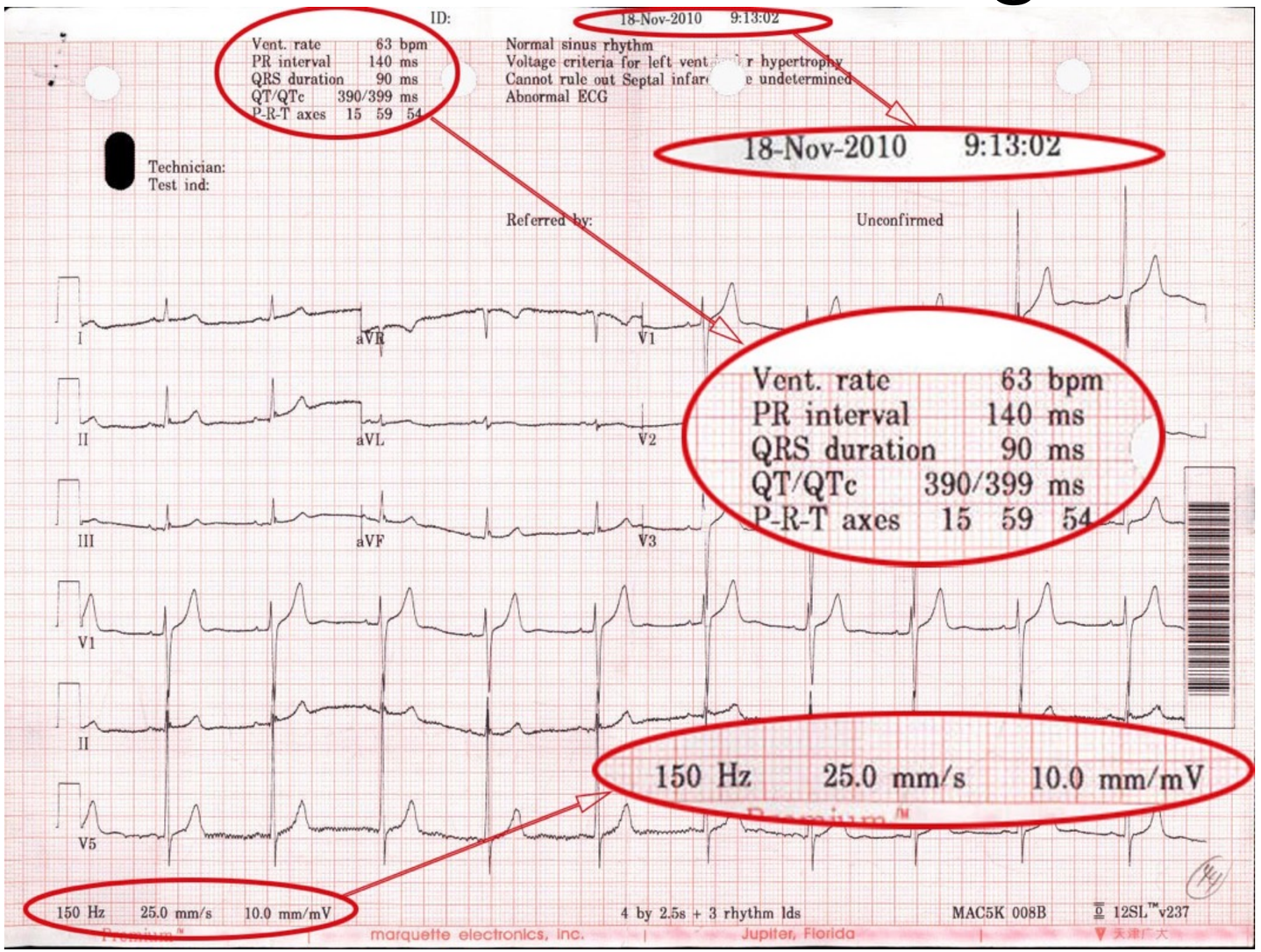
Fault-tolerant Optical Character Recognition from Semi-structured Medical Images

Jinyi Lu
JinyiLu93@gmail.com
2015/03/25

What are Semi-structured Medical Images?

- Electro-cardio-gram (ECG)



What's Fault-tolerate OCR?

- Optical Character Recognition (OCR)
  - Images of typewritten or printed text -> Machine-encoded text
- Fault-tolerate OCR

Motivation

- More than 45GB ECG images from AZ in different forms

Outline

- Input
- ODL
- Manual Correction
- Experimental Results

Outline

- Input
- ODL
- Manual Correction
- Experimental Results

Input

- XML
  - text, coordinates, layout
- Tesseract
- text_box = {c=coor, t=v}

```
<p class="ocr_par" title="box 263 33 444 119">
   <span class="ocr_l" title="box 264 33 336 45">
      <span class="ocrx_w" title="box 264 33 299 45">Vcnt.</span>
      <span class="ocrx_w" title="box 308 34 336 45">rule</span>
   </span>
   <span class='ocr_l'>
      <span class="ocrx_w" title="box 264 51 283 64">PR</span>
      <span class="ocrx_w" title="box 291 51 346 64">Interval</span>
      <span class="ocrx_w" title="box 389 52 411 64">140</span>
      <span class="ocrx_w" title="box 420 55 439 64">ms</span>
   </span>
   ...
   </span>
</p>
<p class="ocr_p" dir="ltr">
   <span class="ocr_l">
      <span class="ocrx_w" title="box 396 33 411 45">53</span>
      <span class="ocrx_w" title="box 420 33 449 48">bpm</span>
   </span>
</p>
```

Outline

- Input
- ODL
  - OCR description language (ODL)
  - Syntax
  - Semantics
- Manual Correction
- Experimental Results

Outline

- Input
- ODL
  - OCR description language (ODL)
  - Syntax
  - Semantics
- Manual Correction
- Experimental Results

