unified training and evaluation framework is setup to favor a fair comparison between the baseline systems and the newly trained face detector. The objective is to carry out a fine-tuning and easily reuse the

ImageNet-pre-trained checkpoints. In this framework, three pre-trained state-of-the art object detection architectures having comparable APs (average precision) over Image-Net dataset are used: EffecientDet [6] (ED-d0) and SSD architecture with two different backbones Mobilenet v1 [4] (sdMNv1) and Mobilenet v2 [5] (ssdMNv2). A baseline is computed over the public face detectors MTCNN [7] and the SSD one-stage detector available through the opencv toolbox (cvDnn).

Training and evaluation datasets. A private dataset of identity documents captured in the wild has been collected internally. The dataset includes: images with face zone occlusion (OVDs presence or light reflections) (IDs 1), difficult dark skinned faces (IDs 2), mixed simple faces with no challenges (Mixed) as well as selfies. False acceptances are evaluated over a dataset of real attacks whose types range from replacing the face by a silhouette, oval shapes, or a smiley or strikes over the face zone (figure 1a) . The dataset counts 100 fake identity documents. The public dataset MIDV2020 [2] is used along the experiments to validate the obtained results. Face detection is carried out in the rectified document images following the work in [3].

(a) Augumentations. (b) Real frauds.

Fig. 1: Examples of fakes examples and augmentations used to fight against.

Experiments over identity document images. The results shown in (table 1) highlights that current public detectors (baseline) are less efficient in the challenging context of identity documents. The three selected models have been fine-tuned using the training dataset. The face detectors confidence can be used to filter the detected zone by comparison to a threshold. Besides, a face zone is only retained as a true acceptance if it has an IOU (intersection over union) of 60% with the ground truth face zone. We can notice from table 1 that it is possible to specialize object detector for the task of face detection in the context of identity documents. Notably, the newly trained detectors outperforms the baseline in challenging conditions (IDs1, and IDs2). The true acceptance rate (TAR) is reported at a false acceptance rate (FAR) of 0.07.

In this initial training, we could achieve TAR in the range of 97% (table 1) for the newly trained architectures. Few works report the confidence threshold over which the TAR has been computed for the target FAR. We study this metric to understand the obtained rates. Challenging faces may be detected with lower confidence allowing their rejection depending on the final application requirements. This scenario can be inferred from the result of the cvDnn face detector over the IDs 2 set (table 1). A TAR of 99.05% is achieved at a confidence threshold of 0.8, meaning that the fake faces in our dataset, if detected by this method have been rejected due to a low confidence of detection. As mentioned earlier those off the shelf detectors have been trained over millions of images and have learned a fair representation of confidence between fairly visible faces and faces with hard challenging conditions.

	Baseline		Trained		
	MTCNN	cvDnn	ED-d0	ssdMNv1	ssdMNv2
IDs 1	62.6%	87.98%	96.2%	96.18%	94.77%
IDs 2	66.10%	39.44%	88.66%	88.13%	85.69%
Mixed	98.89%	99.05%	87.77%	99.5%	99.36%
Total	81.31%	91.03%	97.31%	97.48%	96.7%
Threshold	0.8	0.82	0.94	0.85	0.87

Table 1: Initial results using the raw dataset for training: TAR@0.07 FAR

As by nature OVDs reflection and low quality dark skinned faces resemble to occlusions highly represented in the train dataset, the trained model has learned them as highly confident faces zones. To cope with this problem, we train the models while following an augmentation approach. The train dataset is duplicated in such a way that every image is used in its original format and an augmented version of it. An augmentation (figure 1a) in this context is randomly one of the following types: an oval over the face zone, a rectangle shape covering the face from the top to the middle position of the mouth, brush stricks over the face or a sketch of a smiley generated through the google quickdraw[1] library covering the face. The augmentation types have been selected as they have a similar aspect as low-effort types of document frauds where the user tries to cover a part of the face (figure 1b) in the identity document.

The results of training with the augmented dataset are reported in (table 2). We can observe that the trained model could achieve an overall TAR ranging 98% for SSD architectures with mobilenet v1 & v2. The target threshold is 0.8 compared to 0.85 and 0.87 for Mobilenet v1 & v2 respectively. This shows that the augmented data allowed the model to learn a separation between the low effort fake face synthetically generated in the dataset with our augmentation method and the challenging occlusion over the faces in our train dataset.

The best selected model from our experiments (ssdMNv2) is compared with the public detectors over the 61421 video clips images of the MIDV2020. The results are reported in (table 2b). We notice an improvement of 1% compared to MTCNN. The analysed rejected faces in the dataset consisted of cases where the faces were completely covered by reflection over the document during the video capture. This shows, that all the models have reached a saturation for the detection of the faces in the rectifieds images of the MIDV2020 dataset. This proves the effectiveness of the trained model.

