

Information Extraction From Document Forms Using Deep Learning



By

Khurram Shehzad

00000117369-Fall 2015-NUST-MS-CS-5

Supervisor

Dr. Muhammad Imran Malik

Department of Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree
of MScS

In

School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(May 2019)

Approval

It is certified that the contents and form of the thesis entitled “**Information Extraction From Document Forms Using Deep Learning**” submitted by **Khurram Shehzad** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Muhammad Imran Malik**

Signature: _____

Date: _____

Committee Member 1: **Dr. Faisal Shafait**

Signature: _____

Date: _____

Committee Member 2: **Dr. Hassan Aqeel**

Signature: _____

Date: _____

Committee Member 3: **Dr Adnan Ul-Hassan**

Signature: _____

Date: _____

Abstract

Information extraction from printed documents images remains an active research area. Several methods have been proposed in literature that extract information by utilizing various approaches, e.g., using document geometric or layout information along with various combination of textual attributes. We propose a learning based solution that does not use any layout information and solves this problem using only text blocks contained within the document. We transform the problem into entity relationship mapping and try to find out the probability of a relationship if it is true or not. The method can be used on new documents that are similar in content but can be different in size or layout.

Dedication

I dedicate this thesis to my daughter Hafsah.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Khurram Shehzad**

Signature: _____

Acknowledgment

I would like to thank my advisors, parents, family members and friends.

Table of Contents

1	Introduction and Motivation	1
1.1	Introduction	1
1.2	Challenges	2
1.3	Motivation	2
1.4	Goal	4
2	Literature Review	7
2.1	Overview	7
2.2	Related Work	7
2.3	Conclusion	10
3	Design and Methodology	11
3.1	Proposed Solution	11
3.2	Neural Tensor Network	12
3.3	Methodology	14
3.3.1	Dataset preparation	16
4	Dataset	18
4.1	Composition	18
4.1.1	Original Document Image	21
4.1.2	Pre-Processed Image	21
4.1.3	Text blocks file	21
4.1.4	Ground truth file	22
5	Results	24
5.1	Evaluation Measures	24
5.2	Hyper parameters tuning	25
5.3	Field wise results	33
5.4	Error Analysis	34
6	Conclusions and Future Work	35

List of Figures

1.1	A sample document image	3
1.2	Fields from a portion of a document	4
1.3	A sample data sheet document form	5
1.4	A sample data sheet document form	6
3.1	Proposed solution	11
3.2	Neural Tensor Network	13
3.3	Neural Tensor Network Layers	14
4.1	A sample patent document form	19
4.2	A sample data-sheet document form	20
5.1	Number of iterations vs accuracy	26
5.2	Batch size vs accuracy	27
5.3	Corrupt size vs accuracy	28
5.4	Regularization vs accuracy in data sheets	29
5.5	Regularization vs accuracy in patents	29
5.6	Activation function vs accuracy	30
5.7	Embedding source vs accuracy	31
5.8	Data sheets and patents cross validation.	32

List of Tables

3.1	Entities and words	17
5.1	Data sheets fields wise results	33
5.2	Patents fields wise results	33

Chapter 1

Introduction and Motivation

1.1 Introduction

Information extraction also sometimes termed as document understanding problem is defined as extracting useful and structured information from image of an unstructured or semi-structured document. This involves the task of extracting of information that is understandable by humans and transforming it into format for machine compatible way and then for further processing or storage purpose. The information in a document is present in many content forms e.g. text, images, logos, tables and input fields etc. Information in textual objects further comprises of dates, names, numbers, addresses and other text contents. A document in the form of paper needs to be transformed into electronic format which is then acceptable for the document understanding systems. A straight forward way of obtaining document images is by scanning original printed documents. Extracting information from images of printed documents play an important role in many application domains, e.g. office automation, knowledge management and document archival. The output of a information extraction system may be of different format e.g. extracted named value pairs, table contents, text content classification. The few common documents types include forms, invoices, medical reports, postal envelopes, technical specification sheets, patents, faxes, insurance documents, contracts, letters and scientific research papers. Processing these documents manually is expensive due to human labor costs incurred when processing a large number of documents. Adding a level of automation to the process of extracting information will have a profound impact in many applications e.g. office automation, knowledge management and document archival. Making a fully automated system for information extraction is a huge challenge due to large variety of document layouts and content types.

There are many commercial [14] [29] information extraction systems available which help in automation of various tasks.

A document contains several pieces of information. An example document is shown in figure 1.1 Fields found in documents is the particular area of focus for our work. A field has a label and an associated value as shown in figure 1.2. There are various challenges involve in field detection and identification. A field can be found anywhere in a document, the data type of each field's values could be different, e.g., numeric, text only, date, currency and so on. Variation in font sizes and colors also play a significant role in adding complexity.


Information extraction systems are widely classified into two types. First where document class to be processed is known and second where document class is unknown. Our approach falls into the first category where we know the class of a document being processed. In this type of systems the knowledge about document class should be known in advance.


1.2 Challenges

There are many challenges involved in information extraction from fields of a document. The position of a particular field is not fixed even in same type of document. The length and width of field label and its content is also varying. Fields compromise of variety of data types e.g text only, numeric, alphanumeric, currency, data and time. Difference in fonts also play an important role and adds more complexity during information extraction process. Some fields may be missing even if we consider only one type of document. As an example two portions of document forms from data sheets class of two different vendors are shown in figure 1.3 and 1.4. Only fields with label **Power Dissipation** and **Storage Temperature** are highlighted. Although these two forms are from same document class but they possess significant difference in terms of field position, size and content of their value data type.

1.3 Motivation

The focus of this work if to make up limitations for previous work. Create an information extraction system that is not dependent on layout of documents and can work on documents with varying layouts.

(19)  **Europäisches Patentamt**
European Patent Office
Office européen des brevets



(11) **EP 1 632 691 A2**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication: **08.03.2006** Bulletin 2006/10

(21) Application number: **05019107.1**

(22) Date of filing: **02.09.2005**

(51) Int Cl.: **F16D 55/00 (2006.01)**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IS IT LI LT LU LV MC NL PL PT RO SE SI SK TR

Designated Extension States:
AL BA HR MK YU

(30) Priority: **03.09.2004 JP 2004257201**

(71) Applicants:

- **Honda Motor Co., Ltd.**
Tokyo (JP)
- **NISSIN KOGYO CO., LTD.**
Ueda-shi,
Nagano (JP)

(72) Inventors:

- **Tomita, Hiroaki**
Honda R&D Co., Ltd.
Wako-shi
Saitama (JP)
- **Toda, Makoto**
Honda R&D Co., Ltd.
Wako-shi
Saitama (JP)
- **Takayanagi, Naoki**
Nissin Kogyo Co., Ltd.
Ueda-shi
Nagano (JP)

(74) Representative: **Herzog, Markus et al**
Weickmann & Weickmann
Patentanwältin
Postfach 86 08 20
81635 München (DE)

(54) **Brake caliper structure of a quad bike**

(57) A brake caliper structure of a straddle seat off-load vehicle includes a brake caliper (111) having a caliper bracket (131), a caliper assembly (134) connected to the caliper bracket by two connecting portions (132, 133) so that the caliper assembly can move relative to the caliper bracket to clamp a brake disc (137). One connecting portion (133) includes a slide pin (172) connected to the caliper assembly and slidably received in a guide hole (171a) in the caliper bracket for effectively guiding movement of the caliper assembly relative to the caliper bracket, and the other connecting portion (132) includes a connecting screw (166) as a fixed pin secured to the caliper bracket, and a rubber bushing (163) as an elastic member disposed between the caliper assembly and the fixed pin for taking up the tilting of the brake disc.

