

**Project Title: Information Extractor for Traffic Court Cases**

**Interim Report**

**Student Name: xxx**

**School of Computer Science and Enginering**

**Supervisor: A/P Yeo Chai Kiat**

Table of Contents

[Objectives and Scope 3](#_Toc507627875)

[Project Description 3](#_Toc507627876)

[Implementation 3](#_Toc507627877)

[PDF to text 3](#_Toc507627878)

[Extracting Plaintiff’s Name 3](#_Toc507627879)

[Extracting Plaintiff’s Age 3](#_Toc507627880)

[Extracting Plaintiff’s Gender 4](#_Toc507627881)

[Extracting Date of Assessment 4](#_Toc507627882)

[Accomplishments and Problems 4](#_Toc507627883)

[PDF to Text 4](#_Toc507627884)

[Extracting Plaintiff’s Name 4](#_Toc507627885)

[Extracting Plaintiff’s Age 4](#_Toc507627886)

[Extracting Plaintiff’s Gender 4](#_Toc507627887)

[Extracting Date of Assessment 4](#_Toc507627888)

[Works to be done 5](#_Toc507627889)

[January 5](#_Toc507627890)

[February 5](#_Toc507627891)

# Objectives and Scope

To develop an information extractor that extracts important information from uploaded traffic accident court cases and a web portal that returns traffic accident information stored in database when searched by users.

# Project Description

The information extractor allows users to upload past traffic accident court cases and it will extract important information such as date of assessment, plaintiff’s name, gender, age, injuries and claimable amount. All extracted information will then be stored in a database and ready to be used by the client web portal. The client web portal allows users to search for injuries regarding traffic accidents.

# Implementation

## PDF to text

Pdfminer.six will be used to process uploaded court case and converts it to text form.

## Extracting Plaintiff’s Name

Algorithm 1:

1. Tokenise the converted text into sentences.
2. Scan the sentences and look out for the word “victim” or “plaintiff”
3. Using Name Entity Recognition library, process those sentences.
4. Filter the obtained results once more by checking each individual word with the dictionary library.
5. Name chunks with at least one word that is not a valid English word will be treated as a valid name.

Algorithm 2:

1. Based on observations, each page on a traffic accident court case will have a name pattern example “Name 1 V Name 2”, where Name 1 is the plaintiff’s name. Therefore, by using regular expressions we can extract the plaintiff’s name more accurately.

## Extracting Plaintiff’s Age

Algorithm:

1. Tokenise the converted text into sentences.
2. Scan the sentences and look out for words like “years old” and “at the time”.
3. From the sentences, extract all the numbers from the sentences and get the maximum.

## Extracting Plaintiff’s Gender

Algorithm:

1. Scan the converted text for “his/he” and “her/she”
2. Increase the count for male whenever the word “his” or “he” is identified and increase the count for female whenever the word “her” or “she” is identified.
3. The gender with higher count will be taken into conclusion.

## Extracting Date of Assessment

Algorithm:

1. Using regular expressions, extract all the dates that occurs in the converted text.
2. Filter the obtained results and get the latest date.

# Accomplishments and Problems

## PDF to Text

Accomplishments: The PDF can be successfully converted to text without any loss of information.

Problems: During the conversion, it is unable to detect tables and properly process tabular data. Therefore, resulting in messy extraction for tables.

## Extracting Plaintiff’s Name

Accomplishments: Plaintiff’s name can be successfully extracted without fail for all cases.

Problems: When using algorithm 1, all the names from the processed sentences are being retrieved. However, in some cases, unnecessary noise like NRIC and company name are also being retrieved.

Solution: By using algorithm 2, it can successfully extract the plaintiff’s name without any problem.

## Extracting Plaintiff’s Age

Accomplishments: Plaintiff’s age can be successfully extracted without fail for all cases.

Problems: The court document might include age of more than one person.

Solution: By filtering for the words “plaintiff” and “at the time of hearing”.

## Extracting Plaintiff’s Gender

Accomplishments: Plaintiff’s gender can be successfully extracted without fail for all cases.

Problems: NIL

Solution: NIL

## Extracting Date of Assessment

Accomplishments: Date of Assessment can be successfully extracted without fail for all cases.

Problems: The court document contains several dates.

Solution: Based on observations on all cases, the latest date in the court document is always the date of assessment.

# Future Work

## January

1. Develop an algorithm to extract injuries of plaintiff.
2. Develop an algorithm to extract payable amount of each injury.
3. Design and create a database to stored extracted information.

## February

1. Develop a web portal for users using Python Django.
2. Integration of Information extractor, database and web portal.
3. Testing