Values

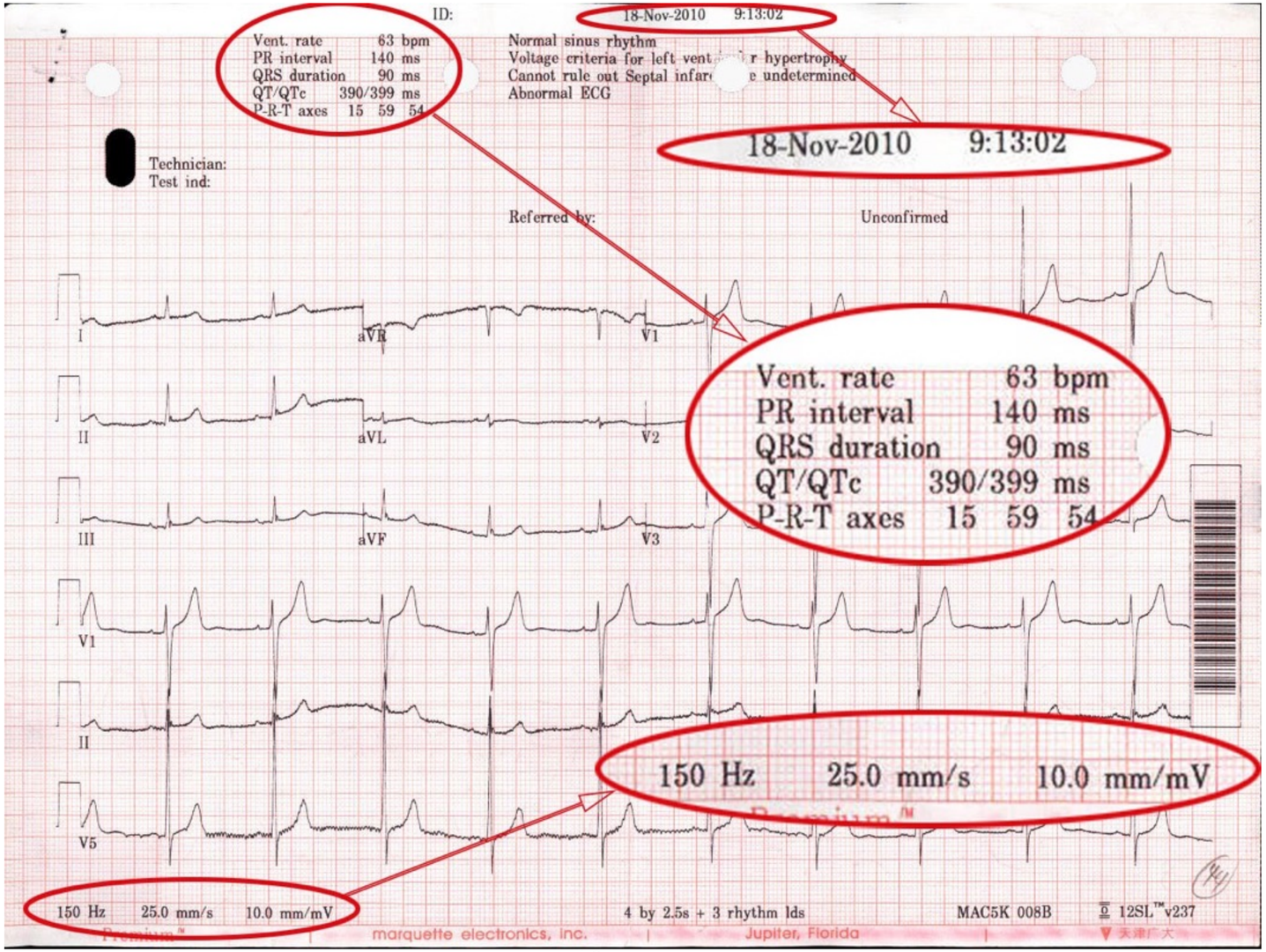
- All the expressions can be parsed into values

$$
\begin{aligned}
\mathbf{v} ::= & () && \text{hskip \s} \\
& | int \\
& | float \\
& | string \\
& | len && \text{100 pixel, 100 cm} \\
& | coor && \text{<_,0.5l, 300 pixel, 600 pixel>} \\
& | \{v_1, ..., v_n\}
\end{aligned}
$$

Abstract Syntax

$e ::= c$	(constant)	“Vent.”
$\mid x$	(name)	x
$\mid e_0(e_1, e_2, ..., e_n)$	(constraints)	x(int), x(<…>)
$\mid hskip\ \ e$	(horizontal skip)	hskip \s
$\mid vskip\ \ e$	(vertical skip)	vskip 10 pixel
$\mid \{e_1 \mid e_2 \mid ... \mid e_n\}$	(union)	{“Jan”|“Feb”}
$\mid \{e_1, ..., e_n\}$	(struct)	{“rate”, x(int)}
$\mid e\ \ list$	(list)	
$\mid e\ \ as\ \ x$	(blinding)	
$\mid e_1\ \ bop\ \ e_2$	(binary operation)	

Example


ID:
18-Nov-2010 9:13:02
Vent. rate 63 bpm
PR interval 140 ms
QRS duration 90 ms
QT/QTc 390/399 ms
P-R-T axes 15 59 54
Normal sinus rhythm
Abnormal ECG
Technician:
Test ind:
Referred by:
Unconfirmed
18-Nov-2010 9:13:02
Vent. rate 63 bpm
PR interval 140 ms
QRS duration 90 ms
QT/QTc 390/399 ms
P-R-T axes 15 59 54
I
aVR
V1
II
aVL
V2
III
aVF
V3
V1
II
V5
150 Hz 25.0 mm/s 10.0 mm/mV
150 Hz 25.0 mm/s 10.0 mm/mV
4 by 2.5s + 3 rhythm lds
MAC5K 008B
12SL™v237
marquette electronics, inc.
Jupiter, Florida