Experiments over selfies. Along the remote identification process, a user is required to submit a selfie to prove his identity by comparison to his face extracted from identity document. It is a high resolution image where the face covers a large space in the foreground of the image. In our experiments, we have not noticed a weakness of the public detectors for this scenario. The results presented in (table 2) are a non-regression proof that the newly trained model is accurate for the detection of faces in selfies of high resolution. In fact, the trained models reached a saturation for the detection of faces in the dataset of selfies included mixed ethnicity.

[1] https://quickdraw.withgoogle.com/data

	ED-d0	ssdMNv1	ssdMNv2
IDs 1	86.63%	95.56%	97.92%
IDs 2	56.20%	90.05%	93.54%
Mixed	97.77%	99.20%	99.17%
Total	90.8%	97.98%	98.59%
Threshold	0.9	0.8	0.8
Selfies	98.95%	99.05%	99.2%

(a) Results for the internal dataset.

	mtcnn	cvDnn	ssdMNv2
TAR	96.59%	94.53%	97.59%

(b) Tests over the 61421 video clips images of the MIDV2020 for the selected target confidence threshold.

Table 2: Results using the augmented dataset for training: TAR@0.07 FAR.

In this experimental study, we relied on a data analysis based approach that involves an alteration of datasets to increase the accuracy of existing ML models by focusing and working on the data to achieve the objectives. The result is a trained face detector allowing high accuracy for face detection along with low effort frauds detection over identity photo in document images. To isolate the impact of tuning the model parameters for a training from scratch, we used pre-trained weights of models archiving comparable APs on ImageNet dataset. The experiments showed that we can reach comparable accuracy with different networks by working on the dataset and its representation.

3 Conclusion

In this paper, we evaluated the state-of-the art open source face detectors on a private dataset of challenging mobile captures of identity documents. Since the work is meant to be used in a complete process of remote identification, we covered the evaluations over high resolution selfies of peoples as well. We adapted an approach in a data analysis fashion. Along the experiments we suggest an augmentation procedure of the training datatset allowing high accuracy's for face detection along with fake faces attempts in documents. We validated the trained model over the challenging dataset of documents MIDV2020. Future work will focus on using the detected faces for face verification application.

References

1. Bakkali, S., Luqman, M.M., Ming, Z., Burie, J.C.: Face detection in camera captured images of identity documents under challenging conditions. ICDARW 2019 (2019)
2. Bulatov, K., et al.: Midv-2020: A comprehensive benchmark dataset for identity document analysis. ArXiv (2021)
3. Chiron, G., Ghanmi, N., Awal, A.M.: Id documents matching and localization with multi-hypothesis constraints. In: ICPR 2020 (2021)
4. Howard, A.G., et al.: Mobilenets: Efficient convolutional neural networks for mobile vision applications (2017), http://arxiv.org/abs/1704.04861
5. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks (2019), http://arxiv.org/abs/1801.04381
6. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. CoRR **abs/1911.09070** (2019), http://arxiv.org/abs/1911.09070
7. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters (2016)

One-Shot classification of ID Documents

Florian Arrestier, Guillaume Chiron, and Ahmad Montaser Awal

ARIADNEXT By IdNow, Cesson-Sévigné, France
{florian.arrestier, guillaume.chiron, montaser.awal}@ariadnext.com

Abstract. This paper shows the generalization capabilities of EfficientNet for ID Document one-shot classification, namely the ability of the network to extend classification to unseen classes while registering only one sample per class. Various experiments with different training setups were done both on academic MIDV datasets and on an industrial one.

Keywords: Identity documents · One-shot classification · Deep-learning

1 Introduction

Classification, localization and analysis of Identity Documents (IDs) has drawn the community's attention in the last few years. The MIDV datasets (e.g. [1, 3]) publicly available provide a great value for benchmarking in the domain. This paper focus on classification. To the best of our knowledge, only few solution published in the literature are viable in an industrial context (i.e. accurate, scalable and maintainable), such as [2, 6] relying on handcrafted features coupled with filtering and matching techniques (e.g. SIFT, FLANN, RANSAC). More recently [4, 5] proposed modern techniques based on machine (deep)-learning. Despite being quite robust, all these approaches suffer from scalability issues. Supporting a growing number of models implies either performances losses, or noticable efforts for retraining and qualifying the models. To overcome theses limitations, we propose in this paper to specifically investigate the one-shot classification capabilities of modern neural network to classify IDs. This study has been conducted on the MIDV500/2020 datasets as well as on an private industrial dataset which supposedly overcome some limitations in terms of class/sample number and diversity encountered in the public datasets.

2 One-Shot Classification of IDs

The proposed approach for one-shot classification is detailed in this section and is composed of 3 phases, namely the training, the registration and the classification phases. In the following, all image samples that are passed to the classification network, including for training and registration phases, are considered pre-cropped, i.e the document of interest occupies most of the input image content. In production, this pre-crop operation is achieved through a dedicated network [5].

