Team Members:

Joseph Cappadona (PennKey: `jcapp`; Email: `jcapp@seas.upenn.edu`)

Assigned Project Mentor:

Anant Maheshwari

Team Member Contributions:

Team Member	Contributions
Joseph Cappadona	*

Code Submission:

Code can be found here: `https://github.com/josephcappadona/cis520-final-project`

Abstract

Information extraction is important to any data pipeline. But automating information extraction can be difficult, and there are many instances where manual data collection and entry is the only option. A robust document recognition and information extraction engine would vastly improve the rate at which certain types of data could be collected. We have taken inspiration from a variety of previous works on structured and unstructured information extraction in an effort to build a machine learning model that can handle a wide variety of document types and data formats. We combine a traditional bag-of-words model with textual clustering and alignment and layout analysis to generate a large number of document features. We found the best performance with tree-based classifiers, achieving, for the Ghega patent data set and Ghega data-sheet data set, generalization errors of 6.41% (Random Forest, n=31, max_depth=26) and 3.23% (Decision Tree, max_depth=28), respectively.

1 Introduction

Information extraction via document classification and recognition can be used to extract data from a wide variety of document types, such as from invoices [1], memos, and advertisements; from Heads-Up Diplays (HUDs), like those used in mission simulators [2] and on sports broadcasts; and from web pages [3]. Due to the unstructured nature of most document types, information extraction algorithms need to consider a large number of both text-level features (words, data types) and document-level features (relative positioning and alignment of text). The goal in this paper is to draw from past research into structured and unstructued information extraction to develop a robust set of features which allows for high quality, generalizable information extraction.

2 Related Work

In [2], the authors built an image processing application with pattern recognition to analyze the HUD of Flight Test Campaigns and extract information important to the analysis of the campaign, such as aircraft position parameters, aircraft configuration information, tracking mode, and time. Major limitations included its naive text recognition system, and its poor image processing efficiency and text recognition accuracy, which were both largely due to the poor quality of the extracted HUD images. Additionally, this type of analysis is only effective for static, structured documents.

[4] approached information extraction from unstructured documents by utilizing automated context-free grammar learning and alignment-based learning. This type of grammar learning provides a dynamic, adaptive approach to information extraction that is necessary for schema learning. Similarly, [5] implemented layout analysis in order to learn common layout schemas for unstructured documents.

In [1], the authors built an unstructured document recognition system for business invoice processing centered around a bag-of-words approach that attempts to capture common layout and text features. They experimented with Naive Bayes, Multiclass (One-vs-All) Logistic Regression, and Multiclass (One-vs-All) Linear SVM. They were able to achieve 8.81% training error and 13.99% test error (through SVM) over a data set of 97 raw invoice images obtained from the internal testing

library of the Oracle Corporation. They found that in all situations $L2$ regularization outperformed $L1$ regularization.

Comprehensive surveys such as [6, 7] demonstrate the potential for the use of deep neural networks for information extraction and document classification. However, these approaches rely on large repositories of documents, which can be problematic for certain use cases.

3 Data Set

The primary data set utilized was the Ghega-dataset, "a dataset for document understanding and classification" (`http://machinelearning.inginf.units.it/data-and-tools/ghega-dataset`). This data set consists of 110 data-sheets of electronic components in English and 136 patents in 7 different languages. For each document, a pre-processed, 300 DPI image is provided, along with a blocks CSV and a groundtruth CSV. The blocks CSV contains all text blocks detected via OCR. The groundtruth CSV contains only the text blocks in the blocks file that correspond to desired pieces of information and the associated labels (if they exist). See Appendices A and B for examples of each document.

Many other types of data sets would have been useful to test with, especially those described in the Introduction, however finding labeled data sets for these applications is difficult. With more time, we would have hand-labeled small sets of HUD data, advertisement data, and web page data to test on.

4 Problem Formulation

Given a set of documents X with associated sets of labels Y, our goal is to train a classifier h to, given a new document instance x, pull out features of interest f_i and classify them into categories $\widehat{c_i}$. Accordingly, we will attempt to minimize the loss

$$L = \sum_{f_i \in x} \mathbf{1}(h(f_i), c_i)$$

or, in other words, a loss of 1 is incurred for each feature f_i which is misclassified. Since feature generation and selection is deterministic, we can be assured that we will generate the same features f_i from each x on each pass over the data.

