Text Analytics for a Real Estate Service Provider

Digital Services

Success Story

www.indiumsoftware.com

Client

The client has a dedicated real estate division to provide range of services to help their customers (real estate property owners) to make better informed decisions, regardless of their position in the real estate pipeline. Being able to put themselves into their customers' shoes, enables the client to efficiently to design and provide services to real estate property owners in U.S

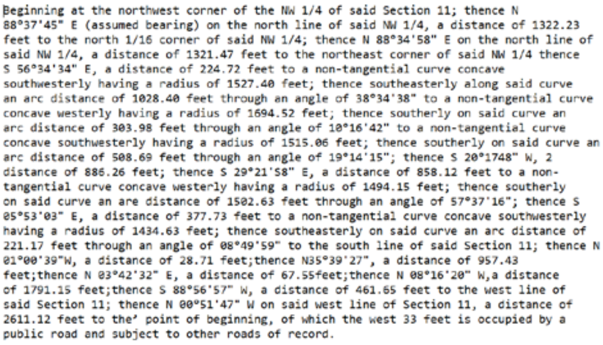
Application Overview

One of the client's processes is to extract the boundary description of the plots from a huge number of PDF deed and lease documents. These manually extracted information were then fed into the drawing software to get the required output. Indium leveraged text analytics method (RegEX, NER) to automate the text extraction process despite the challenges faced in the text extraction process. The client is now able to perform the text extraction with involving only negligible amount of manual time.

Business Requirements

The boundary description of the plots for deed and lease documents is written in Metes and Bounds format. Although still prevalent, it is an archaic verbose format which is not amenable to be used by modern drawing software like ArcGIS, CAD etc. These tools require input in a shorthand format often called traverse files. The conversion from Metes and Bounds to traverse format is generally done manually. This involves lot of manual effort and it takes many hours to generate the required input file.

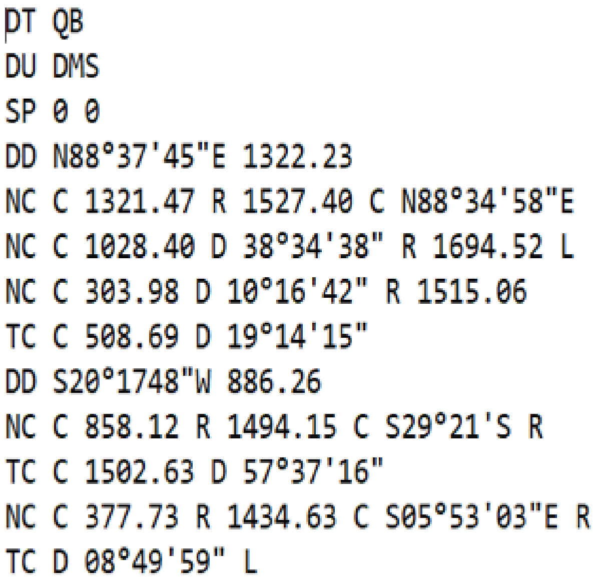
Snapshot of Metes and Bound format (Raw format)