Printed by Jouve, 75001 PARIS (FR)

EP 1 632 691 A2

Figure 1.1: A sample document image

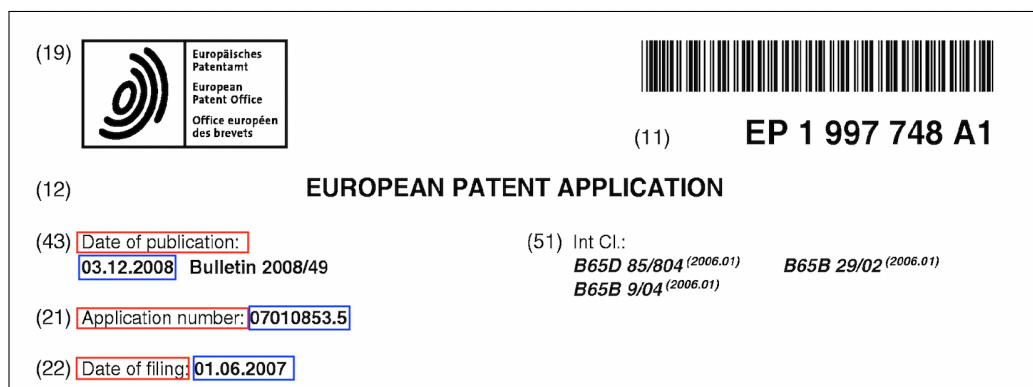


Figure 1.2: Fields from a portion of a document. There are three fields shown here with their labels in red boxes and values in blue boxes. The fields are `Date of publication:`, `Application Number:`, `Date of filing:` and their corresponding values are 03.12.2008, 07010853.5 and 01.06.2007 respectively.

1.4 Goal

The main goal of our work is to design and implement an information extraction system for diverse document forms with respect to layout and size. The information extraction system should not depend on any layout information and that can be easily generalized.

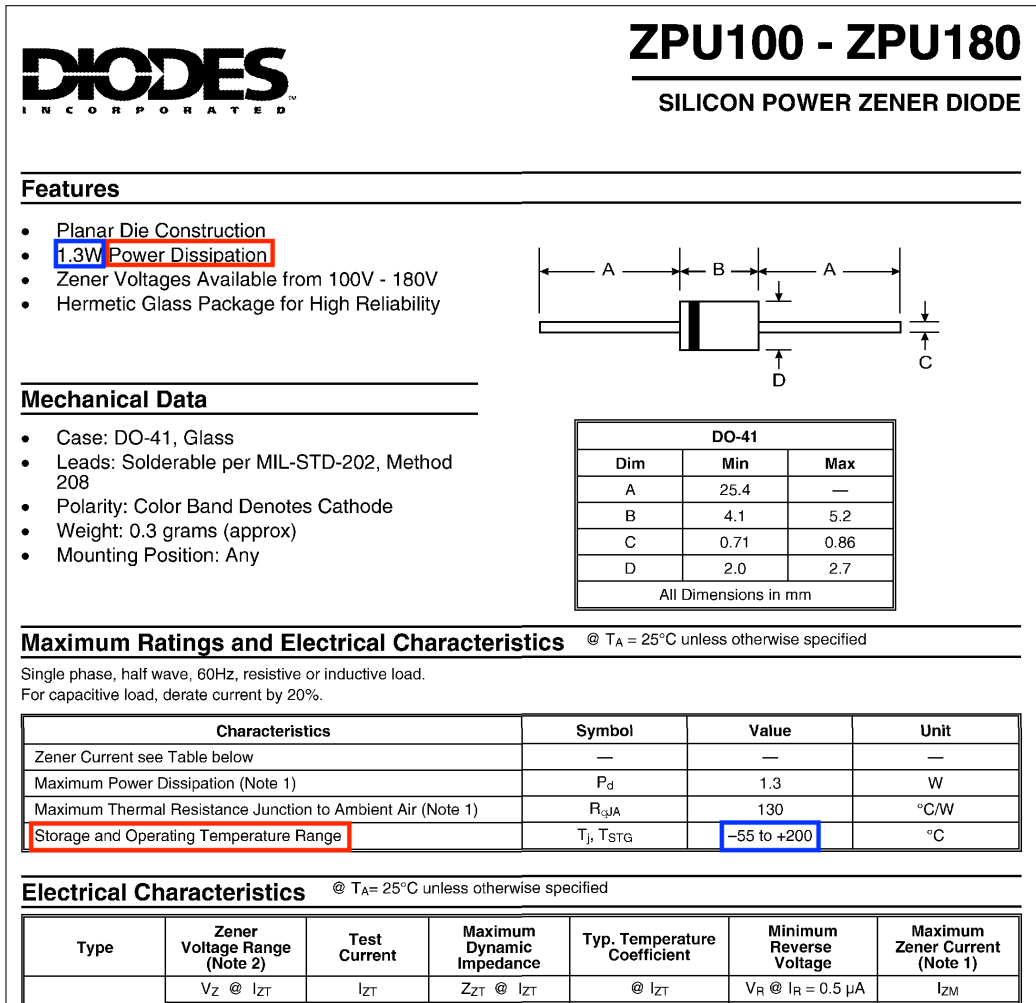


Figure 1.3: A sample data sheet document form. There are two fields shown here with their labels in red boxes and values in blue boxes. The fields are Power Dissipation and Storage temperature and their corresponding values are 1.3W and -55 to +200 respectively.

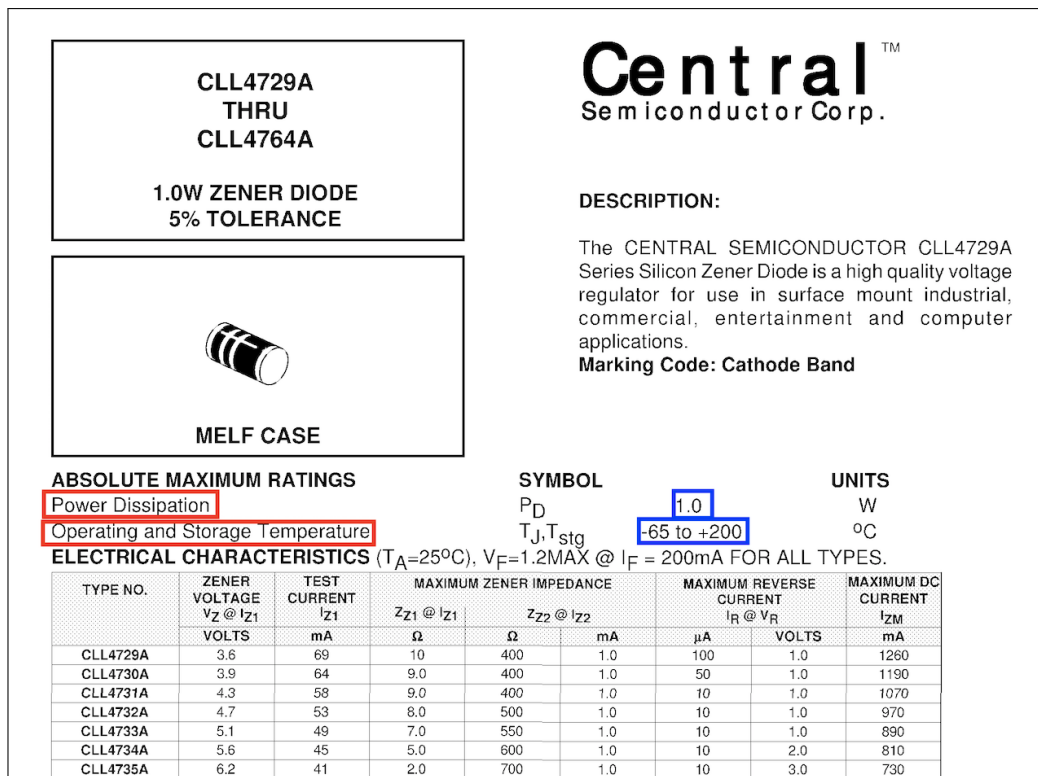


Figure 1.4: A sample data sheet document form. The fields are Power Dissipation and Storage temperature and their corresponding values are 1.0W and -65 to +200 respectively.

Chapter 2

Literature Review

This chapter contains various contributions, related work and challenges involved in the field of information extraction. We also discuss the advantages and short comings of these approaches.

2.1 Overview

Information extraction approaches are classified based on fact if class of document to be handled is known or unknown. If it is not necessary for a system, to have knowledge about the document class, a good quantity of knowledge regarding exact application domain i.e receipts, invoices etc is normally required and is part of system. As an example a list of "main primary tags" has been used in [6] to find labels of information. This work only deals with the invoices and is dependent on language. The systems which require document class to be known in advance are usually more efficient in detection and extraction of required information, but these have to struggle with two problems. First is the problem of association of each document to its corresponding class and second problem is to define document model for every document class. Solving these two problems require human involvement. Excellent information extraction systems are present that can process documents from same template beforehand e.g a keyword, rule or layout based systems. Some collection of systems have been proposed that depend on first classifying the template [10, 13, 16, 27, 33, 35].