Description (Abstract Syntax)

```
{
    {
        day(int, 1, 31),
        "_",
        {
            num(int, 1, 12) |
            {
                "Jan" | "Feb" | "Mar" |
                "Apr" | "May" | "Jun" |
                "Jul" | "Aug" | "Sept" |
                "Oct" | "Nov" | "Dec"
            }
        }
        "_",
        year(int)
    }(<_,_,_,0.31>),

    ...
}
```

Description (Surface Syntax)

```
Osource Ostruct entry_t{
    time_t(<_,_,_,0.31>)  time;
    triple_t(<0.1w,_,0.5w,_>)  tri;
    inter_t(<tri.x1,tri.y0,_,_>)  i;
    vskip(\n)[]  skipline;
    parameter_t(<_,_,_,_>)  para;
};
```

```
Ounion month_str{
    "Jan"; "Feb"; "Mar";
    "Apr"; "May"; "Jun";
    "Jul"; "Aug"; "Sept";
    "Oct"; "Nov"; "Dec";
};

Ounion month_t{
    Oint(1,12)  num;
    month_str str;
};

Ostruct time_t{
    Oint(1,31)  day;
    "_";
    month_t month;
    "_";
    Oint()     year;
};
```

Description (Surface Syntax)

```
Ostruct triple_t{
    "Vent. rate";
    hskip(\s)    skip;
    Oint(60,100)     x;
    "bpm";
};
```

```
Ounion inter_t{
    "Normal ECG";
    "Abnormal ECG";
};
```

```
Ostruct parameter_t{
    Oint()   p1;
    "Hz";
    Ofloat(3, 1)     p2;
    "mm/s";
    Ofloat(3, 1)     p3;
    "mm/mV";
};
```

Type System

$$
\begin{aligned}
\mathbf{t} ::= \quad & unit \\
& |int \\
& |float \\
& |len \\
& |\langle len,\ len,\ len,\ len\rangle \\
& |\{t_1 + t_2 + ... + t_n\} \\
& |\{l_1 : t_1,\ ...,\ l_n : t_n\} \\
& |t\ list
\end{aligned}
$$

$$\frac{\Gamma(x) = t}{\Gamma \vdash x : t} \quad \text{(T-VARIABLE)}$$

$$\frac{\Gamma \vdash e_1 : int\ \ \Gamma \vdash e_2 : int\ \ bop \in \{+, -, *, /, \%\}}{\Gamma \vdash e_1\ \ bop\ \ e_2 : int} \quad \text{(T-INT ARITH)}$$

$$\frac{\Gamma \vdash e_1 : int\ \ \Gamma \vdash e_2 : int\ \ bop \in \{=, !=, <, >, <=, >=\}}{\Gamma \vdash e_1\ \ bop\ \ e_2 : bool} \quad \text{(T-INT REL)}$$

$$\frac{\Gamma \vdash e_1 : float\ \ \Gamma \vdash e_2 : float\ \ bop \in \{+, -, *, /, \%\}}{\Gamma \vdash e_1\ \ bop\ \ e_2 : float} \quad \text{(T-FLOAT ARITH)}$$

$$\frac{\Gamma \vdash e_1 : float\ \ \Gamma \vdash e_2 : float\ \ bop \in \{=, !=, <, >, <=, >=\}}{\Gamma \vdash e_1\ \ bop\ \ e_2 : float} \quad \text{(T-FLOAT REL)}$$

$$\frac{\Gamma \vdash e_0 : t_0}{\Gamma \vdash e_0(e_1, e_2, ..., e_n) : t_0} \quad \text{(T-CONSTRAINT)}$$

$$\frac{\Gamma \vdash e : len}{\Gamma \vdash hskip\ \ e : unit} \quad \text{(T-HSKIP)}$$

$$\frac{\Gamma \vdash e : len}{\Gamma \vdash vskip\ \ e : unit} \quad \text{(T-VSKIP)}$$

$$\frac{for\ \ each\ \ i\ \ \Gamma \vdash e_i : t_i}{\Gamma \vdash \{e_1|...|e_n\} : t_1 + ... + t_n} \quad \text{(T-UNION)}$$

$$\frac{for\ \ each\ \ i\ \ \Gamma \vdash e_i : t_i}{\Gamma \vdash \{e_1, ..., e_n\} : t_1 * ... * t_n} \quad \text{(T-STRUCT)}$$

$$\frac{\Gamma \vdash e : t}{\Gamma \vdash e\ \ list : t\ \ list} \quad \text{(T-LIST)}$$

Outline

- Input
- ODL
  - OCR description language (ODL)
  - Syntax
  - Semantics
- Manual Correction
- Experimental Results

Semantics

- Automatically generate the most reliable parsing results
  - Input data and parsing results
  - Scoring policy
  - Evaluation rules

Input Data and Parsing Results

$$
\begin{aligned}
\text{text\_box} ::= & \{c = \text{coor}, t = v\} \\
& \mid \{\} \\
\text{parse\_tree} ::= & ((e, text\_box), [parse\_tree_1, ..., parse\_tree_n]) \\
& \mid () \\
\text{input\_data} ::= & \{text\_box \mid \ \forall text\_box \ \ in \ \ XML\}
\end{aligned}
$$

```
time
  day                F     "18"
  "_"                F
  month
  str                F     "Nov"
  "_"                F
  year               F     "2010"

triple
  "Vent. rate"       E     "Vcnt. rule"
  skip1              F
  x                  E     "53"
  "bpm"              F
  skip2              F

inter
  "Abnormal ECG"     F

para
  p1                 E     "15o"
  "Hz"               F
  p2                 F     "25.0"
  "mm/s"             F
  p3                 E     "1o.o"
  "mm/mV"            F
```