Beginning at the northwest corner of the NW 1/4 of said Section 11; thence N 88°37'45" E (assumed bearing) on the north line of said NW 1/4, a distance of 1322.23 feet to the north 1/16 corner of said NW 1/4; thence N 88°34'58" E on the north line of said NW 1/4, a distance of 1321.47 feet to the northeast corner of said NW 1/4 thence S 56°34'34" E, a distance of 224.72 feet to a non-tangential curve concave southwesterly having a radius of 1527.40 feet; thence southeasterly along said curve an arc distance of 1028.40 feet through an angle of 38°34'38" to a non-tangential curve concave westerly having a radius of 1694.52 feet; thence southerly on said curve an arc distance of 303.98 feet through an angle of 10°16'42" to a non-tangential curve concave southwesterly having a radius of 1515.06 feet; thence southerly on said curve an arc distance of 508.69 feet through an angle of 19°14'15"; thence S 20°1748" W, 2 distance of 886.26 feet; thence S 29°21'58" E, a distance of 858.12 feet to a non-tangential curve concave westerly having a radius of 1494.15 feet; thence southerly on said curve an are distance of 1502.63 feet through an angle of 57°37'16"; thence S 05°53'03" E, a distance of 377.73 feet to a non-tangential curve concave southwesterly having a radius of 1434.63 feet; thence southeasterly on said curve an arc distance of 221.17 feet through an angle of 08°49'59" to the south line of said Section 11; thence N 01°00'39"W, a distance of 28.71 feet;thence N35°39'27", a distance of 957.43 feet;thence N 03°42'32" E, a distance of 67.55feet;thence N 08°16'20" W,a distance of 1791.15 feet;thence S 88°56'57" W, a distance of 461.65 feet to the west line of said Section 11; thence N 00°51'47" W on said west line of Section 11, a distance of 2611.12 feet to the' point of beginning, of which the west 33 feet is occupied by a public road and subject to other roads of record.

Snapshot Travese Format

(Required input format to be fed into the drawing softwares)



```
DT QB
DU DMS
SP 0 0
DD N88°37'45"E 1322.23
NC C 1321.47 R 1527.40 C N88°34'58"E
NC C 1028.40 D 38°34'38" R 1694.52 L
NC C 303.98 D 10°16'42" R 1515.06
TC C 508.69 D 19°14'15"
DD S20°1748"W 886.26
NC C 858.12 R 1494.15 C S29°21'S R
TC C 1502.63 D 57°37'16"
NC C 377.73 R 1434.63 C S05°53'03"E R
TC D 08°49'59" L
```

Business

Text Analytics

Domain

Real Estate

Tools

Pytesseract, Poppler, Align, RegEx, NER ,Python (pycrfcuite, BeautifulSoup, OpenCV, Tabula, xPDF), GATE (General Architecture for Text Engineering) training data preparation

Key Highlights

- Automate the process of traverse file creation reducing the manual time to almost zero
- 83% increase in the number of input documents
- 100% text extraction achieved

Goal

Reduce traverse file creation time by automating the process of Metes and Bounds to traverse file conversion.

Challenges

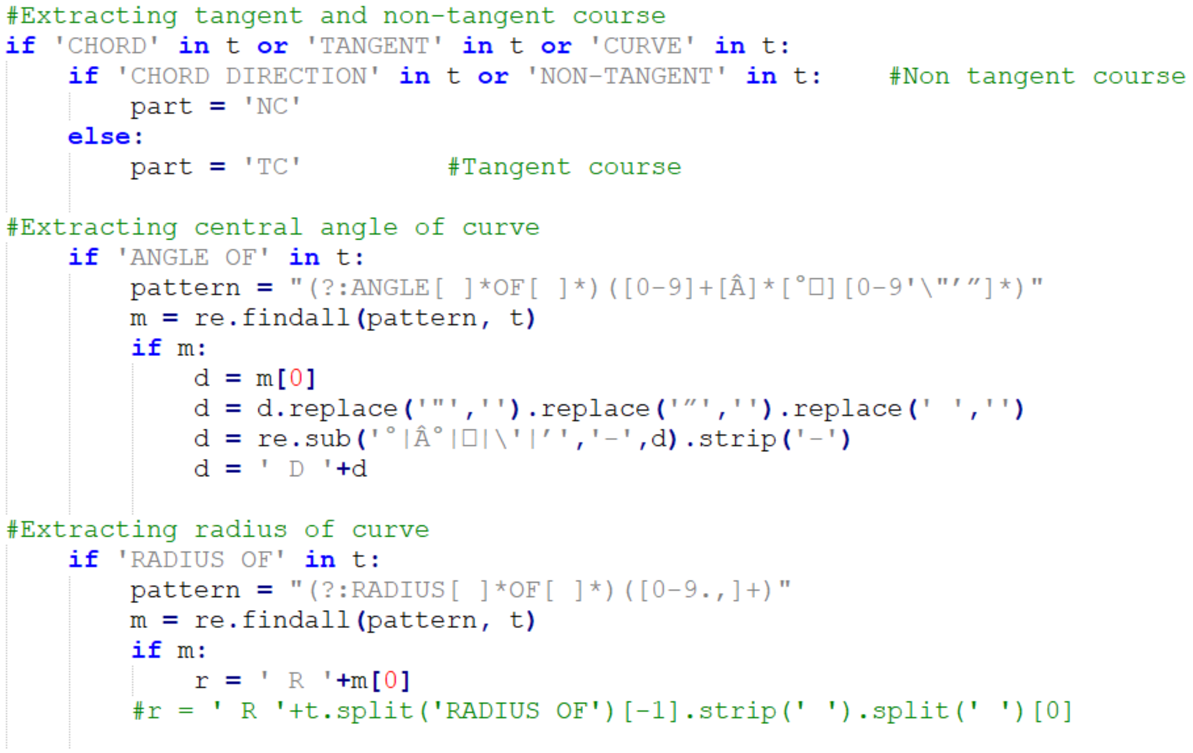
- Metes and Bounds deeds were present as PDF files containing multiple variations such as
  - Scanned copies of deeds saved as PDFs (often with poor lighting)
  - Rotated text
  - Highlighted (with marker) text
  - Watermark in the background