2.2 Related Work

A lot of work on document understanding deals with invoices [6,10,11,18] and forms [4,34] because of economic value of these types of documents in terms

of volume and cost. There are numerous workflows involved in information extraction depending on particular document type and end user application. Document understanding systems can be categorized by the approach they use. There are various approaches found in literature ranging from a simple technique of using regular expressions [1], using pre-defined keywords [19] to the more complex scenarios of finding addresses [22] or contents of tables [6]. The complication involved in document understanding systems and various challenges involved are studied in [15] using black pixels spatial density and image edges. The work in [4] is related to automatic processing of printed forms. The authors have proposed a method document structure grammar which is represented by Table Form Markup Language TFML. This is a semi-automatic method which analyzes a blank form layout and characterizes its structure in TFML. This work make use of printed rules which may not exist in some other type of documents. Although our work does not directly deal with document classification, the task of matching document with classes is discussed in detail in [5] and [37].

The approach discussed in [11] expresses the document in the form of attributed relational graphs. A good performance in identifying the searched information is stated but this has a limitation of experimentation on only two document classes. Another similar approach is presented in [3]. In order to increase the limitations of coverage of graph based work, they have used decision trees along with use of bi-grams and tri-grams which are then applied on to the textual content blocks. The objective is to identify the document structure; that is recognizing general information (for example captions, body, title). The underlying model is composed of collection of logical and geometrical structures and is based on statistical methods. This works efficiency is tested on dataset consisting of 800 documents of single page, where the complexity is dependent on three levels basing on the count of objects, and these objects belongs to text regions with in the document. The work in [23] identifies the document structure. The authors have made fuzzy logical rules for the classification of textual blocks, which involve both textual and layout features.

In the work by Daniel and Michael [35], whole document (after performing optical character recognition)is represented in XML hierarchical structure with top level beginning at page, paragraphs, lines words and characters. Each document is associated with a template. First it will try to find the document template and then use the template along with the results of OCR in XML are are passed to another module called indexer. Indexer performs the actual work of information extraction. Three types of indexers are used. First is fix field indexer for each field which is fix value across template. Second is a position-based indexer with fix position and variant value. Third

is context-based indexer with variant position and variant value.

The work presented by Bela d [6] is based on morphological tagging for invoice analysis. The reason for using this approach is to tag columns and fields of tables. The drawback is that tables to be processed are already extracted before tagging. The approach discussed in [18] have used some words are flagged as keywords for the purpose of information extraction.

Marcal et al. [33] develop a structural model that encodes pairwise relationship between a field to be extracted and all other words that appear in the document. This structural model is represented by a star graph. Where each node of graph holds the word transcription and each edge represents the spatial relationship between word and field in polar coordinates. Daniel et al. [13] presented a positional based approach. It works by learning generic position of each field from a document template. Each included word in a document is concatenated with the position of its occurrence. Matsumoto et al. [26] have considered layout based properties e.g italic or bold characters for generating rules.

Cesarini et al [10] have worked on invoice documents. First a layout structure of invoice is extracted. This layout is extracted using a bottom up approach and attempts to cluster pixels into physical objects(e.g., lines, words and logos). Next the document understanding module uses a combination of position and value based methods on word physical objects to obtain final results.

Dengel et al [16] use different set of rules to extract information. The rules are governed by field data type, its value and position with respect to document page. In the work by Medvet et al [27] the document is represented by a set of blocks. Each block constitutes its position, size and content. Further a document also has also an associated schema and a model. Schema dictates what information needs to be extracted from the given document and model defines how to discover that information. A model has set of rules where each rule corresponds to a single field from schema. A rule is a triplet of cardinality, matching probability and an extraction function to determine the best match with respect to the fields. Pandey et al. [31] have used similar probabilistic based model to identify index fields and text content from tables. A drawback of these trainable approaches is that it require a large number of sample documents and require annotations manually.

Rasmus et al [30] have presented an approach for invoice analysis that require no configuration at start. The work does not depend on layout of invoice but uses a global model for invoice and which can be generalized to new and unseen invoices with different layout. This single model is trained from the data which is automatically extracted from the feedback provided by end user. The advantage is that it eliminates the dependency on user

to provide precisely annotated data. Although authors have claimed their work to be able to process invoices with unseen and varying layout but it still require a global invoice model. Another issue is the results are reported on a private data set of invoices.

Another problem which is not directly linked information extraction from printed documents but is dependent on document analysis; is the recognition of order of reading. The work given in [2] and [40] find many features (foreground color, background color, font size, font type, coordinates etc) from colorized English language scanned documents and then using spatial inference and natural language processing

2.3 Conclusion

Information extraction and document understanding is one of the focus areas for many automation systems. Many information extraction systems have been described that are represented by a specific layout. Many organizations have to process a very large quantity of documents that are described by different layout and which correspond to different classes. In majority of the cases the system require a document template and/or a document model in order to perform information extraction process. The major disadvantage is that as system depend on having seen the template before, these are not able to handle documents with unseen templates with enough accuracy.

Chapter 3

Design and Methodology

We have transformed the information extraction problem into entity relationship mapping also sometimes called common sense reasoning [39] and used Neural Tensor Network [36] for training and evaluation purpose.

3.1 Proposed Solution

We have proposed solution as a series of steps. These steps are listed below and shown in figure 3.1

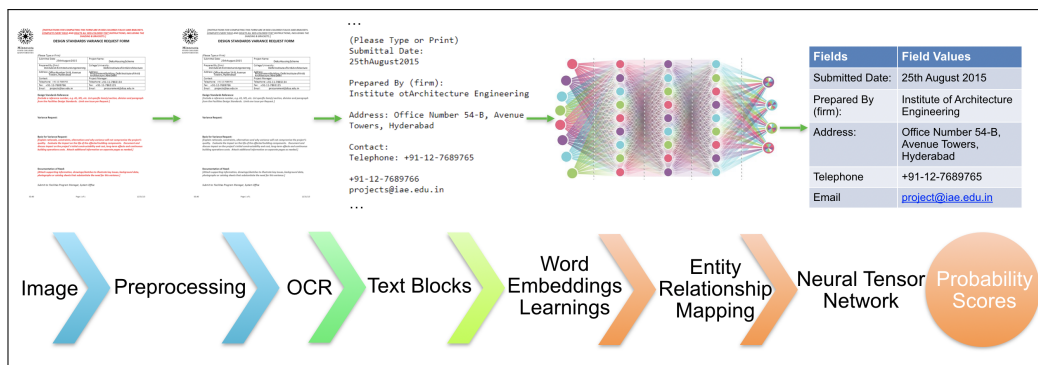


Figure 3.1: Steps in our proposed solution

1. Pre-processing
2. OCR
3. Text Blocks

4. Word Embeddings Learning
5. Entity Relationship Mapping
6. Neural Tensor Network

The **Pre-processing**, **OCR** and **Text block** are part of data set and are discussed in chapter 4 and in sections 4.1.1, 4.1.2 and 4.1.3. The word embeddings learning and entity relationship mapping is described in 3.3.1 and 3.3 respectively. We have explained Neural Tensor Network below in section 3.2.

3.2 Neural Tensor Network

The goal of network is to learn models for common sense reasoning, the capability to comprehend that certain facts hold entirely because of other previous relations. The goal can be described in other words as to link prediction in an existing network of relationship between entities. A neural tensor network is good for reasoning over relationship between two entities. In this paradigm there are two entities and a relationship between them. The entities and relationship are expressed in the form of triplet (e_1, R, e_2) where e_1 and e_2 are entities and R is relationship between them. The objective of this approach is argument such that two entities (e_1, e_2) are in relationship R . Input to the network is entities along with relationship and network outputs the probability value of relationship R holding true between entities e_1 and e_2 as shown in figure 3.2. The network gives a high probability value if relationship is true and low otherwise. An example of entity relationship triplet is **russell_bufalino gender male**. Here **russell_bufalino** and **male** are entities and the relationship between them is **gender**.

