Abstract—In this paper, we report on two experiments on identification and extraction of Date of Birth instances. The objective of these experiments is to increase the recall level by increasing the edit distance while obtaining a reasonable precision.

Keywords—information extraction; duality; patterns; relations; OCR

I. INTRODUCTION

Federal agencies, state governments, and local agencies are regularly making records available to the public via the Internet or other media. All these records must be scanned for Personally Identifiable Information (PII) prior to public release. An example is the Licensing Support Network (LSN)[1]. The LSN is a large document repository of over forty million pages that will provide information to the proceedings for licensing Yucca Mountain’s nuclear waste repository. A typical example of PII is the occurrences of individuals and their dates of birth. Other examples are individuals’ addresses, phone numbers, fax numbers, email addresses, and credit card information.

Manual review for PII of large collections is not only expensive but also impractical. Knowing that the number of documents to be reviewed for PII is substantial and human review is costly, it is clearly advantageous to seek automatic methods for identifying PII in documents. Since most records are retained in paper form, an automatic solution begins with conversion to an electronic format using optical character recognition (OCR). Subsequently, the process would then apply sophisticated information extraction (IE) and/or categorization techniques to identify records with PII. Finally, these documents must then be redacted prior to public release.

In this paper we describe the development and effectiveness of a fuzzy pattern-based extraction program for identification of Date of Birth PII. We found that fuzzy extraction provides high recall for this task. However, we also discovered that fuzzy matching produces a high recall at the expense of a low precision.

We will first describe the problem our information extraction (IE) program is designed to solve in section 2. Section 3 describes the detailed design of our solution. The collecting process of the clean and OCRRed text is detailed in Section 4. The setup for blind testing this text is given in section 5 and the actual precision and recall numbers in section 6. Finally in section 7, we provide conclusion and the future direction for our research.

II. THE PROBLEM: EXTRACTING DATE OF BIRTH INFORMATION

Many of the PII patterns are of the form E1 R E2, where E1 and E2 are entities and R is the relationship between these entities. In the Date of Birth example, the relationship R establishes that the entity E2 is the date of birth for the entity E1. The study of patterns of this sort can lead to a partial solution to the PII problem. There are many known techniques for extracting pattern of this form. The most popular methods are a combination of Hidden Markov Model (HMM), language pattern processing, and ad-hoc algorithms[2, 3, 4]. Most of the extraction techniques rely on tokens such as “born” or “birth” to establish the relationship. These tokens could have been corrupted during the conversion process using OCR. Consequently, one has to modify the process to overcome some of the difficulties associated with the OCR errors.

Our method of evaluation for finding date of birth will mirror the standard method in most information extraction experiments. The abbreviation DOB stands for “date of birth”. We use:

\[
\text{recall} = \frac{\text{hits}}{\text{collection hits}}
\]

\[
\text{precision} = \frac{\text{hits}}{\text{reported hits}}
\]

where hits is the number of correct DOB hits, collection hits is the total number of DOBs in the document collection, and reported hits is the total number of DOBs reported by the program.

Hits were measured on a line-by-line basis. We choose our document collection so it would contain a high concentration of DOB. In practice, when filtering DOB information it will be much more sparse. There will be fewer DOB hits per line and per document. All but two of the fifty, or 96%, test documents contained DOB information. The density of DOB information for LSN documents will typically be less than 10% of the documents[2].

III. FUZZY MATCHING AND HIGH RECALL

The objective of this experiment was to achieve a high recall of over 95% with a reasonable precision of around 30%. We constructed two fuzzy patterns using agrep[5, 6] to
identify dates and birth indicators. Agrep stands for approximate regular expression parser. The first pattern is a dash with a number on the left and right, each within 40 characters of the dash. The intention is to capture patterns of the following forms, where the first three are from the clean documents and the second three are from the OCR documents:

- 1870 -1924
- (c. 581 - November, 644)
- (ca. 570/571 Mecca[مكة] – June 8, 632)
- t3lt4ltsl @a. 5701571 Mecca[سـ. IIIi&'] - June 8,632
- Lwnuis Pasteur (1 822- 1 895)
- (t92zg - 1 953)

The second pattern is a number with the token “birth” or “born” with 100 characters of the number. The tokens “birth” or “born” can be as much as an edit distance of 2 away from “birth” or “born”. So “born” is an edit distance of 2 away from born. The intention is to capture patterns of the form:

- born: Jan 01, 1751
- he was born -on Feb. 11, 1947
- Born: August 16, 1769

- Mendel was born in Hyncice, Moravia on 22 July
- tsirth: c1400 in Mainz, Germany
- b, 18 November 1787; d. 10 JulY 1851

Below we outline the two patterns to capture these two instances of birthdates.