  Converting such PDFs to text files led to significant upfront data loss and OCR error.
- Multiple types of curve components which can occur randomly w/o a fixed pattern. Hence, writing a logical rule is impossible.

Indium Software's Approach and Implementation

- Indium created a solution using OpenCV and Deep Learning based methods to pre-process PDFs (rotation, highlighting correction etc) and convert them to text.
- Indium implemented two approaches to automate the conversion of text file to traverse file
  - Regular Expression - Identify curve components and their values by finding occurrence of keywords using regular expression patterns e.g. Chord Direction or Non-Tangent for a Non-Tangential Curve

Regular Expression

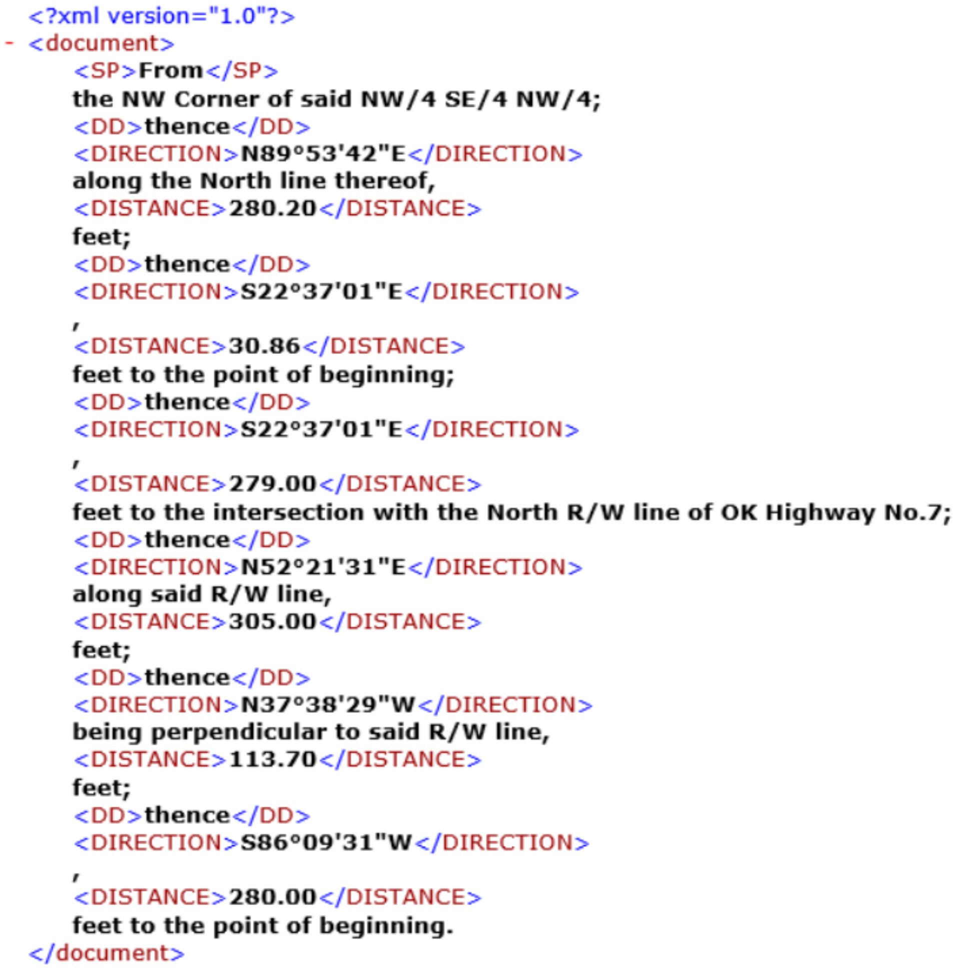


```
#Extracting tangent and non-tangent course
if 'CHORD' in t or 'TANGENT' in t or 'CURVE' in t:
    if 'CHORD DIRECTION' in t or 'NON-TANGENT' in t:     #Non tangent course
        part = 'NC'
    else:
        part = 'TC'          #Tangent course

#Extracting central angle of curve
    if 'ANGLE OF' in t:
        pattern = "(?:ANGLE[ ]*OF[ ]*)([0-9]+[Â]*[°□][0-9'\"′″]*)"
        m = re.findall(pattern, t)
        if m:
            d = m[0]
            d = d.replace('"','').replace('″','').replace(' ','')
            d = re.sub('°|Â°|□|\'|′','-',d).strip('-')
            d = ' D '+d

#Extracting radius of curve
    if 'RADIUS OF' in t:
        pattern = "(?:RADIUS[ ]*OF[ ]*)([0-9.,]+)"
        m = re.findall(pattern, t)
        if m:
            r = ' R '+m[0]
        #r = ' R '+t.split('RADIUS OF')[-1].strip(' ').split(' ')[0]
```