Scoring Policy

- Score for how likely a text box is matched with the expression
- Why?
  - Errors and noises in the OCR results
  - Rough description
- Two kinds
  - Data description
  - Spatial description

Data description

- To tolerate the errors and noises in the OCR results

$$es(c,t) = ed(c,t)$$

$$es(x(int,a,b),t) = \min\{ed(i,t)|\ \ \forall i \in Z, i \in [a,b]\}$$

$$es(x(float,l,p,a,b),t) = \min\{ed(i,t)|\ \ \forall i \in R', i \in [a,b]\}$$

Data description

“Vent. rate”	“Vcnt. rule”	3
x(int, 60, 100)	53	1
y(float, 3, 1)	1o.o	2

Scoring Policy

- Spatial description
  - Spatial description is used to describe the rough spatial relationship based on human estimates
  - cc: the position of the current cursor

$$ss(cc, coor) = ||cc - coor||$$

Scoring Policy

score(e, tb, cc) = es(e, tb.t) +k * ss(cc, tb.c)

Judgment Form

- E: environment
  - cc: the position of the current cursor
- D: input data
- D’: remaining data

$$\text{Judgment Form}: \ E, D; e \Downarrow (D'; parse\_tree)list$$

Judgment Form

- Why list?
  - Scoring policy: lots of satisfied text box
  - All candidate parsing results
  - Final result: sum up the scores of all leaf nodes

Evaluation Rules

- Examples
  - {“Vent. rate”, x(int, 60, 100)}
  - {{<…>, “Vcnt. rule”}, {<…>, “b”}, {<…>, “53”}}
- Try all the candidates

$$\frac{}{E, \{\}; x \Downarrow []} \quad \text{(E-EMPTY)}$$

$$\frac{p \in D \;\; E(c) = coor \;\; Cons(p, coor, const) = true \;\; E, (D - p); x \Downarrow r}{E, D; const \Downarrow ((D - p); (const, p), []) :: r} \quad \text{(E-C1)}$$

$$\frac{p \in D \;\; E(c) = coor \;\; Cons(p, coor, const) = false \;\; E, (D - p); x \Downarrow r}{E, D; const \Downarrow (D; (const, \{\}), []) :: r} \quad \text{(E-C2)}$$

$$\frac{p \in D \;\; E(c) = coor \;\; E, (D - p); x \Downarrow r}{E, D; x \Downarrow ((D - p); (x, p), []) :: r} \quad \text{(E-X)}$$

$$\frac{E, D; e_0 \Downarrow r_0 \quad \forall i : r_{i+1} = Filter(r_i, e_i)}{E, D; e_0(e_1, e_2, ..., e_n) \Downarrow r_{n+1}} \quad \text{(E-CONSTRAINT)}$$

$$\frac{Find(D, coor, nil) = D' \;\; E[c \mapsto Begin(D')], D'; e \Downarrow (D''; t)list}{E, D; e(coor) \Downarrow (D - D' + D''; t)list} \quad \text{(E-COOR)}$$

$$\frac{E, D; e \Downarrow r \;\; E(c) = coor \;\; Hskip(D, coor, r) = D'}{E, D; hskip \;\; e \Downarrow (D'; []) :: []} \quad \text{(E-HSKIP)}$$

$$\frac{E, D; e \Downarrow r \;\; E(c) = coor \;\; Vskip(D, coor, r) = D'}{E, D; vskip \;\; e \Downarrow (D'; []) :: []} \quad \text{(E-VSKIP)}$$

$$\frac{E, D; e_1 \Downarrow r_1 \;\; E, D; \{e_2|..|e_n\} \Downarrow r_2}{E, D; \{e_1|e_2|...|e_n\} \Downarrow r_1 + r_2} \quad \text{(E-UNION)}$$

$$\frac{E, D; e_1 \Downarrow (E', D'; parse\_tree)list \quad \forall i : E'_i[c \mapsto End(D - D'_i)], D'_i; \{e_2, ..., e_n\} \Downarrow r_i}{E, D; \{e_1, e_2, ...e_n\} \Downarrow \sum_{i=1}^{m} (D'; parse\_tree)list * r_i} \quad \text{(E-STURCT)}$$

$$\frac{E, D; head \;\; e \;\; list \Downarrow (E', D'; parse\_tree)list \quad \forall i : E'_i[c \mapsto End(D - D'_i)], D'_i; tail \;\; e \;\; list \Downarrow r_i}{E, D; e \;\; list \Downarrow \sum_{i=1}^{m} (D'; parse\_tree)list * r_i} \quad \text{(E-LIST)}$$

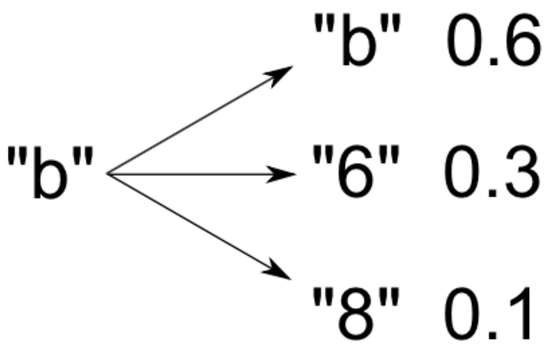
Outline

- Input
- ODL
- Manual Correction
  - Incremental Learning Correction Model
  - Manual Correction Policy
- Experimental Results

Correction Model

- Correction strategies S
  - m: original substring
  - n: candidate substring after correction
  - A: set of the correct candidates and probabilities

$$S = \{(m, n, p) |\ \forall (n, p) \in A\}$$



- Correction model M

New Score Policy

$$cor(t, S) = \{(t', p) |\ \forall (m, n, p) \in S, rep(t, m, n) = t'\}$$

$$esm(e, t, M) = \min\{es(e, t') * p |\ \forall S \in M, \forall (t', p) \in cor(t, S)\}$$