The objective of network is to be able to predict if a relationship R holds true for two entities (e_1, e_2) . For example a triplet $(e_1, R, e_2) = (\text{russell_bufalino gender male})$ is true and with that probability. The Neural Tensor Network (NTN) consists of a bilinear tensor layer that directly relates to two entity vectors across multiple dimensions. The network calculates the probability of two entities in a relationship by below NTN function

$$g(e_1, R, e_2) = u_R^T f(e_1^T W_R^{[1:k]} e_2 + V_R \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_R)$$

Here $f = \tanh$ is nonlinearity that is applied element-wise. $W_R^{[1:k]} \in \mathbb{R}^{d \times d \times k}$ represents a single tensor and vector $h \in \mathbb{R}^k$ is the result of bilinear product

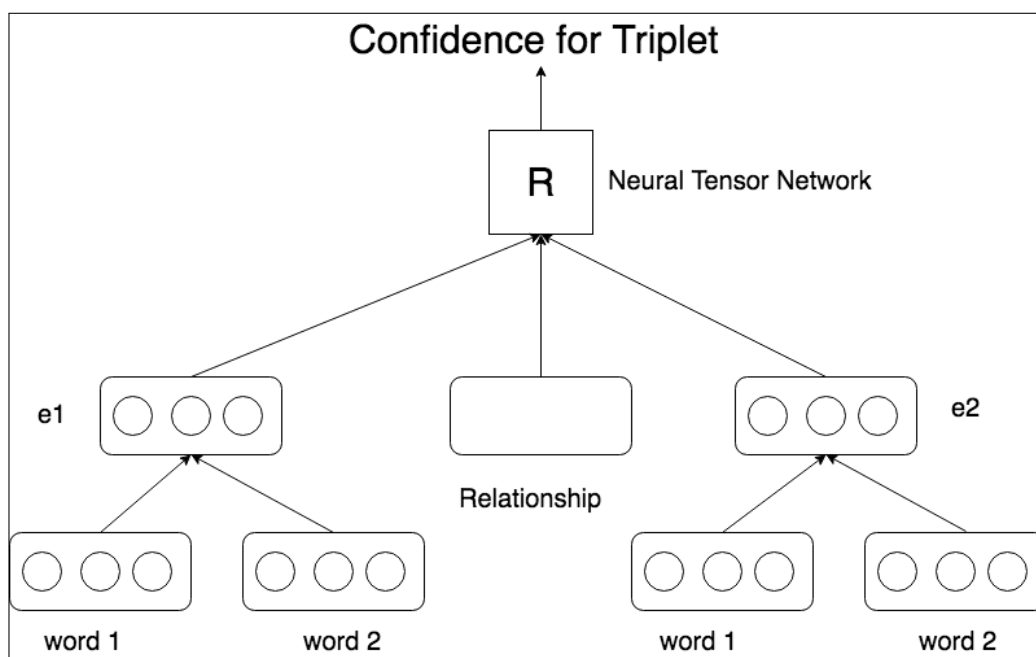


Figure 3.2: Entities are fed into the network in the form of corresponding word embeddings along with the relationship. The network outputs probability of relationship being true.

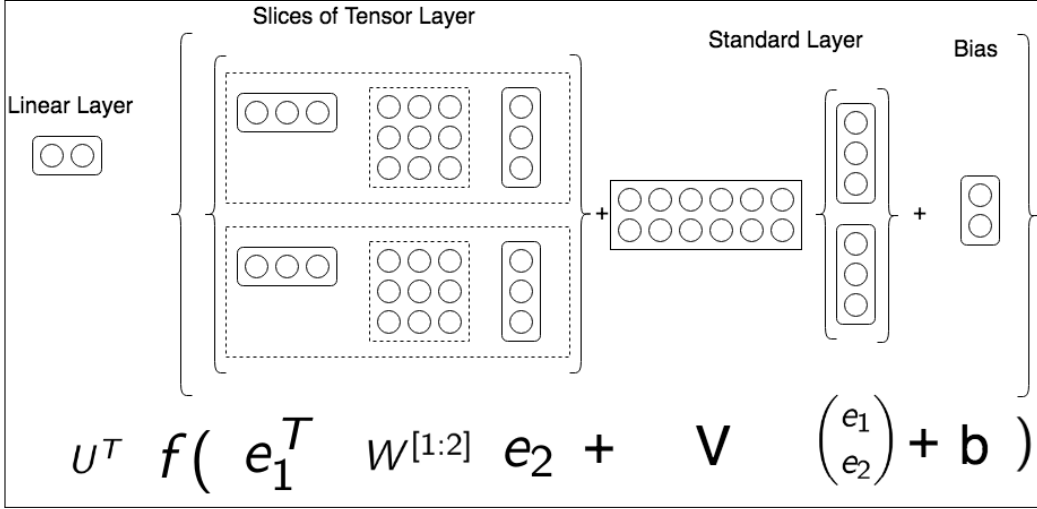


Figure 3.3: A neural tensor network suitable for reasoning over relationship between two entities.

of $e_1^T W_R^{[1:k]} e_2$. The remaining parameters for relationship R are in the from the standard neural network i.e $V_R \in \mathbb{R}^{k \times 2d}$, $U \in \mathbb{R}^k$ and $b_R \in \mathbb{R}^k$.

A detail description of network is shown in figure 3.3. There are few hyper parameters in network and these are

- Batch Size
- Corrupt Size
- Regularization Parameter
- Number of Iterations
- Activation Function

3.3 Methodology

In order to use the Neural Tensor Network we have to convert all of the entities and relationships into embeddings. Vectorized representation of text are known as embeddings and serve as a fundamental component in many Natural Language Processing(NLP) systems. Embeddings can be generated for words [28], sentences [21], paragraphs [12] and even for whole documents [24]. Embeddings play a vital role in our approach as we are not using any layout information from documents and our method only depends upon text blocks. There are two choices available with respect to the use of embeddings. First

choice is to use randomly initialized embeddings and the other is to learn embeddings using an algorithm. We have used **Continuous Bag Of Words** (CBOW) algorithm by Tomas et al [28] to learn embeddings for entities and relationships. Embeddings help in establishing similarity between two entities. In our case an entity may contains more then one word and thus forming a multi word entity e.g **Storage Temperature**. If we assign a single vector to each entity as in [8] [20] [7] then it does not permit sharing of statistical power between words making up entity. For this reason we have modeled every word as d -dimensional vector $\in \mathfrak{R}^d$ and calculated entity vector as the composition of its word vectors. Therefore for a total of N_W unique words making N_E entities, if training on word levels (during training the word vectors also receive error derivatives through back propagation) and represent entities using word vectors, then the complete embedding has the dimensionality of $E \in \mathfrak{R}^{d \times N_W}$ or else if we represent every entity as a single vector so its dimensionality will be $E \in \mathfrak{R}^{d \times N_E}$. We have used entity vector by averaging its words vectors. As an example $V_{StorageTemperature} = \frac{V_{Storage} + V_{Temperature}}{2}$. This can be generalized as

$$V_{entity} = \frac{V_{w_1} + V_{w_2} + V_{w_3} + \dots + V_{w_n}}{n}$$

There is an additional advantage of training word vectors that we can benefit from already trained unsupervised word vectors, which generally adds some syntactic and semantic information. All embeddings are of $d = 100$ -dimensional vectors.

All the model are trained with objective function of max-margin. The central thought is that every triplet from training set $T^{(i)} = (e_1^{(i)}, R^{(i)}, e_2^{(i)})$ will get a much higher score then a triplet in where one of the entities is being replaced with a random entity. We mention the triplet with a random entity corrupted and express the corrupted triplet as $T_c^{(i)} = (e_1^{(i)}, R^{(i)}, e_c^{(i)})$. Here we have taken entity e_c randomly which can come at that position. Since we have only one relation so for the above both original and corrupted triplets $i = 1$ for $R^{(i)}$. Let the NTN parameters be $\Omega = u, W, V, b, E$. We try to minimize the below function

$$J(\Omega) = \sum_{i=1}^N \sum_{c=1}^C \max(0, 1 - g(T^{(i)}) + g(T_c^{(i)})) + \lambda \|\Omega\|_2^2$$

where N is the total number of training triplets and the correct triplet is scored higher than corresponding corrupted one with up to margin of 1. For every triplet we generate C number of random corrupted triplets. Standard

L_2 regularization parameter is used for all parameters and weighted by the hyperparameter λ .

The training of model is done by taking the derivatives with respect to the five groups of parameters. Similar to as in general backpropagation the derivatives of the standard neural network weight V are the same. Excluding the relation specific index R , below is the derivative for the j 'th slice of full tensor

$$\frac{\partial g(e_1, R, e_2)}{\partial W^{[j]}} = u_j f'(z_j) e_1 e_2^T$$

where

$$z_j = e_1^T W^{[j]} e_2 + V_j \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_j$$

where

V_j is the j 'th row of V matrix and

z_j is the j 'th element of the k -dimensional hidden tensor layer

We have used minibatch L-BFGS [25] for optimization which has property to converge to local optimum for the non-convex objective function. We have also tried AdaGrad [17] but its performance is not better then L-BFGS.