5 Algorithms

Text Extraction: Although feature and label data were provided in the Ghega-dataset, text extraction was implemented using pytesseract, a Python wrapper for the Tesseract Optical Character Recognition (OCR) engine.

Text Cluster Formation: In order to provide a more robust document model, extracted text was clustered using Density-Based Spatial Cluster of Applications with Noise (DBSCAN). This allows for the generation of more complex feature sets than would otherwise be possible. See Appendices A and B for examples of the results of DBSCAN text clustering.

Feature Generation: Feature generation consisted of analyzing the content in blocks as well as relative positioning and content between blocks. Each text block was first cleaned by replacing particular data types with generic representations ("MONEY", "DATE", "TELE", "EMAIL", "NUM", "ALPHANUM"). Next, for each text block, the following features were generated: `num_words`, `num_chars`, `is_text`, `is_numeric`, `is_alphanumeric`, `cluster`, `x`, `y`, `w`, `h`, `hAlign_WORD`, `vAlign_WORD`, `sameCluster_WORD`, `vecTo_WORD_x`, `vecTo_WORD_y`. Most of these features are self-explanatory, but `cluster` corresponds to the document cluster (generated with k-Means cluster, k=50) that the text block is closest to, and for the features with `WORD` in them, `WORD` was replaced with each word in the block's text cluster (the text cluster generated using DBSCAN) for which the feature is true. That is, `vAlign_WORD` (`hAlign_WORD`) is true for each word that a text block is vertically (horizontally) aligned with. `sameCluster` is true for each word that a text block shares a text cluster with. And `vecTo_WORD_x` and `vecTo_WORD_y` correspond to the x and y components of the vector pointing from the text block to each word that is in the same text cluster. Additionally, in order to minimize the number of features generated, we only generated a feature corresponding to `WORD` if that word was in the 750 most common words throughout the data. The goal of these features, particularly the features related to text clusters, was to better capture spatial features than a bag-of-words approach with elementary layout analysis. This approach resulted in approximately 4000 features for both data sets.

Feature Selection: To further narrow down the number of features, the relative entropy between the feature and label distributions was computed. In particular, the Kullback-Leibler (KL) Divergence was computed for the distributions $p(x_i)p(y)$ and $p(x_i, y)$, for each feature x_i. The 2000 highest scoring features were kept.

Learning Algorithms: Several machine learning algorithms were applied to the data. The algorithms that yielded particularly promising results were Decision Trees (and other tree-based classifiers), Logistic Regression, and SVM.

6 Experimental Design and Results

For Ghega-dataset Patents:

Algorithm	**err_train (%)**	**err_test (%)**	**Hyperparameter**
Decision Tree	3.04	9.03	d=26
Bagging	0.05	7.09	DT
Random Forest	0.04	6.41	n=31
Logistic Regression	4.41	18.07	L1
SVM	1.77	21.46	-
kNN	13.37	19.71	k=5

The tree-based classifiers (Decision Tree, Bagging-DT, Random Forest) yielded the lowest generalization error by far, with the lowest being Random Forest's 6.41%. Logistic Regression, SVM, and k-Nearest Neighbors did not generalize well, as shown by the generalization errors of 18.07%, 21.46%, and 19.71%, respectively.

For Ghega-dataset Data-sheets:

Algorithm	**err_train (%)**	**err_test (%)**	**Hyperparameter**
Decision Tree	0.52	3.23	d=28
Bagging	0.10	3.70	DT
Random Forest	0.07	3.57	n=31
Logistic Regression	1.92	9.74	L1
SVM	0.20	28.90	-
kNN	6.04	7.87	k=5

Similarly to the patent data, the tree-based classifiers yielded the lowest generalization errors, with a depth-28 Decision Tree yielding the lowest generalization error with 3.23%. Also similar to the patent data, Logistic Regression and SVM did not generalize well. Unlike the patent data, however, k-Nearest Neighbors classifiers did surprisingly well with only 7.87% generalization error.

7 Conclusion and Discussion

For the both data sets, the tree-based classifiers substantially outperformed all other classifiers. We think that this is most likely due to the fact that the features generated were designed with the idea in mind that most documents, especially those considered here, are tree-structured, with text blocks and text clusters related to one another through parent-child and child-child relationships. Given the large number of features, it is unsurprising that Logistic Regression and SVM did not generalize well. Unlike in [1], we found that $L1$-regularization outperformed $L2$-regularization substantially. This provides evidence that the regularization technique that will perform best is dependent on the document being modeled.