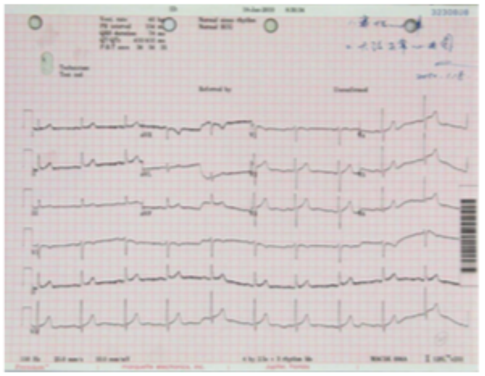
Manual Correction Policy

- Random
- Most frequent error description elements

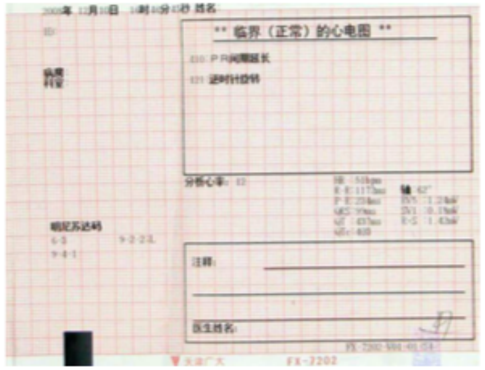
Outline

- Input
- ODL
- Manual Correction
- Experimental Results
  - Dataset
  - Other Approach
  - Extraction Accuracy
  - Incremental Manual Correction

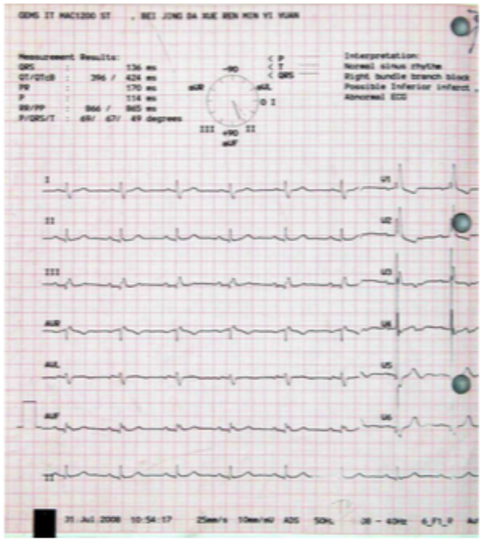
Dataset



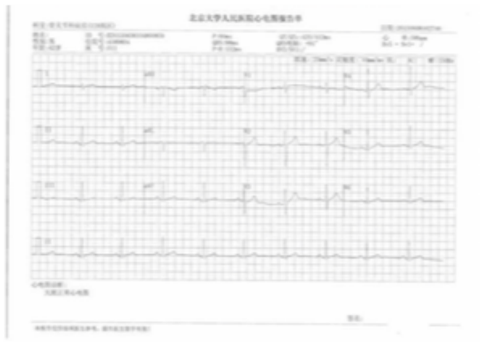
(a) Format 1



(b) Format 2



(c) Format 3

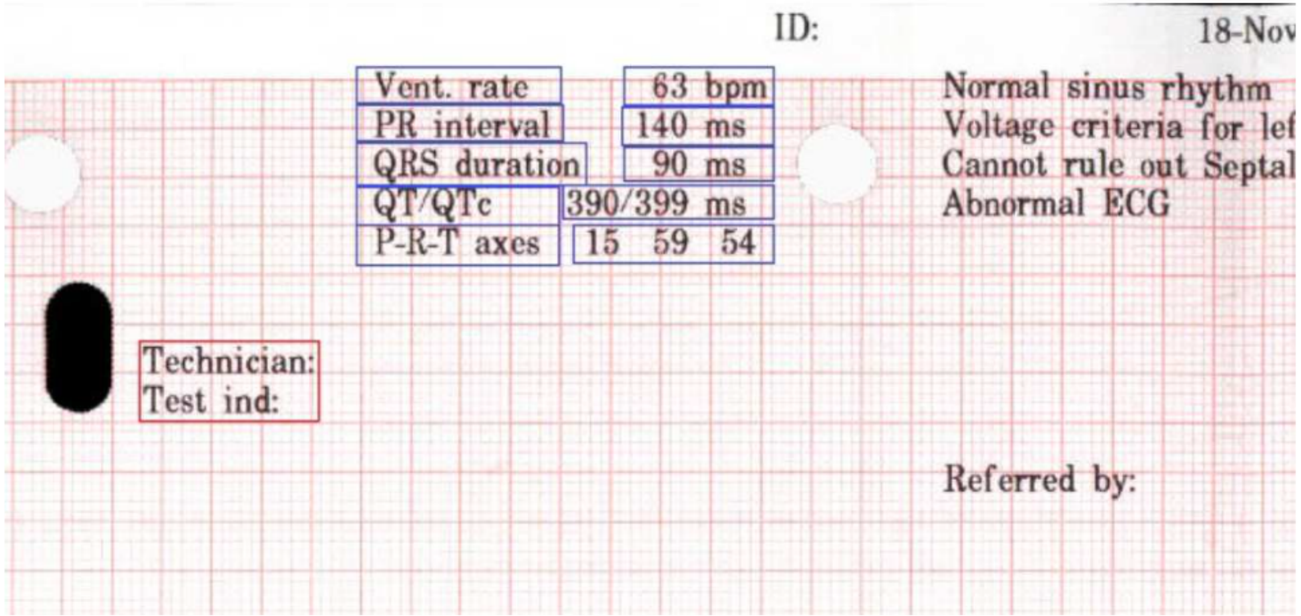


(d) Format 4

Format	1	2	3	4
Number of Images	124	113	102	97
Number of Attributes per Image	17	16	18	15