Our objective is to make prediction of correct facts in the form of the relations (e_1, R, e_2) in the testing data. We have find a threshold T_R in the development set such that $g(e_1, R, e_2) \geq T_R$ then (e_1, R, e_2) holds else it does not holds. In the process of making testing set for classification, we had randomly changed entities from correct testing triplets and that has resulted in a total of `2×Testing triplets` where number of positive and negative samples are equal. For example a correct triplet is `-55C to +150C, is, StorageTemperature` and its corresponding negative example might be `-55C to +150C, is, Model`. The triplets which are classified correctly made up the final accuracy value.

3.3.1 Dataset preparation

The Ghega dataset is in the form of a set of text blocks for each document image along with ground truth. Before using neural tensor network we have to convert this data set in to the format of entities and relationships. We read both CSV files and generate corresponding entity relationship entries triplets. An example of such a triplet is

`7. November 2002 (07.11.2002)#is#FilingDate`

where `7. November 2002 (07.11.2002)` is first entity and it is the text block read from CSV file, `FilingDate` is second entity and is also called field in our case, `is` is the relationship between these two entities and `#` is special symbol(separator) used to separate entities and relationship in a single line.

	Data Sheets	Patents
Entities	7847	6218
Words	8141	8952

Table 3.1: total number of entities and words in **Data sheets** and **Patents**

An entity may consist of one or more words. The total number of entities and words for both **Patents** and **Data-sheets** are shown in table 3.1. We have only one relationship for both **Date sheets** and **Patents**.

Chapter 4

Dataset

We have used ghega dataset [27] for training and testing of our approach. This dataset is super set of a public dataset and is used by [10].

4.1 Composition

The documents in data set are divided into two groups. This partition in data set is based on different domains of documents. First document group is *Patents* and second is *Data-sheets*. The text in data set consists of printed English language characters.

Patents consists of 136 patent document images obtained from 10 different patent sources where every source of patent is related to a different class. The class which is largest of all classes consists of 22 patents and 7 classes consists of 10 or more patents. There are total eleven important fields in patents listed below

- Title
- Applicant
- Inventor
- Representative
- Filing Date
- Publication Date
- Application Number
- Publication Number

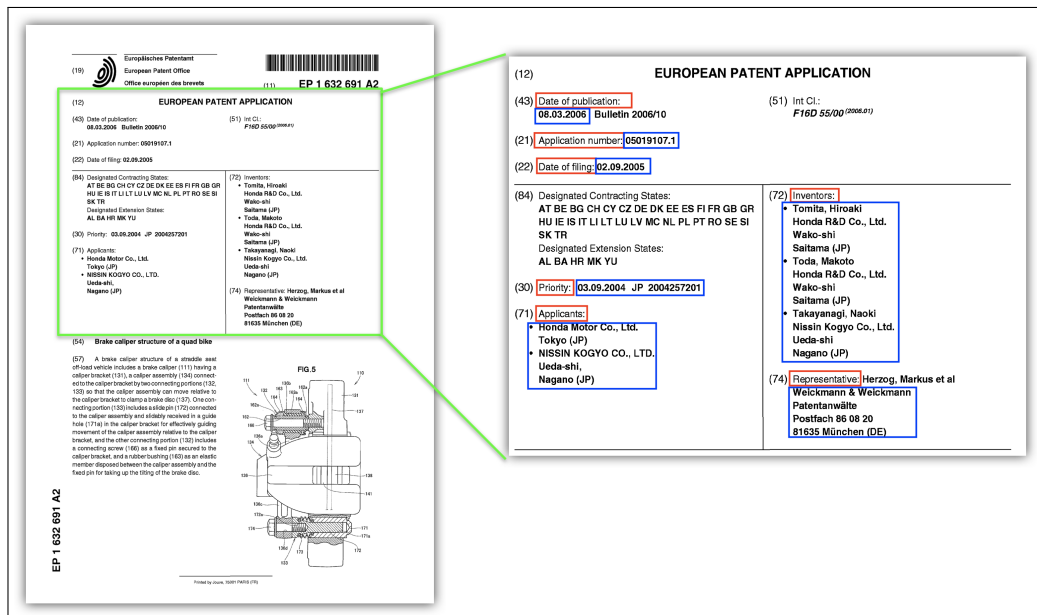


Figure 4.1: A sample patent document form is shown on left side and a zoomed in snapshot of same document is shown on right side. There are seven fields shown here with their labels in red boxes and values in blue boxes. Note that some fields have their values on right side and some have on their bottom.

- Priority
- Classification
- Abstract 1st line

A sample patent document form is shown in figure 4.1

Data-sheets contains 110 data sheets of different electronic components (e.g Zener diodes) from different vendors and divided in 10 classes. There are total eight important fields in data-sheets listed below

- Model
- Type
- Case
- Power Dissipation
- Storage Temperature

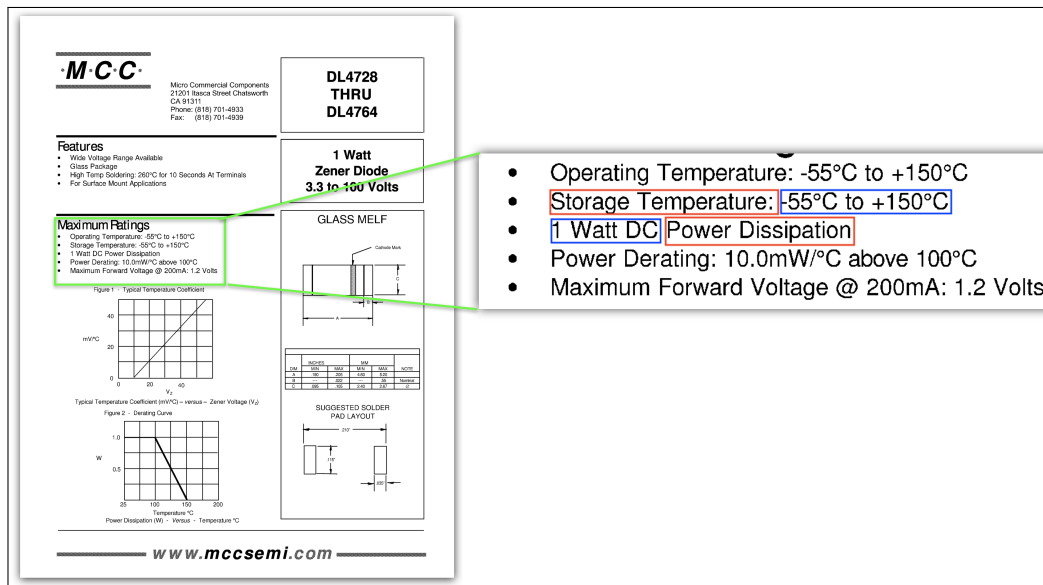


Figure 4.2: A sample data-sheet document form is shown on left side and a zoomed in snapshot of same document is shown on right side. There are two fields shown here with their labels in red boxes and values in blue boxes. The fields are **Storage temperature** and **Power Dissipation** and their corresponding values are **-55 to +200** and **1.3Watt DC** respectively.

- Voltage
- Weight
- Thermal Resistance

A sample data-sheet document form is shown in figure 4.2. Data-sheets of same class have shared producer type and consumer type. The argest 5 classes contain approximately 85% of these documents.

In the document data set the following three situations may occur.

1. There are certain classes whose corresponding documents do not contain a given field at all e.g in a certain producer of a certain electronic type component may not present **Storage Temperature** field.
2. There are certain classes for which only few documents contain a certain field e.g a certain patent source has **Representative** field while some other many not contain this field.
3. There are certain classes whose document contain multiple occurrence of certain fields e.g there are certain documents that may contain a

field of `Total Amount` in more than one page or even more than once on same page.

Each document sample in data set compromises of four different files. Each document is obtained from their corresponding PDF sources and converted into binary images at 300 dpi. Approximately 50% of those PDFs were gathered by scanning the corresponding paper document. Each document image is accompanied by three more files, so in total there are four files for each sample document.

4.1.1 Original Document Image

First file is an original document image in the form of png image at 300dpi .

4.1.2 Pre-Processed Image

Second file is processed image of original image. This processing includes de-skewing and binarization of original image. Every document image is transformed into collection of text blocks by utilizing an OCR engine [2] [9].