Network Training: In this work the lightweight EfficientNet-B0 [8] network is used as backbone, with pretrained weights on ImageNet[1]. Fine-tuning of the network is done on an industrial dataset with strong regularization from the Albumentations library[2] such as color/contrast variations, warping, or background swappings. Training is done using softmax loss with the logits layer being disregarded after training. Importantly, for performance reasons, input images are rescaled to a $112 * 112$ pixels resolution.

Registration of supported classes: Registration consists in passing a sample of a document class through the backbone network and saving the corresponding embeddings. The registration step is preferably performed offline and once per document class. Importantly, the number of supported classes is not fixed nor limited.

Classification: Once the backbone is trained, and reference samples are registered, the classification is straightforward. Classification is performed by computing the cosine similarity of the embeddings of a query sample with all of the registered reference ones, the winner class being the one with the highest similarity. This classification scheme is relatively cheap on modern hardware as it consists of only two tensors normalization + 1 matrix multiplication. The cheap cost of classification allows for further improvements such as providing multiple views (e.g rotated, warped or scaled) of each reference sample, improving classification performances.

3 Experiments and results

In this section 3 different experiments are conducted to evaluate the performance and the scalability of the proposed approach and to compare it to existing end-to-end solutions.

Generalization Performance: This first experience emphasizes the importance of the training dataset for generalization and scalability of the trained backbone to unseen classes. Indeed, 3 datasets are compared, the private and industrial AXT-Internal dataset (94 classes) and the Midv500 (50 classes) and Midv2020 (10 classes) public datasets. In Table 1, the network is trained on each of the three datasets and tested on the others. The first observation is that the network trained on 100% of the AXT-Internal dataset outperforms networks trained on any of the other datasets. Interestingly, the network trained on 50% of the AXT-Internal classes outperforms the network trained on the Midv500 dataset while containing roughly the same number of classes with 47 and 50 classes, respectively. Indeed, the network trained on the industrial dataset achieves an average accuracy of 97% on AXT-Internal 84 classes test Midv2020 datasets compared to 91.84% for the network trained on the Midv500 dataset. Finally, the network trained on 100% of the AXT-Internal train dataset matches the accuracy of the network trained on Midv2020 All split dataset on

[1] PyTorch Image Models: https://github.com/rwightman/pytorch-image-models

[2] Fast and Flexible Image Augmentations: https://github.com/albumentations-team/albumentations

all Midv2020 datasets. This result proves the generalization of this network to unseen data and the importance of the number of training classes.

Train \ Test		**AXT-Internal**	**Midv2020**				**Midv500**	
		TestSet	scan rotated	scan upright	photo	clip	* All	** TestSet
AXT-Internal	100%	**99.80**	100.00	100.0	**98.80**	**98.09**	**99.70**	**99.71**
	75%	96.20	100.00	100.0	97.70	97.55	99.57	99.54
	50%	92.52	100.00	99.70	95.80	96.96	99.22	99.29
	25%	84.20	98.60	98.00	91.10	94.87	98.59	98.87
Midv500*	All	71.21	99.80	97.70	94.30	96.21	99.96†	99.94†
Midv500**	TrainSet	49.00	79.70	79.10	58.40	58.19	88.79	87.55
Midv2020	All split	66.11	100.00†	100.00†	99.00†	99.75†	94.35	95.43

Table 1: Cross dataset one-shot classification accuracies. * filtered using [5] criterion, ** filtered using [7] criterion. † test samples overlap with training samples, values only serves as references of "ideal classification accuracy".

Scalability Performance: This experiment studies the performance of the trained backbone to an increasing number of classes. Figure 1 shows the classification accuracy on the Midv2020 photo dataset for 4 backbones trained on different number of classes of the AXT-Internal dataset, and for an increasing number of reference document classes (from 10 to 910). Figure 1 shows that increasing the number of training classes improves the robustness of the network to new classes. Interestingly, the network trained with 70 classes outperforms the one trained with 94 classes in the high number of reference regime.

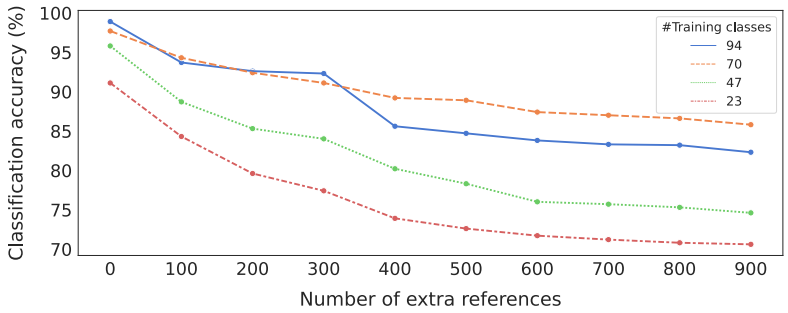


Fig. 1: One-shot classification accuracy on Midv2020 photo dataset (10 classes registered by default), with up to 900 other classes additionally registered.

End-To-End Comparison with State-of-The-Art: As mentioned in Section 2, in all training and previous experiments, the results are obtained by passing a pre-cropped image to the network. Up to this section, the pre-cropping of the documents was obtained using the annotated ground truth of each dataset.