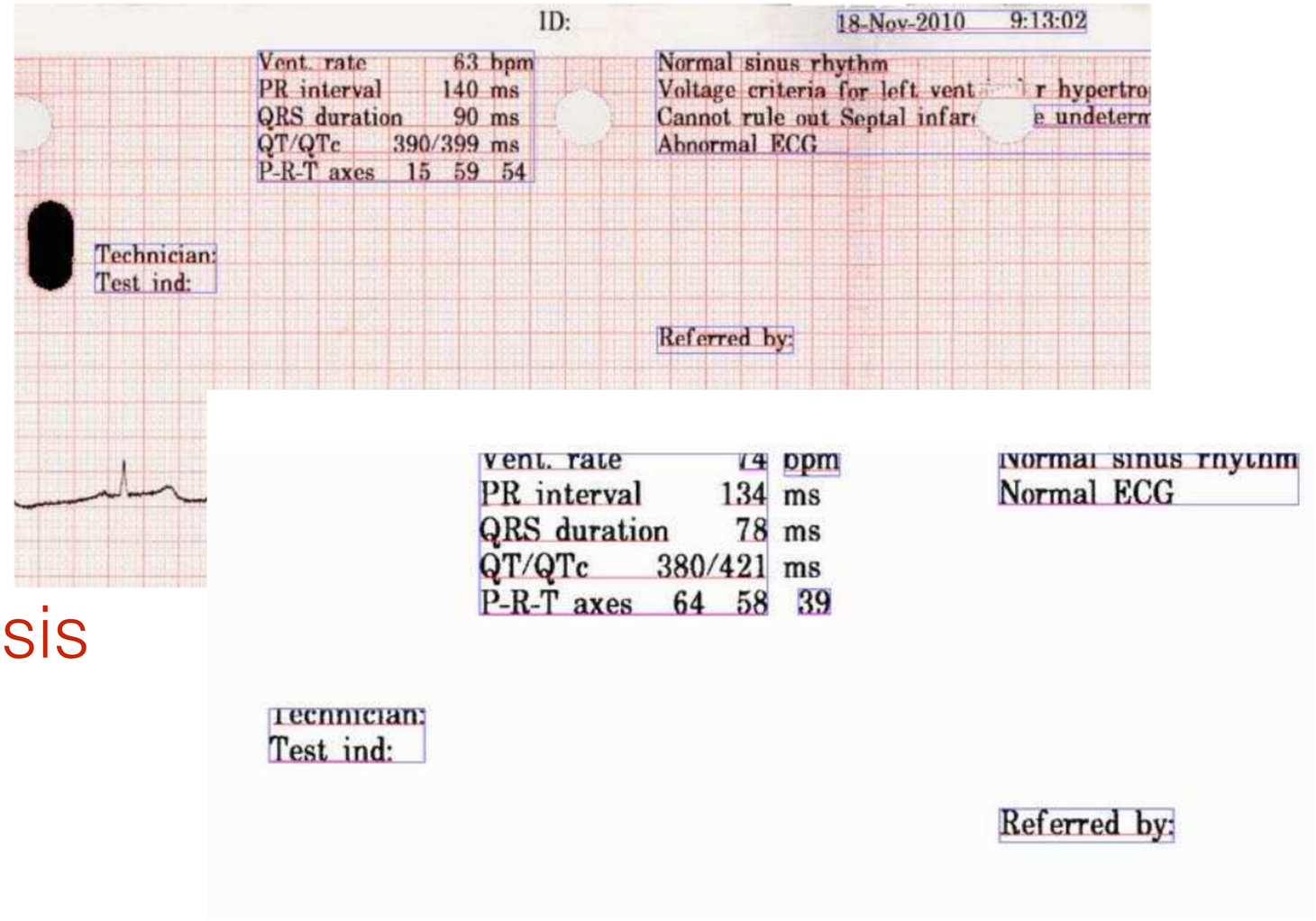
Other Approach

- Exact match
- Zonal OCR
- Page layout analysis



Other Approach

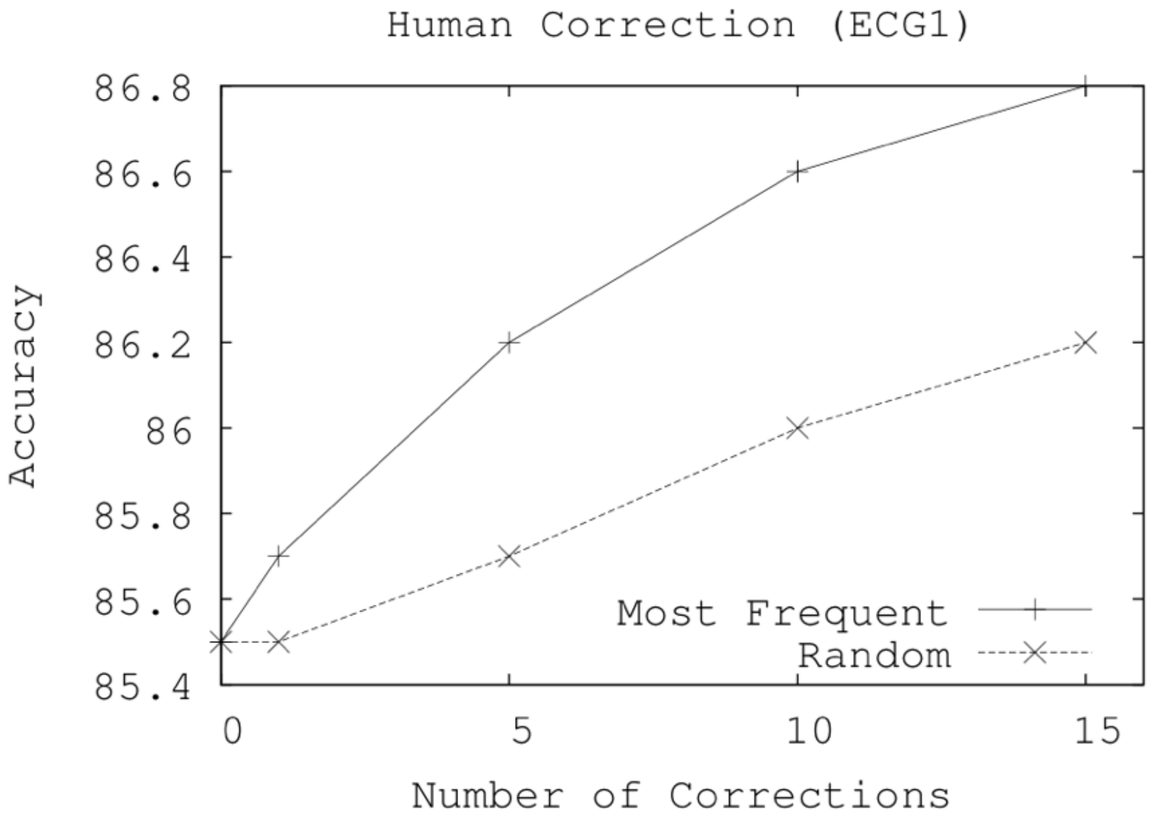
- Exact match
- Zonal OCR
- Page layout analysis