Although quite effective, this approach needed manual intervention to define rules. Moreover, defining a comprehensive set of rules with high accuracy is tricky and prone to manual errors. Hence, Indium came up with another method by using Conditional Random Field.

- Machine Learning based tagging using Conditional Random Field
    - Conditional Random Fields is a sequence-based machine learning algorithm. The text in the text file is considered as a sequence of tokens. Based on the token and features defined around it (what is the 3rd last word, what is the 2nd last word, is the token an adjective, does it end with a '?' mark etc.), CRF learns optimum weights for each feature and hence predicts the probability of each token belonging to a certain tag.
    - A training data is created using GATE tool. GATE tool enables the user to highlight selected text token and give a tag to it. All the tokens of interest are highlighted and tagged. GATE outputs an XML file which is used in training and looks as shown below.

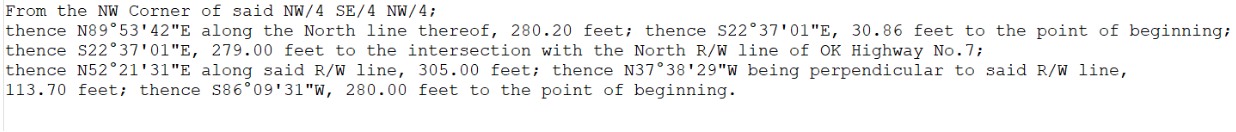
Training Data



```
  <?xml version="1.0"?>
- <document>
    <SP>From</SP>
    the NW Corner of said NW/4 SE/4 NW/4;
    <DD>thence</DD>
    <DIRECTION>N89°53'42"E</DIRECTION>
    along the North line thereof,
    <DISTANCE>280.20</DISTANCE>
    feet;
    <DD>thence</DD>
    <DIRECTION>S22°37'01"E</DIRECTION>
    ,
    <DISTANCE>30.86</DISTANCE>
    feet to the point of beginning;
    <DD>thence</DD>
    <DIRECTION>S22°37'01"E</DIRECTION>
    ,
    <DISTANCE>279.00</DISTANCE>
    feet to the intersection with the North R/W line of OK Highway No.7;
    <DD>thence</DD>
    <DIRECTION>N52°21'31"E</DIRECTION>
    along said R/W line,
    <DISTANCE>305.00</DISTANCE>
    feet;
    <DD>thence</DD>
    <DIRECTION>N37°38'29"W</DIRECTION>
    being perpendicular to said R/W line,
    <DISTANCE>113.70</DISTANCE>
    feet;
    <DD>thence</DD>
    <DIRECTION>S86°09'31"W</DIRECTION>
    ,
    <DISTANCE>280.00</DISTANCE>
    feet to the point of beginning.
  </document>
```