4.1.3 Text blocks file

The OCR used was configured in best possible way and it may deskew image if required. The OCR system may add some errors and in most of the scenarios these errors were result of different scanning artifacts. These errors from OCR can be categorized in two categories, first is segmentation errors and second is text-recognition errors. First type of error results in text blocks having different text components amid different documents from the same class. Second type of error resulted in textual content values which are different from what is really printed on the paper document. It is generally seen that segmentation errors normally imply text recognition errors. In addition low printing quality e.g documents which were printed from dot matrix printer mostly generate text recognition errors. The data set also includes the documents where OCR engine has produced errors, this has affected the text blocks that have the values being searched. These has made us to access our method capability in a very practical environment with respect to these OCR errors. Next the third file is a CSV file which contains all the text blocks which were found by applying this OCR to the processed document image. This blocks file contains multiple blocks where each block is rectangular piece from processed image file where OCR engine has found a single-line of text. OCR is used with default configuration with respect to line segmentation. A single line contains

- block type (just one fixed value)
- page (Starting from 0)
- x position from upper-left corner of the page, in inches
- y position from upper-left corner of the page, in inches
- width in inches
- height in inches
- found text
- Serialized data (not used)

An example of text line from this CSV file is

TextLineBlockCommon, 0, 0.4833329916000366, 3.809999942779541, 2.1266698837280273, 0.11333300173282623, ELECTRICAL CHARACTERISTICS, [B@2dee1281

4.1.4 Ground truth file

The fourth file called ground truth file also in the form of a CSV file. A single line of file contains

- Element type
- Page of the label block(-1 if not present)
- x location of the label block
- y location of the label block
- w(width) of the label block
- h(height) of the label block
- text of the label block
- page of the value block
- x location of the value block
- y location of the value block
- w(width) of the value block

- h(height) of the value block
- text of the value block

An example of text line from ground truth CSV file is

PowerDissipation, 0, 0.5066670179367065, 3.549999952316284, 1.926669955253601, 0.14333300292491913, Power Dissipation

Chapter 5

Results

5.1 Evaluation Measures

There are many evaluation criteria available e.g. accuracy, precision, recall F1 score. The results in our work have been analyzed using all the measuring scales and final comparison has been made on the basis of F1 score because it outperforms the other scales.

Accuracy

Accuracy is defined as the number of entities relation triplets identified as correct for both positive and negative pairs divided by the total number of entities relation triplet in test data.

Precision

Precision describes the ratio of correctly identified entities relation triplets among the truly identified entities relation triplets. It is also sometimes referred to as exactness.

$$Precision = \frac{T_p}{T_p + F_p}$$

Recall

Recall describes the number of positive identifications divided by the number of positive values in test dataset.

$$Recall = \frac{T_p}{T_p + F_n}$$

F1 Score

F1 score is the mean of precision and recall. This is the most suitable measure scale as it deals with the non-uniform data distribution of data

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

We have tried different values of hyper parameters described earlier in section 3.3.1 and come with some optimum values of parameters which provide best results. The hyper parameters are tuned for giving maximum accuracy value.

5.2 Hyper parameters tuning

Below are the effects of using various values of hyper parameters on accuracy value. Each hyper parameter is tuned for both types of documents and their values are shown along side each other.

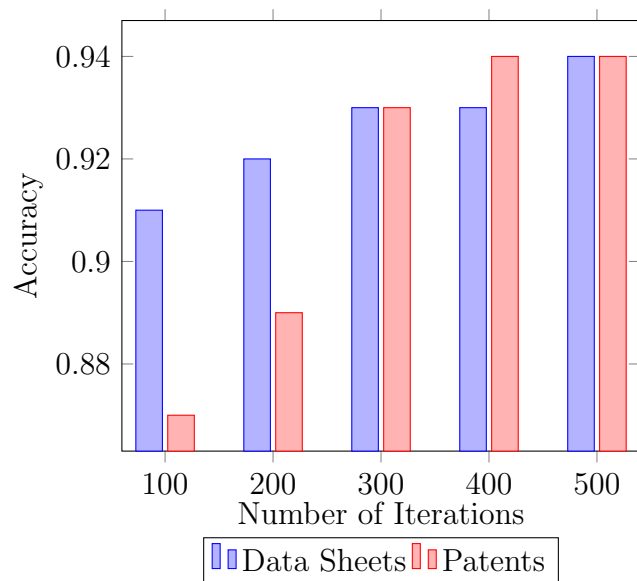


Figure 5.1: Variation in number of training iterations and its effects on accuracy. Data sheets and patents have best performance for number of training iterations at 500 and 400 respectively

We have performed the training on data set using various values of number of iterations while keeping all other hyper parameters constant. The Number of iterations for data sheets and patents provide best results for the values of 500 and 400 respectively see figure 5.1.

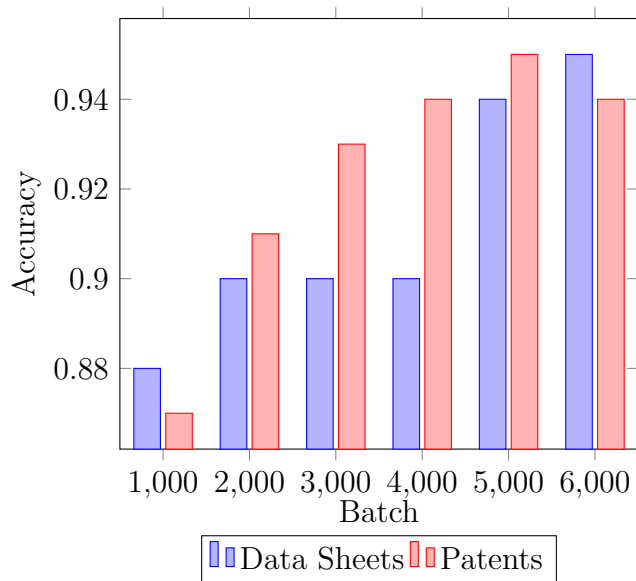


Figure 5.2: Variation in batch size and its effects on accuracy. Data sheets and patents have best performance for batch size of 6000 and 5000 respectively.

The batch size is the number of training samples used in one single iteration. There are three options available for batch size. 1. Batch Mode: The batch size value is equal to the total number of samples in data set and thus make iteration and epoch values same. 2. Mini-batch mode: The batch size value is less than the total number of samples in data set but is greater than one. 3. Stochastic Mode: The batch size is equal to the value of one. In this case the gradient and parameters are updated after every sample. For batch size the values of 6000 and 5000 work best for data sheets and patents respectively see figure 5.2.

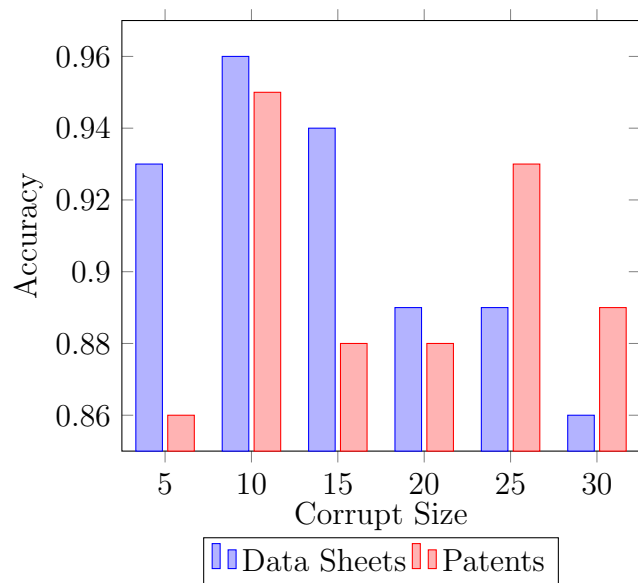


Figure 5.3: Variation in corrupt size and its effect on accuracy. Data sheets and patents have best performance for corrupt size of 10.

We pick a triplet T and replace its second entity i.e e_2 with another randomly picked entity. The score of objective function should be higher then corrupted with a margin of one. This phenomena is explained in 3.3. The corrupt size of 10 work best for data sheets and patents see figure 5.3.