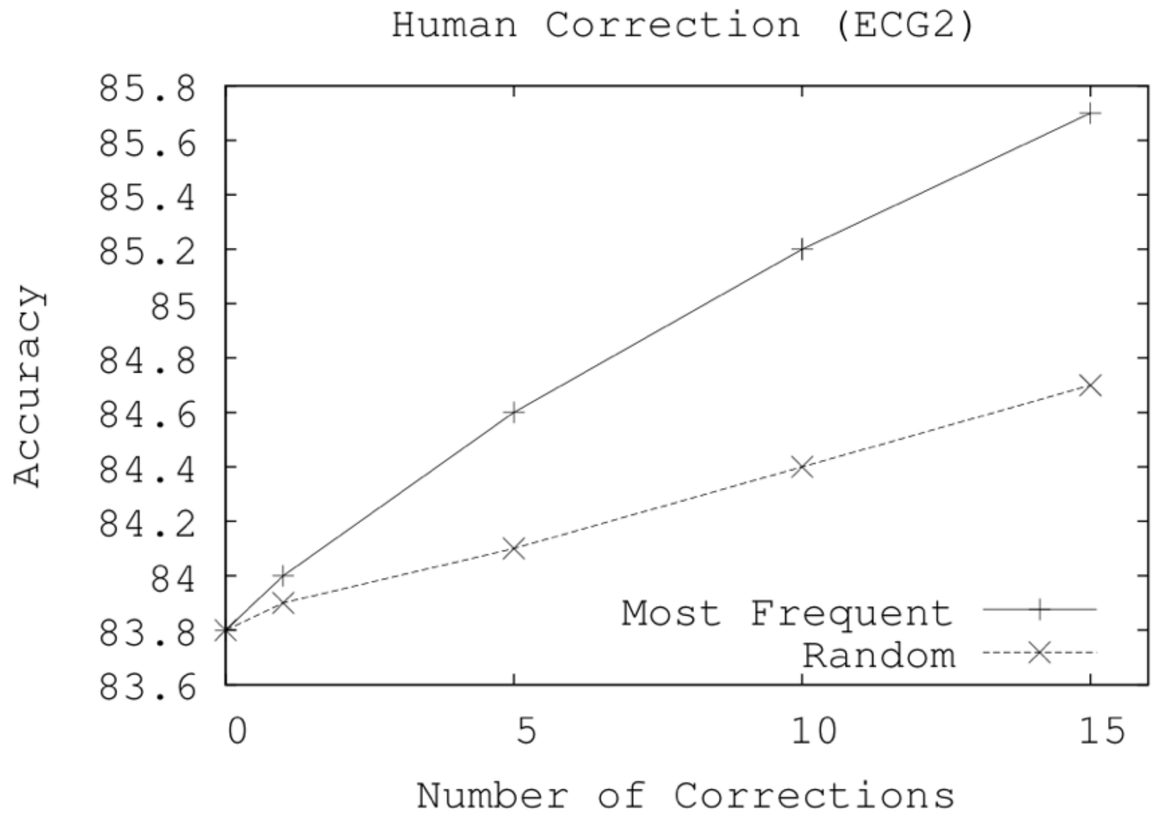


Extraction Accuracy

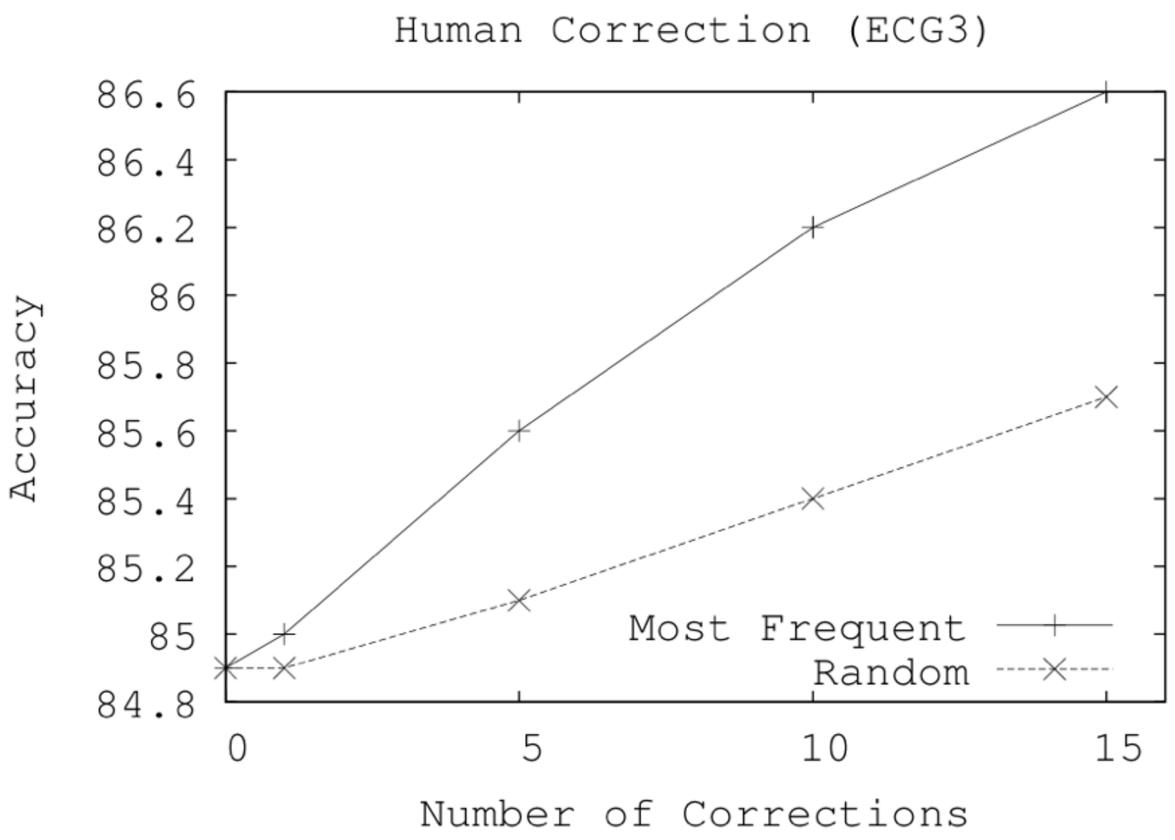
Format	1	2	3	4
Exact Match	58.8%	56.3%	61.1%	53.4%
Zonal OCR	81.2%	79.8%	81.7%	80.6%