In the following experiment, to ensure a fair comparison with existing end-to-end methods (i.e detection + classification), a similar approach to [5] is used to localize IDs and then the classification is performed only on the area detected when available. Table 2 shows the classification results of different approaches

on the different datasets. The proposed approach consistently outperforms or matches the accuracy of all other detection + classification methods on every datasets. Overall, the proposed approach achieves 98.51% classification accuracy on average across all datasets, far ahead of any existing methods.

Dataset	**AXT-Internal**	**Midv2020**				**Midv500**	
Method	TestSet	scan rotated	scan upright	photo	clip	* All	** TestSet
RFDoc [7]	-	-	-	-	-	-	93.46
SURF + Filters [2]	-	100.00 [3]	100.00 [3]	95.10 [3]	64.38 [3]	97.20 [6]	-
Beblid256 [3]	-	100.00	99.90	98.20	81.75	-	92.78 [7]
Beblid512 [3]	-	100.00	**100.00**	**98.70**	84.48	-	93.51 [7]
EffDet + Mnasnet [5]	94.98	-	-	-	-	93.91	-
EffDet + Ours	**99.34**	**100.00**	99.40	96.10	**95.78**	**99.50**	**99.47**

Table 2: Results of the proposed one-shot classification approach and other published classification methods, "-" stands for no results available.

Conclusion

In this paper, we showed how a classification network can be used for one-shot classification of identity documents of unseen classes during training. We showed that the quality of the training dataset along with the number of training classes greatly impact the generalization to unseen classes. Importantly, these preliminary results shows that even a small classification network used in such a setting outperforms methods based on handcrafted descriptors by a significant margin while still leaving the door for improvements.

References

[1] Vladimir V. Arlazarov et al. "MIDV-500: A Dataset for Identity Documents Analysis and Recognition on Mobile Devices in Video Stream". In: *CoRR* (2018).

[2] A. M. Awal et al. "Complex Document Classification and Localization Application on Identity Document Images". In: *14th IAPR ICDAR*. 2017.

[3] Konstantin Bulatov et al. "MIDV-2020: A Comprehensive Benchmark Dataset for Identity Document Analysis". In: *preprint arXiv:2107.00396* (2021).

[4] Alejandra Castelblanco et al. "Machine Learning Techniques for Identity Document Verification in Uncontrolled Environments: A Case Study". In: *MCPR*. Springer. 2020.

[5] Guillaume Chiron et al. "Fast End-to-End Deep Learning Identity Document Detection, Classification and Cropping". In: *ICDAR*. Springer. 2021.

[6] Chiron G. et al. "ID documents matching and localization with multi-hypothesis constraints". In: *25th ICPR*. IEEE. 2020.

[7] Daniil Matalov et al. "RFDoc: Memory Efficient Local Descriptors for ID Documents Localization and Classification". In: *ICDAR*. Springer. 2021.

[8] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *ICML*. 2019.

Combining Hadamard Matrix with Deep Learning for Sentence Embedding

Mircea Trifan[0000−0002−1594−5168], Bogdan Ionescu[0000−0002−6698−3404], and Dan Ionescu[0000−0002−7774−1347]

University of Ottawa, Ontario, Canada
{mircea, bogdan, dan}@ncct.uottawa.ca

Abstract. This paper presents a method of generating sentence embeddings with applications in Natural Language Processing (NLP). An emdedding maps a sentence to a vector of real numbers. Our approach uses: word embeddings, dependency parsing, Hadamard matrix with spread spectrum algorithm and a deep learning neural network trained on a corpus. The dependency parsing labels are associated with rows in a Hadamard matrix. The word embeddings are stored at corresponding rows in another matrix. Using the spread spectrum encoding algorithm the two matrices are combined into a single unidimensional vector. This embedding is then fed to a neural network achieving 80% accuracy while the best competition score from the SEMEVAL 2014 is 84%.

Keywords: Sentence Embedding · Hadamard Matrices · Deep Learning.

1 Introduction

Sentence embedding is an important NLP technique that maps a sentence to a vector of real numbers. It is used for many NLP related tasks like sentence similarity and Natural Language Inference (NLI). This paper is an extension of a previous article: [7]. Here, we use deep learning on top of the previous embedding algorithm in order to improve the accuracy. Also, we focus on NLI instead of similarity. This paper is organized as follows. Section 2 describes related work in regards to sentence embeddings and NLI. Section 3 is reviewing the previous Hadamard sentence embeddings, section 4 describes the new deep learning addition and section 4 concludes.

2 Related Work

The SEMEVAL 2014 competiton introduces the text entailment and similarity tasks within the Sentences Involving Compositional Knowledge (SICK) corpus [1]. The corpus comprises 10,000 sentence pairs annotated with similarity and inference labels: entails, contradicts and neutral.