- The model is written in Python using pycrfsuite library. BeautifulSoup library is used to parse the XML. The new incoming text file is passed through the trained model to tag each token as a curve component or NA (not belonging to any entity). The tokens with entity tags are filtered out and then combined properly to create traverse file.
- The result of the tagging from the CRF model look as shown below

Input

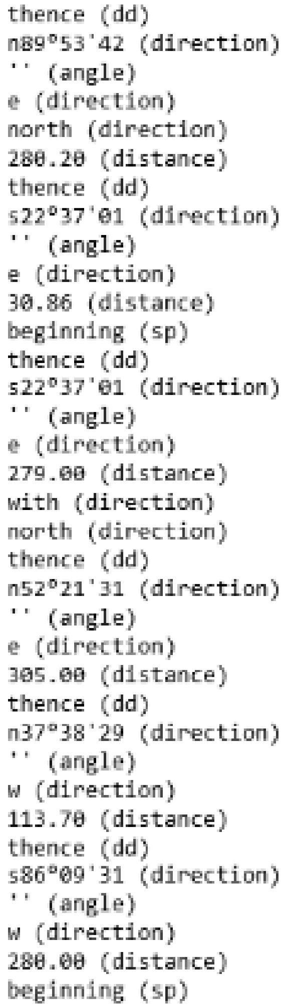


```
From the NW Corner of said NW/4 SE/4 NW/4;
thence N89°53'42"E along the North line thereof, 280.20 feet; thence S22°37'01"E, 30.86 feet to the point of beginning;
thence S22°37'01"E, 279.00 feet to the intersection with the North R/W line of OK Highway No.7;
thence N52°21'31"E along said R/W line, 305.00 feet; thence N37°38'29"W being perpendicular to said R/W line,
113.70 feet; thence S86°09'31"W, 280.00 feet to the point of beginning.
```

Business Impact

- Automate the process of traverse file creation reducing the manual time to almost zero.
  Text extraction was done on 100% of the PDF documents despite the watermarks andhighlighted text which usually causes issues while text extraction.
- 100% Conversion pipeline between human friendly Metes and Bounds format and machine friendly Traverse files.
- 83% increase in the number input documents fed into the drawing software per day.

Output



```
thence (dd)
n89°53'42 (direction)
'' (angle)
e (direction)
north (direction)
280.20 (distance)
thence (dd)
s22°37'01 (direction)
'' (angle)
e (direction)
30.86 (distance)
beginning (sp)
thence (dd)
s22°37'01 (direction)
'' (angle)
e (direction)
279.00 (distance)
with (direction)
north (direction)
thence (dd)
n52°21'31 (direction)
'' (angle)
e (direction)
305.00 (distance)
thence (dd)
n37°38'29 (direction)
'' (angle)
w (direction)
113.70 (distance)
thence (dd)
s86°09'31 (direction)
'' (angle)
w (direction)
280.00 (distance)
beginning (sp)
```

☎

INDIA

Chennai | Bengaluru | Mumbai
Toll-free: 1800-123-1191

USA

Cupertino | Princeton
Toll-free: 1 888 207 5969

UK

London

SINGAPORE

+65 9630 7959

General Inquiries
info@indiumsoftware.com

Sales Inquiries
sales@indiumsoftware.com