Future work should involve conducting similar analyses on new data sets. Information extraction can be applied to a wide variety of documents, and as this study shows, different classifiers will perform differently on different types of documents. While tree-based classifiers worked well for the data analyzed in this paper, we hypothesize that the more structured a document is, the less complex the features need to be, and less complex classifiers will perform better.

References

[1] Wenshun Liu, Billy Wan, and Yaqi Zhang. Unstructured document recognition on business invoice. *Stanford CS 229 Machine Learning Final Projects, Autumn 2016*, 2016.

[2] Luiz Eduardo Guarino de Vasconcelos, André Yoshimi Kusomoto, and Nelson Paiva Oliveira Leite. Using image processing and pattern recognition in images from head-up display. In *International Telemetering Conference Proceedings*. International Foundation for Telemetering, 2013.

[3] Tomas Gogar, Ondrej Hubacek, and Jan Sedivy. Deep neural networks for web page information extraction. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 154–163. Springer, 2016.

[4] Ramesh Thakur, Suresh Jain, Narendra S Chaudhari, and Rahul Singhai. Information extraction from the un-structured document using grammatical inference and alignment similarity. *Procedia Technology*, 4:365–369, 2012.

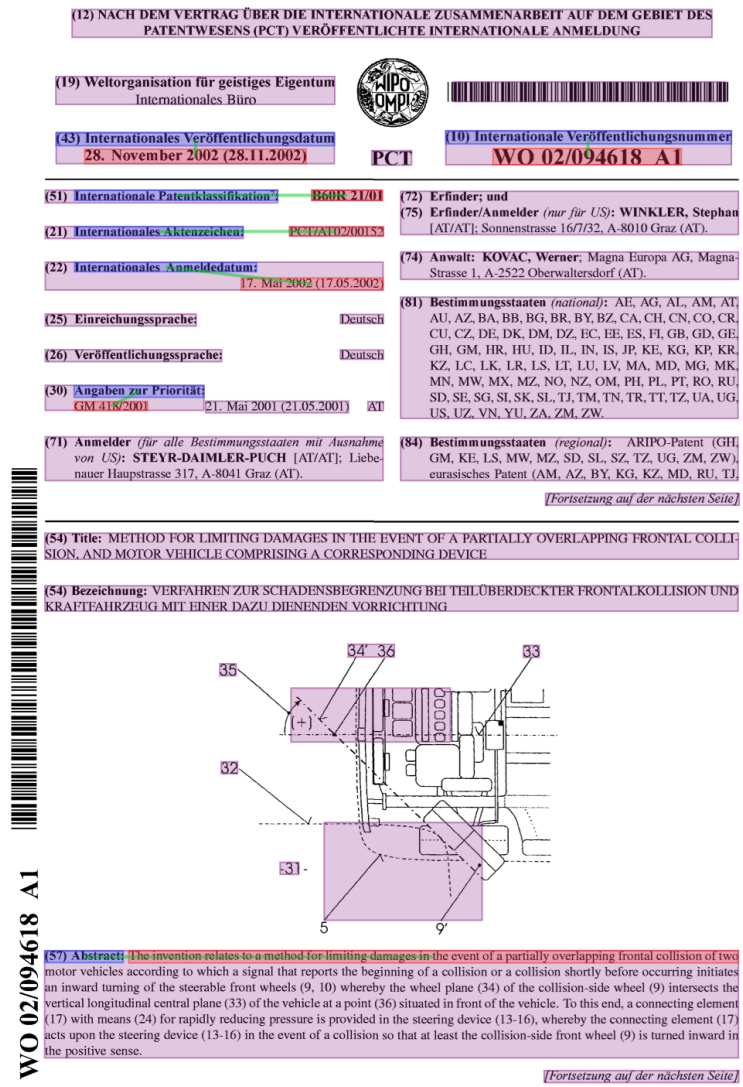
[5] Herve Dejean. Extracting structured data from unstructured document with incomplete resources. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 271–275. IEEE, 2015.

[6] Shantanu Kumar. A survey of deep learning methods for relation extraction. *arXiv preprint arXiv:1705.03645*, 2017.

[7] Thien Huu Nguyen. *Deep Learning for Information Extraction*. PhD thesis, New York University, May 2017.