A larger corpus than SICK is the Stanford Natural Language Inference (SNLI). It contains more than 570k annotated premise and hypothesis sentences

with the entailment, contradiction and neutral categories. [2] presents a synopsis of published results.

The top system described in the paper [6] has 340 million parameters and achieves a state of the art accuracy of 92.1 on the test dataset. It uses Multi-Task Learning (MTL) to transfer knowledge between NLP tasks. A novel transformer based architecture with a conditional attention mechanism is described.

[3] uses the SNLI corpus to train universal sentence representations. A wide range of transfer tasks benefit from NLI training: sentiment analysis, question answering, product reviews, subjectivity/objectivity, opinion polarity, SICK dataset tasks for both entailment (SICK-E) and semantic relatedness (SICK-R) and Semantic Textual Similarity. For SICK-E entailment task this system reports an accuracy of 86.3%. For the sentence encoder they use: LSTM, GRU, BiLSTM with mean/max pooling, Self-attentive network and Hierarchical ConvNet. We use a similar training scheme for the downstream neural network. However, for the sentence encoder we use our Hadamard encoder augmented with 2 linear and ReLU layers to reduce the dimensionality.

3 Hadamard Sentence Embeddings

For word embeddings we use Word2vec [5] computed on a very large corpus of more than 100 billion words. Each word vector in the vocabulary has a length of 300.

Dependency parsing overlays a grammatical structure graph on top of the words within a sentence. The first sentence of SEMEVAL 14: SICK test annotated is presented at top of Figure 1. The dependency graph is in green while the parse labels in orange. In the implementation, Stanford Dependency Parser [4] was used to parse the sentences from the SICK corpus.

The Hadamard square matrix [8] is composed of +1 or -1 entries. Higher order Hadamard matrices can be recursively obtained from lower order Hadamard matrices. The rows are mutually orthogonal. Due to their orthogonality property, Hadamard matrices are used in spread spectrum technologies like CDMA. Multiple signals are spread over a common frequency band. During despreading the individual signals are reconstructed from the common signal.

Figure 1 depicts the Hadamard embedding of a sentence. Consider the sentence on top of Figure 1. It is annotated with a dependency parsing graph consisting of parse labels and directed arcs from each governor word to it's corresponding dependent word. One such relation is *advmod(playing, outdoors)*. The parse labels are associated with specific rows in the Hadamard matrix. For each arc, two entries are reserved in the Hadamard matrix one for governor and one for dependent. Thus the grammar aspects are captured by the Hadamard matrix: H(128x128) while the matrix D(128x300) consists of the word embeddings for the words in the deep parsing relation. Both matrices H and D are combined using the spreading algorithm in a single vector E(1x38400) that represents the sentence embedding used as input for the upstream deep learning

neural network. The neural network learns from the SICK corpus to account for linguistic knowledge.

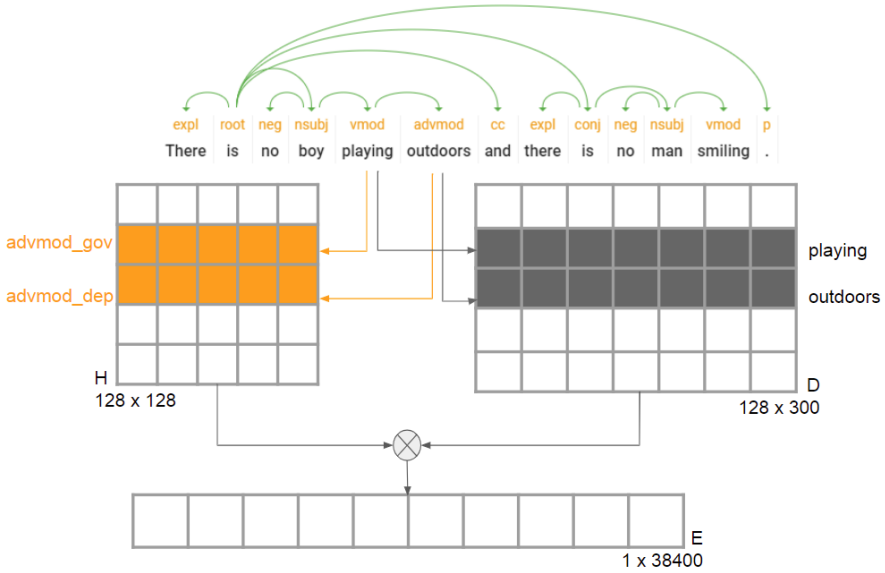


Fig. 1. Hadamard Embedding of a Sentence.

4 Deep Learning Architecture

The architecture is presented in Figure 2. There are two sentence encoders for the premise and hypothesis sentence pairs in the SICK dataset. Each sentence encoder starts with a Hadamard embedding described previously that has its dimension reduced by two pairs of Linear 1 and Linear 2 and coresponding ReLU 1 and ReLU 2 layers, with a final output of a 5000 size vector. Following the approach in [3], a large input vector if formed by concatenating: both sentence embeddings, the absolute value of the difference and the element wise product and fed to the input of Linear layer 3 and correponding ReLU 3. Another Linear layer 4 is further reducing the dimension from 500 to 3 nodes processed by the final Log Softmax. The final output is the computed embedding label.