**Pattern 1** is used on both the clean and OCR documents.

| [0-9]+.|{0,40} [-].{0,40}[0-9]+ |

Date ranges surrounded with parenthesis are a subset of the date ranges without parenthesis. For this reason, we dropped the parenthesis from this pattern to cover both cases. Both cases occur about equally often for birth dates. The precision is higher when the parentheses are present. Because our main priority for this study is high recall, this is an acceptable simplifying assumption.

Pattern 1 is used on both the clean and OCR documents. For pattern 2 a, the date and keyword can occur in either order. For pattern 2 b, the date must immediately follow the “b.”.

**Pattern 2**

- clean documents:
  - Pattern 2 a: [0-9]+ AND \b(birth|born)b
  - Pattern 2 b: \b[.] [0-9]+

- OCR documents:
  - Pattern 2 a: [0-9]+ AND \b(birth|born){-1+2#2~2}b
  - Pattern 2 b: \b[.] [0-9]++

The patterns are specified using POSIX compliant regular expressions. A string of digits is specified as [0-9]+. A comma or period are specified in ‘.‘. These two characters were found to OCR confused for one another in the training data. The “b” denotes a word boundary. All matching was set to be case-insensitive.

Appending approx-settings to the subpattern sets the approximate matching settings for a subpattern. Limits for the number of errors are specified. The count-limits were used to set limits for the number of insertions (+), deletions (−), substitutions (#), and total number of errors (−). For the approx-setting {-1+2#2~2}, the matching allows at most one deletion, and at most two deletions and at most two substitutions. The total number of errors allowed is two. When speaking more broadly, an edit distance of 2 is a rough approximation of the settings used here.

Consider the misrecognized ‘bom’ for ‘born’. Deleting the ‘r’ and replacing the ‘n’ with ‘m’ can transform one word into the other. Because there were two errors edit operation, the edit distance is 2, which is within the limit of 2 for this approx-setting. So ‘(born){-1+2#2~2}’ will match ‘bom’.

**IV. COLLECTING DOCUMENTS WITH DOB INFORMATION**

For this study, documents on the web about people were needed. A popular list of the one hundred most influential people in history is detailed in Hart[8]. Each person’s name was googled and the first hit was considered as a candidate document. If the URL had a domain name previously seen then that URL was disqualified and the next document in the search query results was considered. This allowed a more random sampling of sites from the web. If we didn’t do this disqualification process, websites like Wikipedia would account for many of the documents and they all have too similar a writing style and formatting.

The first page of each document was printed on a laser printer and scanned using a scanner, model CanoScan LiDE 200 from Cannon. ScanSoft OmniPage SE OCR, provided with the scanner, was used for converting the scanned images into text files. The default scan resolution of 300 dots per inch with grey scale was used for the OCR software. These setting, software, and hardware generated a much higher error rate than state-of-the-art OCR systems.
Regardless the errors are representative of many OCR collections because the initial image quality is sometimes very poor and leads to high error rate even with the state-of-the-art OCR software with optimal settings. At the end of this process, one hundred single page OCRed documents were created as a text file.

The generation of the clean text documents was a little less straightforward. The files were saved as text using the web browser Firefox’s save as text feature. Some files were in UTF-8, ASCII, and ISO-8859 file formats. Some files had illegal byte sequences, which were manually removed. In some cases where the save text failed, the text was saved using cut and paste from the browser into a text editor. Not all of the HTML was removed from the text using Firefox. In particular, there are vestiges left over from hyperlinks and embedded pictures. It is pretty clear when date of birth matches happened on one of these lines, which happens less than five percent of the time. These matches were not considered matches during the statistics collection of precision and recall.

The line terminators were a mix of Window, Unix, and Mac formats. These were all normalized. This made viewing and verification efforts easier. Files were then manually trimmed to the first page, and parts of columns not visible on the first page removed. In some files, the HTML had no line breaks for the text and line breaks were added. The lines breaks were not matched up with the OCR text line-for-line, as this would take much manual data entry work.

All date of birth matches in both the OCR and clean text were assured to occur on a single line. This was done by manually editing of about five percent of the matches. In a more commercial implementation as done in Taghva[2], windows of text centered about key phrases would be used to isolate potential matches rather than examining the text line at a time. Brin[3] used a similar simplification of ignoring multi-line matches in his relation extraction study. No matches occurred across three or more lines in the original text.

Because a dash was used a key matching character, all instances of dashes were normalized to ASCII short dash. Even within the same document before normalization, dashes were encoded with different Unicode sequences yet they might look nearly identical when visually displayed.