8 Appendices

8.1 Appendix A: Ghega Patent Data Example



(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
28. November 2002 (28.11.2002)

PCT

(10) Internationale Veröffentlichungsnummer
WO 02/094618 A1

(51) Internationale Patentklassifikation?: B60R 21/01

(21) Internationales Aktenzeichen: PCT/AT02/00152

(22) Internationales Anmeldedatum:
17. Mai 2002 (17.05.2002)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
GM 418/2001 21. Mai 2001 (21.05.2001) AT

(71) Anmelder *(für alle Bestimmungsstaaten mit Ausnahme von US)*: STEYR-DAIMLER-PUCH [AT/AT]; Liebenauer Hauptstrasse 317, A-8041 Graz (AT).

(72) Erfinder; und
(75) Erfinder/Anmelder *(nur für US)*: WINKLER, Stephan [AT/AT]; Sonnenstrasse 16/7/32, A-8010 Graz (AT).

(74) Anwalt: KOVAC, Werner; Magna Europa AG, Magna-Strasse 1, A-2522 Oberwaltersdorf (AT).

(81) Bestimmungsstaaten *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Bestimmungsstaaten *(regional)*: ARIPO-Patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches Patent (AM, AZ, BY, KG, KZ, MD, RU, TJ,

[Fortsetzung auf der nächsten Seite]

(54) Title: METHOD FOR LIMITING DAMAGES IN THE EVENT OF A PARTIALLY OVERLAPPING FRONTAL COLLISION, AND MOTOR VEHICLE COMPRISING A CORRESPONDING DEVICE

(54) Bezeichnung: VERFAHREN ZUR SCHADENSBEGRENZUNG BEI TEILÜBERDECKTER FRONTALKOLLISION UND KRAFTFAHRZEUG MIT EINER DAZU DIENENDEN VORRICHTUNG

(57) Abstract: The invention relates to a method for limiting damages in the event of a partially overlapping frontal collision of two motor vehicles according to which a signal that reports the beginning of a collision or a collision shortly before occurring initiates an inward turning of the steerable front wheels (9, 10) whereby the wheel plane (34) of the collision-side wheel (9) intersects the vertical longitudinal central plane (33) of the vehicle at a point (36) situated in front of the vehicle. To this end, a connecting element (17) with means (24) for rapidly reducing pressure is provided in the steering device (13-16), whereby the connecting element (17) acts upon the steering device (13-16) in the event of a collision so that at least the collision-side front wheel (9) is turned inward in the positive sense.

[Fortsetzung auf der nächsten Seite]

WO 02/094618 A1

Key: LABEL — LINK — VALUE — TEXT CLUSTER

8.2 Appendix B: Ghega Data-Sheet Data Example

Preliminary

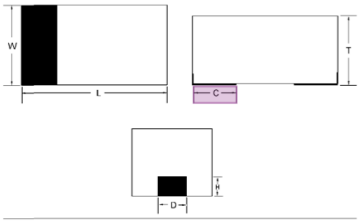
TAIWAN SEMICONDUCTOR

RoHS COMPLIANCE

TSZU52C2 – TSZU52C39

150mW SMD Zener Diode

0603



Features

- 150mW Power dissipation.
- High voltages from 2 ~ 39V
- Designed for mounting on small surface
- Extremely thin/leadless package
- Pb free product

Mechanical Data

- Cases: 0603 standard package, molded plastic
- Terminals: Gold plated, solderable per MIL-STD-750, method 2026.
- Polarity: Indicated by cathode band
- Weight: 0.003 gram (approximately)

Item	0603
L	0.071(1.80) 0.063(1.60)
W	0.039(1.00) 0.031(0.80)
T	0.033(0.85) 0.027(0.70)
C	0.014(0.35) Typical
D	0.018(0.45) Typical
H	0.012(0.30) Typ.

Dimensions in inches and (millimeters)

Maximum Ratings and Electrical Characteristics

Rating at 25 °C ambient temperature unless otherwise specified.
Single phase, half wave, 60 Hz, resistive or inductive load.
For capacitive load, derate current by 20%

Type Number	Symbol	Value	Units
Maximum Forward Voltage Drop at I_F=10mA	V_F	0.9	V
Maximum Power Dissipation	Pd	150	mW
Forward Current, Surge Peak 8.3ms Single half Sine-Wave superimposed on Rate Load (JEDEC method)	I_{FSM}	2.0	A
Operating Junction and Storage Temperature Range	T_{STG}, T_J	-65 to +125	°C

Version: A07

Key: LABEL — LINK — VALUE — TEXT CLUSTER