Page Layout	79.7%	80.2%	81.2%	81.1%
Our Fuzzy Match	**85.5%**	**83.8%**	**84.9%**	**84.0%**

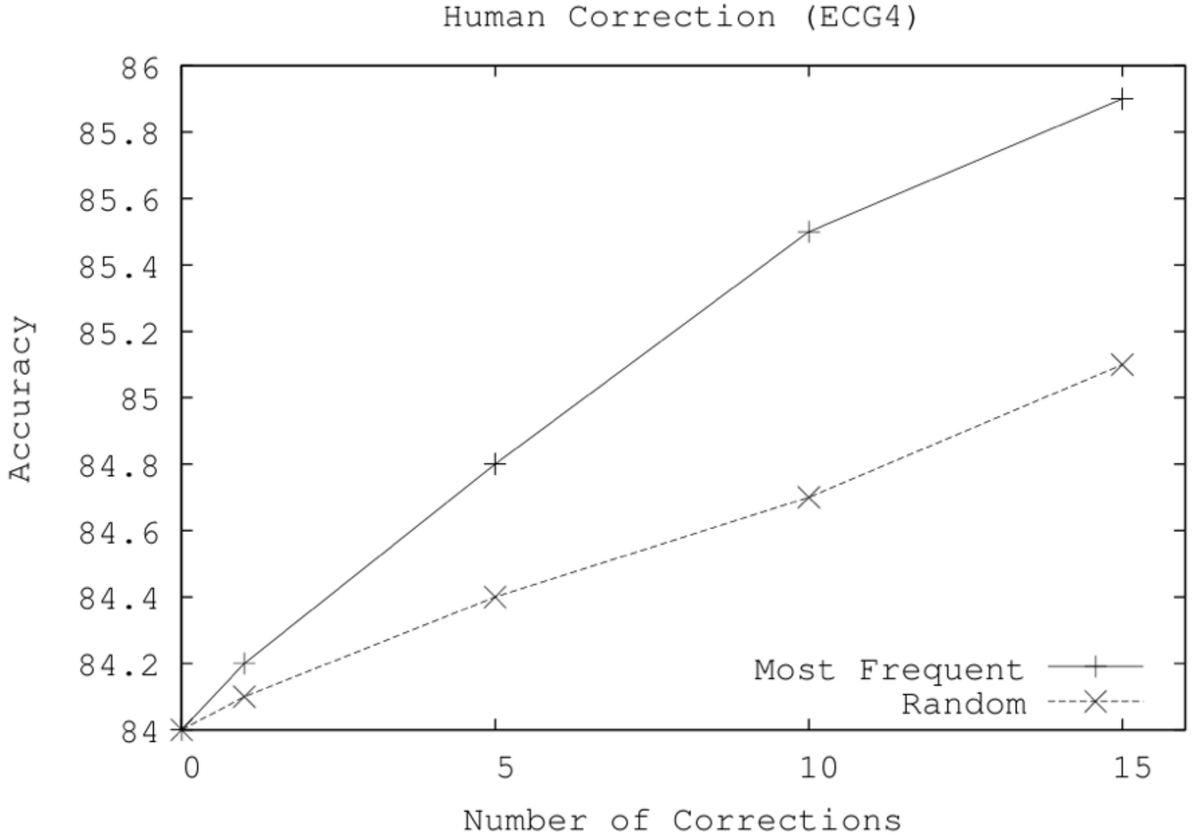
Incremental Manual Correction





Incremental Manual Correction





Thank you!