We used a Hadamard matrix of order 128 that is large enough to capture dependency parse labels in the SICK corpus. The learning rate is 0.001.

The algorithm is evaluated on the SICK dataset for entailment (SICK-E) while in the previous article [7] we explored semantic relatedness (SICK-R). The results of SEMEVAL 2014 competition for NLI are available online at [1]. The best system reports an accuracy of 84%, while our system achieves an accuracy of 80% on the testing dataset (82.2% for the training dataset).

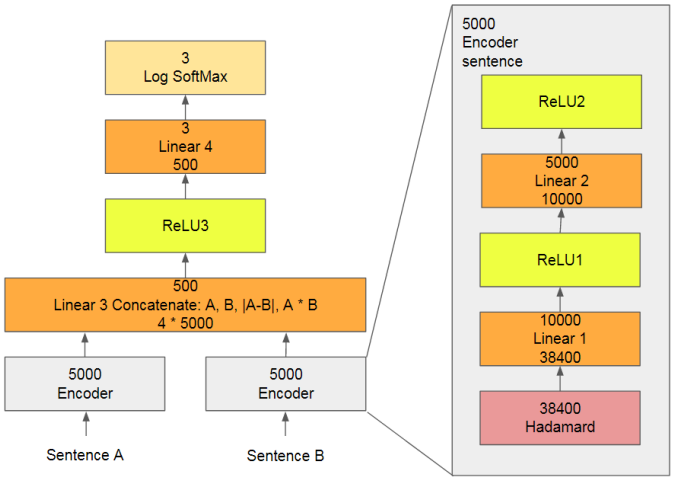


Fig. 2. NLI Training Architecture

5 Conclusion

A previous Hadamard based sentence encoding method is extended with a deep learning neural network and evaluated for the NLI task with the SICK corpus. It achieves 80% accuracy using only linear, ReLU and log SoftMax layers. One advantage of our technique is that it allows for encoding of variable length sentences within a vector of real number. For the future, we plan to test our algorithm on the SNLI corpus.

References

1. Semeval-2014 task 1: Entailment subtask. https://alt.qcri.org/semeval2014/task1/index.php?id=results, accessed: 2022-04-12
2. The stanford natural language inference (snli) corpus. https://nlp.stanford.edu/projects/snli, accessed: 2022-04-12
3. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. (2017)
4. de Marneffe, M.C., Manning, C.D.: Stanford dependencies manual. (2008)
5. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. (2013)
6. Pilault, J., Elhattami, A., Pal, C.: Conditionally adaptive multi-task learning: Improving transfer learning in nlp using fewer parameters & less data. (2021)
7. Trifan, M., Ionescu, B., Gadea, C., Ionescu, D.: A graph digital signal processing method for semantic analysis. (2015)
8. Wang, R.: Introduction to Orthogonal Transforms: With Applications in Data Processing and Analysis. Cambridge (2012)

Evaluating Table Structure Recognition: A New Perspective

Tarun Kumar and Himanshu Sharad Bhatt

American Express AI Labs, India
{tarun.kumar16, himanshu.s.bhatt}@aexp.com

Abstract. Existing metrics used to evaluate table structure recognition algorithms have shortcomings with regard to capturing text and empty cells alignment. In this paper, we build on prior work and propose a new metric - TEDS based IOU similarity (TEDS (IOU)) for table structure recognition which uses bounding boxes instead of text while simultaneously being robust against the above disadvantages. We demonstrate the effectiveness of our metric against previous metrics through various examples.

Keywords: Evaluation metric · Table Structure Recognition · Intersection over Union (IOU).

1 Introduction

A huge amount of information flows through enterprise documents; thus, it is imperative to develop efficient information extraction techniques to extract and use this information productively. While documents comprise multiple components such as text, tables, figures etc.; tables are the most commonly used structural representation that organize the information into rows and columns. It captures structural and geometrical relationships between different elements and attributes in the data. Moreover, important facts/numbers are often presented in tables instead of verbose paragraphs. For instance, tables in financial domain are a good example where different financial metrics such as "revenue", "income" etc. are presented for different quarters/years. Extracting the content of a table into a structured format (csv or JSON) [1], [2], [3] is a key step in many information extraction pipelines.