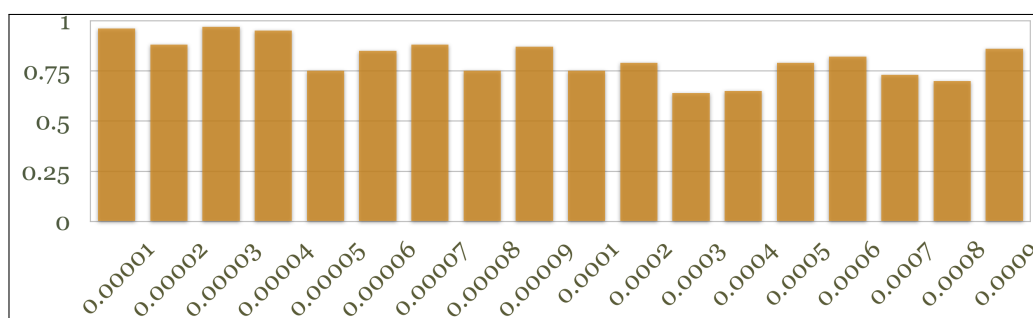


Figure 5.4: Variation in regularization and its effects on accuracy of Data sheets. It has best performance with value of 0.00001

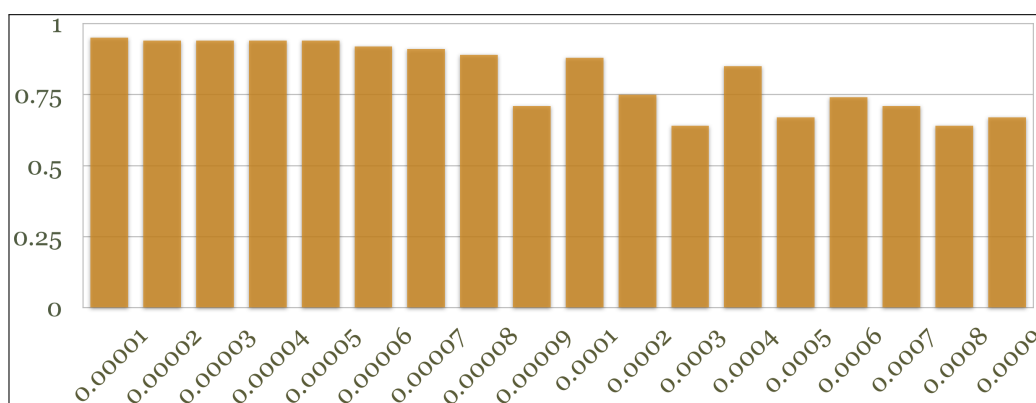


Figure 5.5: Variation in regularization and its effects on accuracy of Patents. It has best performance with value of 0.00001

The regularization helps in avoiding the model to overfitting to the training data. We have tried a range of values for regularization parameter and found that the value of 0.00001 is best for both data sheets and patents 5.4 and 5.5.

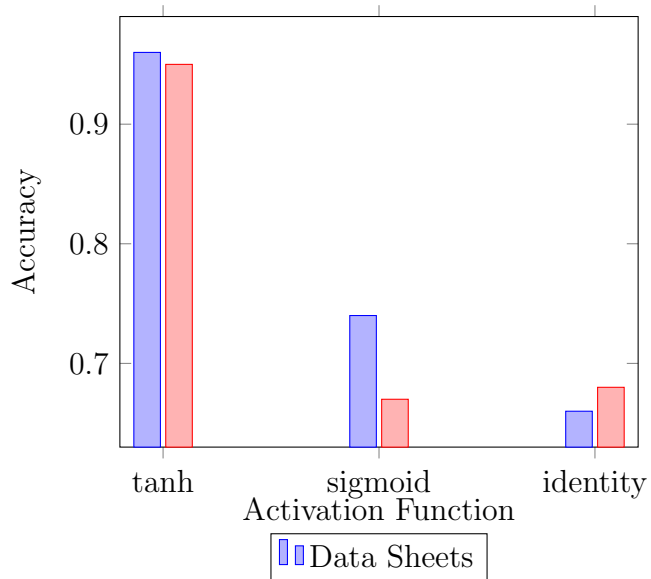


Figure 5.6: Effect of using different activation functions on accuracy. Data sheets and patents have best performance for tanh.

We had experimented with three activation functions namely `tanh`, `sigmoid` and `identity` and found that `tanh` works best for both data sheets and patents see figure 5.6.

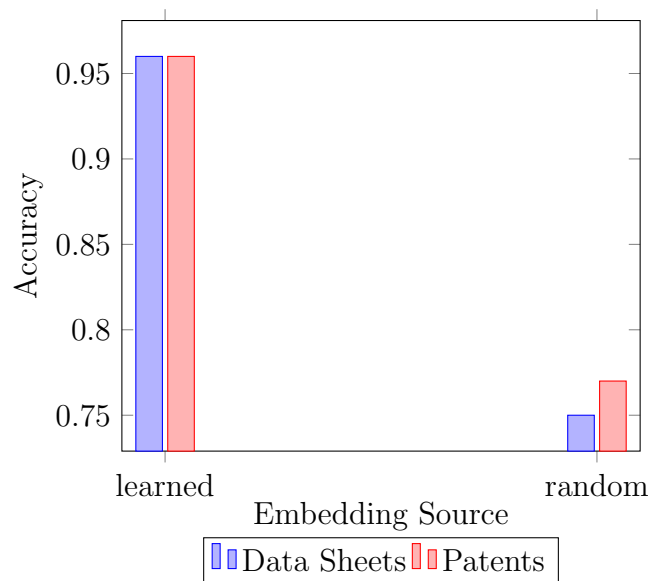


Figure 5.7: Effect of using different embeddings source on accuracy. Data sheets and patents have best performance when embeddings are learned.

As stated earlier we have converted all the entities into embeddings before using NTN, there are two methods that we used for embeddings generation one is to initialize them randomly and other is to learn embeddings. We have found that learning embeddings provide best results for both data sheets and patents see figure 5.7.

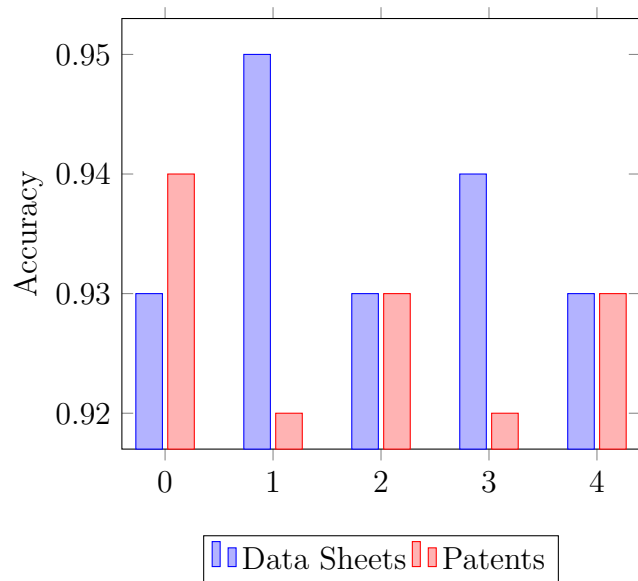


Figure 5.8: Data sheets and patents cross validation.

5.3 Field wise results

We have shown evaluation measurements discussed in 5.1 for each of the fields for both document types. The field wise results for **Date sheets** and **Patents** are shown in table 5.1 and 5.2 respectively. Their error analysis is explained in section 5.4.

	Accuracy	Error	Precision	Recall	F1
Case	0.98	0.02	0.92	0.98	0.95
Model	0.98	0.02	0.96	0.98	0.97
Power Dissipation	0.89	0.11	0.94	0.89	0.92
Storage Temperature	0.93	0.07	0.90	0.93	0.91
Thermal Resistance	0.96	0.04	0.99	0.96	0.98
Type	0.97	0.03	0.98	0.97	0.97
Voltage	0.94	0.05	0.91	0.99	0.95
Weight	0.99	0.01	0.91	0.99	0.95

Table 5.1: Data sheets fields wise results

	Accuracy	Error	Precision	Recall	F1
Abstract 1st Line	0.86	0.14	0.92	0.86	0.89
Applicant	0.95	0.05	0.90	0.95	0.92
Application Number	0.80	0.20	0.93	0.80	0.86
Classification	0.91	0.09	0.97	0.91	0.94
Filing Date	0.95	0.05	0.91	0.95	0.93
Inventor	0.92	0.08	0.97	0.92	0.94
Priority	0.89	0.11	0.96	0.89	0.92
Publication Date	0.96	0.04	0.89	0.96	0.92
Publication Number	0.99	0.01	0.89	0.99	0.94
Representative	0.99	0.01	0.92	0.99	0.95
Title	0.91	0.09	0.87	0.91	0.89

Table 5.2: Patents fields wise results

5.4 Error Analysis

The accuracy is worst for **Power Dissipation** and **Application Number** for **Data sheets** and **Patents** respectively. There are two reason for this. First is due to a large number of OCR errors. Second is because of difference in content value type (numeric vs alphanumeric) which arises due to different sources of these documents. A few examples of **Power Dissipation** are 250, 1 90 and I 75 and examples of **Application Number** are 05019107.1 and PCT/HU2004/000120. We have got best accuracy for fields whose value type have uniform appearance e.g **Weight** and **Publication Number** and have less number of OCR errors.