Unlike traditional machine learning problems where the output is a class (classification) or number (regression), the outcome of a table parsing algorithm is always a structure. There needs to be a way to compare one structure against another structure and define some measure of "similarity/distance" to evaluate different methods. A number of metrics quantifying this "distance" have been proposed in literature and multiple competitions. Existing metrics evaluates the performance of table parsing algorithms using the structural and textual information. This paper presents the limitation of existing metrics based on their dependence on the textual information. We emphasize that textual information introduces additional dependency on the OCR (text detection/recognition),

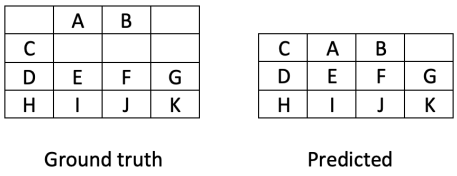


Fig. 1. Original table and an example prediction for the same. For Adjacency relation (Text), the characters can be considered as representing the text inside cells. For Adjacency relation (IOU), characters can be considered as labels representing cells.

which is a separate area in itself and should not be included in evaluating how good is the detected table structure. This paper presents a "true" metric which is agnostic to the textual details and accounts only for the layout of cells in terms of its row number/column number and bounding box.

2 Existing Metrics in Table Parsing

Two of the existing metrics are adjacency relation set-based F1 scores with different definitions of the set. They break and linearize the table structure into two dimensions, one along the row and one along the column. Adjacency Relation (Text) [2] computes pair-wise relations between non-empty adjacent cells and the relation is considered correct only if the direction (horizontal/vertical) and text of both the participating cells match. It does not take into account empty cells and multi-hop cell alignment. Adjacency Relation (IOU) [1] is a text-independent metric where original non-empty cells are mapped to predicted cells by leveraging (multiple) IOU thresholds and then adjacency relations are calculated. This metric takes a weighted average of the computed F1-scores at different IOU thresholds {0.6, 0.7, 0.8, 0.9}. Finally, the predicted relations are compared to the ground truth relations and precision/recall/F1 scores are computed.

The third metric considers the structure as a HTML encoding of the table. In this representation, the table is viewed as a tree with the rows being the children of the root $< table >$ node, and cells being the children (represented by $< td > [text] < /td >$) of the individual rows. A Tree edit-distance (TEDS) metric [6] is proposed which compares two trees and reports a single number summarizing the similarity.

While there are other metrics used in literature such as BLEU-4 [5] (which is more language based), this paper only considers the above three most widely used metrics for evaluating the performance of table structure recognition.

3 Proposed Metric

This paper highlights the limitations of the previous metrics and also proposes a new metric, Tree-Edit-Distance Based Similarity with IOU *(TEDS-IOU)*, for evaluating table structure recognition algorithms. The paper also demonstrates how *TEDS-IOU* addresses the limitations of existing metrics.

Table 1. Existing metrics in literature and their limitations

Metric	Limitations
Adjacency Relation (Text)	Doesn't handle empty cells, misalignment of cells beyond immediate neighbours & text dependent
Adjacency Relation (IOU)	Doesn't handle empty cells, misalignment of cells beyond immediate neighbours
TEDS (Text)	Text dependent but less strict due to Levenshtein distance

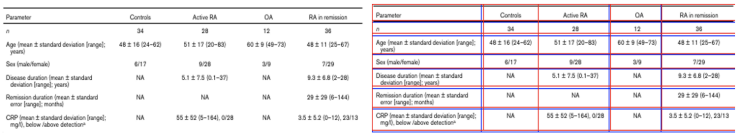
Table 1 describes the limitations of the commonly used metrics in table structure recognition literature. For example, in figure 1, even though the predicted table missed one entire row and 4 empty cells, in terms of adjacency relations, the only extra relation in the predicted table is the {C, A, Horizontal}, where 'Horizontal' is the direction of relation. This only affects precision but the recall is still 100% which clearly should have been penalised. Also, in the case of the IOU based metric, lets assume label mapping, i.e. cell represented by "C" in ground truth is a mapped to the "C" cell in predicted table using IOU thresholds. We still have that same extra relation {"C", "A", Horizontal}, where 'Horizontal' is the direction, which demonstrates the inability to capture empty cells and misalignments. We should note that metric is still better then the text-based version, since it does not rely on comparing text. Accurately detecting and recognizing text (OCR) is a separate field in itself, while in table structure recognition, we are primarily interested in localizing the cell boundaries and assign text to them.

TEDS (Text) metric solved the shortcomings of previous metrics with regard to empty cells and multi-hop mis-alignments [6]. In TEDS, all cells, with or without text are considered, thereby also including empty cells as part of computation. So, TEDS (text) will penalise the absence of a row and all the alignment mismatches when comparing ground truth table against predicted table in figure 1. But it computes the edit distance between cells' texts as compared to the exact match in Adjacency Relation (Text).