Chapter 6

Conclusions and Future Work

Information extraction from printed documents has been an active area of research for past many years. It has got many challenges despite being a lot of work is done in this domain. Most of the techniques require a document model, a document schema and combination of these in some ways along with layout knowledge for information extraction. We have proposed an approach to extract information without using any layout or positioning attributes. We had demonstrated this by transforming information extraction problem into common sense reasoning domain. We have used Neural Tensor Network for information extraction task and have achieved a reasonable level of success. This method can work on new documents that are similar in content but they can have a very different layout or size. Our approach is a generalized method and is not bound to any specific document type and can easily be extended to other type of documents.

There are some opportunity places in our work where there is a great chance of improvements. The first issue is minimizing the OCR errors. Although our work is not related to improving OCR results but OCR is an important step in increasing performance of any information extraction system. In this regard [38] have proposed to minimize OCR related errors. The second place for improvement is to combine learning of word and entity embeddings from other sources such as [32] which make use of unsupervised learning. Third improvement factor is to combine Neural Tensor Network and embedding learning network and make it an end to end learning system.

Bibliography

- [1] Serif Adali, A Coskun Sonmez, and Mehmet Gokturk. An integrated architecture for processing business documents in turkish. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 394–405. Springer, 2009.
- [2] Marco Aiello, Christof Monz, and Leon Todoran. Combining linguistic and spatial information for document analysis. In *Content-Based Multimedia Information Access-Volume 1*, pages 266–275. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D’INFORMATIQUE DOCUMENTAIRE, 2000.
- [3] Marco Aiello, Christof Monz, Leon Todoran, and Marcel Worring. Document understanding for a broad class of documents. *International Journal on Document Analysis and Recognition*, 5(1):1–16, 2002.
- [4] Akira Amano, Naoki Asada, Masayuki Mukunoki, and Masahito Aoyama. Table form document analysis based on the document structure grammar. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(2-3):201–213, 2006.
- [5] Alberto Bartoli, Giorgio Davanzo, Eric Medvet, and Enrico Sorio. Improving features extraction for supervised invoice classification. In *Proceedings of the 10th IASTED International Conference*, volume 674, page 401, 2010.
- [6] Yolande Belaïd and Abdel Belaïd. Morphological tagging approach in document analysis of invoices. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 1, pages 469–472. IEEE, 2004.
- [7] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In Neil D. Lawrence and Mark Girolami, editors, *Proceedings*

- of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 127–135, La Palma, Canary Islands, 21–23 Apr 2012. PMLR.
- [8] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI’11, pages 301–306. AAAI Press, 2011.
- [9] Thomas M. Breuel. The ocropus open source ocr system, 2008.
- [10] F. Cesarini, E. Francesconi, M. Gori, and G. Soda. Analysis and understanding of multi-class invoices. *Document Analysis and Recognition*, 6(2):102–114, Oct 2003.
- [11] Francesca Cesarini, Marco Gori, Simone Marinai, and Giovanni Soda. Informys: A flexible invoice-like form-reader system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):730–745, 1998.
- [12] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.
- [13] Klemens Muthmann Michael Berger Alexander Schill Daniel Esser, Daniel Schuster. Automatic indexing of scanned documents: a layout-based approach, 2012.
- [14] F. Deckert, B. Seidler, M. Ebbecke, and M. Gillmann. Table content understanding in smartfix. In *2011 International Conference on Document Analysis and Recognition*, pages 488–492, Sept 2011.
- [15] Andreas R. Dengel. Making documents work: Challenges for document understanding. In *Seventh International Conference on Document Analysis and Recognition*. IEEE, 2003.
- [16] Andreas R. Dengel and Bertin Klein. smartfix: A requirements-driven system for document analysis and understanding. In Daniel Lopresti, Jianying Hu, and Ramanujan Kashi, editors, *Document Analysis Systems V*, pages 433–444, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
- [17] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011.

- [18] Hatem Hamza, Yolande Belaïd, and Abdel Belaïd. Case-based reasoning for invoice analysis and recognition. In *International conference on case-based reasoning*, pages 404–418. Springer, 2007.
- [19] Bill Janssen, Eric Saund, Eric Bier, Patricia Wall, and Mary Ann Sprague. Receipts2go: the big world of small documents. In *Proceedings of the 2012 ACM symposium on Document engineering*, pages 121–124. ACM, 2012.
- [20] Rodolphe Jenatton, Nicolas Le Roux, Antoine Bordes, and Guillaume Obozinski. A latent factor model for highly multi-relational data. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’12, pages 3167–3175, USA, 2012. Curran Associates Inc.
- [21] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’15, pages 3294–3302, Cambridge, MA, USA, 2015. MIT Press.
- [22] Bertin Klein, Stevan Agne, and Andreas Dengel. Results of a study on invoice-reading systems in germany. In *International workshop on document analysis systems*, pages 451–462. Springer, 2004.
- [23] Stefan Klink, Andreas Dengel, and Thomas Kieninger. Document structure analysis based on layout and textual features. In *Proc. of International Workshop on Document Analysis Systems, DAS2000*, pages 99–111. Citeseer, 2000.
- [24] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *CoRR*, abs/1607.05368, 2016.
- [25] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(1-3):503–528, August 1989.
- [26] Toshiko Matsumoto, Mitsuharu Oba, and Takashi Onoyama. Sample-based collection and adjustment algorithm for metadata extraction parameter of flexible format document. In *International Conference on Artificial Intelligence and Soft Computing*, pages 566–573. Springer, 2010.

- [27] Eric Medvet, Alberto Bartoli, and Giorgio Davanzo. A probabilistic approach to printed document understanding. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(4):335–347, Dec 2011.
- [28] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [29] Opentext. Opentext capture center. <http://www.opentext.com/2/global/products/products-capture-and-imaging/products-opentext-capture-center.htm>, 2012, 2012.
- [30] Rasmus Berg Palm, Ole Winther, and Florian Laws. Cloudscan-a configuration-free invoice analysis system using recurrent neural networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 406–413. IEEE, 2017.
- [31] Gaurav Pandey and Rakshit Daga. On extracting structured knowledge from unstructured business documents. In *Proc IJCAI Workshop on Analytics for Noisy Unstructured Text Data*, pages 155–162, 2007.
- [32] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
- [33] M. Rusiol, T. Benkhelfallah, and V. P. dAndecy. Field extraction from administrative documents by incremental structural templates. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1100–1104, Aug 2013.
- [34] Hiroshi Sako, Minenobu Seki, Naohiro Furukawa, Hisashi Ikeda, and Atsuhiko Imaizumi. Form reading based on form-type identification and form-data recognition. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 926–930. IEEE, 2003.
- [35] D. Schuster, K. Muthmann, D. Esser, A. Schill, M. Berger, C. Weidling, K. Aliyev, and A. Hofmeier. Intellix – end-user trained information extraction for document archiving. In *2013 12th International Conference on Document Analysis and Recognition*, pages 101–105, Aug 2013.
- [36] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q.

- Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 926–934. Curran Associates, Inc., 2013.
- [37] Enrico Sorio, Alberto Bartoli, Giorgio Davanzo, and Eric Medvet. Open world classification of printed invoices. In *Proceedings of the 10th ACM symposium on Document engineering*, pages 187–190. ACM, 2010.
- [38] Enrico Sorio, Alberto Bartoli, Giorgio Davanzo, and Eric Medvet. A domain knowledge-based approach for automatic correction of printed invoices. In *Information Society (i-Society), 2012 International Conference on*, pages 151–155. IEEE, 2012.
- [39] Niket Tandon, Gerard De Melo, and Gerhard Weikum. Deriving a web-scale common sense fact database. In *AAAI*, 2011.
- [40] Leon Todoran, Marco Aiello, Christof Monz, and Marcel Worring. Logical structure detection for heterogeneous document classes. In *Document Recognition and Retrieval VIII*, volume 4307, pages 99–111. International Society for Optics and Photonics, 2000.