Table structure recognition algorithms aim at predicting the location (bounding boxes) of cells and their logical relation with one another, irrespective of the text in the cell. Therefore, the evaluation metric should not penalize an algorithm for inaccuracies in text. With this observation, this paper propose TEDS (IOU) which replaces the string edit distance between cells' text with the IOU distance between their bounding boxes. This effectively, removes dependency on text or OCR, while also preserving the benefits of the original TEDS (text) metric. Specifically, we compute TEDS (IOU) as follows: cost of insertion & deletion operations is 1 unit; while substituting a node n_s with n_t - cost of edit is 1 unit if either n_s or n_t is not $< td >$, cost of edit is 1 unit if both n_s & n_t is $< td >$ and the column span or row span of n_s & n_t is different, otherwise, cost of edit is $1 - IOU(n_s.bbox, n_t.bbox)$. Finally,

$$TEDS\_IOU(T_a, T_b) = 1 - \frac{EditDistIOU(T_a, T_b)}{max(|T_a|, |T_b|)} \tag{1}$$

TEDS (IOU) $\in [0, 1]$, the higher the better. $|.|$ denotes cardinality. IOU distance ($IOU_d = 1 - IOU$) being a Jaccard index [4], is a metric as it satisfies:



Parameter	Controls	Active RA	OA	RA in remission
n	34	28	12	36
Age (mean ± standard deviation [range]; years)	48 ± 16 (24–62)	51 ± 17 (20–83)	60 ± 9 (49–73)	48 ± 11 (25–67)
Sex (male/female)	6/17	9/28	3/9	7/29
Disease duration (mean ± standard deviation [range]; years)	NA	5.1 ± 7.5 (0.1–37)	NA	9.3 ± 6.8 (2–28)
Remission duration (mean ± standard error [range]; months)	NA	NA	NA	29 ± 29 (6–144)
CRP (mean ± standard deviation [range]; mg/l; below /above detection[a]	NA	55 ± 52 (5–164), 0/28	NA	3.5 ± 5.2 (0–12), 23/13

(a)

Parameter	Controls	Active RA	OA	RA in remission
n	34	28	12	36
Age (mean ± standard deviation [range]; years)	48 ± 16 (24–62)	51 ± 17 (20–83)	60 ± 9 (49–73)	48 ± 11 (25–67)
Sex (male/female)	6/17	9/28	3/9	7/29
Disease duration (mean ± standard deviation [range]; years)	NA	5.1 ± 7.5 (0.1–37)	NA	9.3 ± 6.8 (2–28)
Remission duration (mean ± standard error [range]; months)	NA	NA	NA	29 ± 29 (6–144)
CRP (mean ± standard deviation [range]; mg/l; below /above detection[a]	NA	55 ± 52 (5–164), 0/28	NA	3.5 ± 5.2 (0–12), 23/13

(b)

Fig. 2. (a) is a table from PubTabNet dataset. In (b), red lines denote the predicted structure and blue lines depict the true structure.

1. $IOU_d(A, B) = 0 \iff A = B$ *Identity*
2. $IOU_d(A, B) = IOU_d(B, A)$ *Symmetry*
3. $IOU_d(A, C) <= IOU_d(A, B) + IOU_d(B, C)$ *Triangle Inequality*

To demonstrate the effectiveness of the proposed TEDS (IOU) metric, we compute the all four metrics for the predicted table in figure 2(b). In the example above, we had known OCR issues where it was unable to recognize the ± symbol (it got recognized as +) and all the cells with"NA" were detected as empty. Adjacency Relation (Text) got a very poor score of 13.7 F1 due to the exact text match constraint. Adjacency Relation (IOU), being text independent, is more robust and achieves a Weighted Avg. F1 of 59.8. TEDS (text) matches text through edit distances, therefore, for it, only the "NA" cells gave high edit distance (of 1) and it scores 71.6 on this table. TEDS (IOU) being text independent and computing the IOU distance between cells, assigns a higher score of 80.6 which seems to be the most representative one of the prediction.

4 Discussion & Future Work

We proposed a new metric for table structure recognition and demonstrated its benefits against existing metrics. As future steps, we plan to compare these metrics across different datasets and models. A possible extension of this work can be to introduce different thresholds for the IOU as in Adjacency Relation (IOU), instead of using absolute numbers.

References

1. Gao, L., Huang, Y., Déjean, H., Meunier, J.L., Yan, Q., Fang, Y., Kleber, F., Lang, E.: Icdar 2019 competition on table detection and recognition (ctdar). In: 2019 ICDAR. pp. 1510–1515. IEEE (2019)
2. Göbel, M., Hassan, T., Oro, E., Orsi, G.: Icdar 2013 table competition. In: 2013 ICDAR. pp. 1449–1453. IEEE (2013)
3. Jimeno Yepes, A., Zhong, P., Burdick, D.: Icdar 2021 competition on scientific literature parsing. In: ICDAR. pp. 605–617. Springer (2021)
4. Kosub, S.: A note on the triangle inequality for the jaccard distance. Pattern Recognition Letters **120**, 36–38 (2019)
5. Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., Li, Z.: Tablebank: A benchmark dataset for table detection and recognition. arXiv preprint arXiv:1903.01949 (2019)
6. Zhong, X., ShafieiBavani, E., Jimeno Yepes, A.: Image-based table recognition: data, model, and evaluation. In: ECCV. pp. 564–580. Springer (2020)

Short Paper Booklet

15th IAPR International Workshop on Document Analysis Systems

22 — 25 May 2022

DAS 2022 - La Rochelle

La Rochelle Université

Informatique Image Interaction
La Rochelle Univ.

IAPR