V. BLIND TESTING ON BOTH CLEAN AND OCR RED TEXT

The DOB extractor was developed in two successive versions. For the first version, we built an extractor using the first 50 clean text documents. Because these web pages were carefully sampled from the entire Web, they represent a broad range of examples of all the possible text patterns for expressing DOB. The focus here was making sure we did not miss any birth dates, in other words no false negatives. For the second version, we added approximate matching limits for keyword based extraction patterns. The focus here was relaxing the constraints of the extraction patterns without introducing so many false positives that the precision drops below 30%. Again only the first 50 documents were used for developing these extracting patterns and arriving at approximate matching limits.

Once the different extraction patterns were developed, they were not allowed to be modified during blind testing. The testing was done the unseen last 50 clean documents and then last 50 OCRed documents. All the relevancy judgments were made on a line-by-line basis. There were roughly 10K lines of clean text and 3K lines of OCR text across the 50 unseen documents. This difference in number of lines is due to the bloating the text when using Firefox’s save as text feature. Whitespace resembles the HTML source more than it resembles the whitespace of the HTML rendered.

VI. PRECISION AND RECALL RESULTS

<table>
<thead>
<tr>
<th>Pattern 1</th>
<th>Pattern 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>tp: 37</td>
<td>tp: 41</td>
</tr>
<tr>
<td>fn: 0</td>
<td>fn: 0</td>
</tr>
<tr>
<td>precision: 69.81%</td>
<td>recall: 100.00%</td>
</tr>
</tbody>
</table>

TABLE I. PRECISION AND RECALL RESULTS ON 50 UNSEEN CLEAN TEXT DOCUMENTS

<table>
<thead>
<tr>
<th>Pattern 1</th>
<th>Pattern 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>tp: 35</td>
<td>tp: 36</td>
</tr>
<tr>
<td>fn: 0</td>
<td>fn: 3</td>
</tr>
<tr>
<td>precision: 54.69%</td>
<td>recall: 100.00%</td>
</tr>
</tbody>
</table>

TABLE II. PRECISION AND RECALL RESULTS ON 50 UNSEEN OCR TEXT DOCUMENTS

For pattern 1, the recall remains 100% in the presence of OCR errors. The precision dropped 15.12%. The dash in the date range is never misrecognized in the training or unseen documents. For this pattern, there was never a date that was so poorly recognized that there was not even a single digit. A string of digits makes a good proxy for a date when combined with just a few more details for filtering out many of the false positives. We used the same exact pattern on the clean text and the OCR text. The precision is below 70% even on clean text.

For pattern 2, the recall drops 7.69% in an OCR context. The precision dropped 72.79%. On the clean text we use exact matching of keywords. On the OCR text, we use approximate string matching. We don’t exactly predict the OCR errors; we just know that most of the OCR corrupt words will fall near the clean version of the word. We measure distance between OCRed and clean words using the Levenshtein distance, also known at the edit distance.

There are three false negatives in three separate documents that prevented a 100% recall on pattern 2 in the OCR text documents. The first one is attributed to a very small HTML font that caused the space character to be unrecognized.
Clean Text: Jean-Jacques Rousseau was born to Isaac Rousseau and Suzanne Bernard in Geneva on June 28, 1712

OCR Text: JacquesRousseauorsbornrotolisaacRousseauandSuzanneBernardinGenevaonune23, 1712.

The second false negative was due to an unrecognized date because not a single digit was recognized correctly.

Clean Text: Vladimir Lenin was born Vladimir Ilich Ulyanov on April 10, 1870

OCR Text: Vladimir Lenin was born Vladimir Ilich Ulyanov on April ro, rBTo

For both the two errors above, they could have just as easily have effected pattern 1 because both patterns 1 and 2 use the same date recognition strategy. The last error occurred because the pattern in “b. at Tauresium in Illrium, May 11, 483;” where the date doesn’t not immediately follow the “b.” was not seen in the training data. A larger training data set would have captured this extraction pattern.

Because this is a relatively small document collection, any changes in precision or recall greater than 5% must be considered significant. Only recall for the pattern 1 remained unchanged with the introduction of OCR errors. There is an inverse relationship between recall and precision. A high recall is being maintained at the cost of a lower precision.

VII. CONCLUSIONS

Again, the objective of this experiment was to achieve a high recall of over 95% with a reasonable precision of around 30%. For pattern 1, this was achieved with 54.69% precision and 100.00% recall. For pattern 2, this was not achieved with a 24.83% precision and 92.31% recall. We were within 5.17% of this goal for precision and 2.69% for recall. With some more experimentation with the approx-settings, there is some possibility of improving the precision of pattern 2. The 2 out of the 3 recall errors, or false negatives, could just as easily have occurred during the testing of pattern 1. These errors were due to a very noisy OCR results where dates are unrecognizable when relying on digit characters. This need was not foreseen in the training document set. The date extraction pattern could be improved by expanding the date pattern to include dates without numbers but with the names of months.

The degree of OCR corruption in the unseen OCR documents was extreme. For example, in the date “April ro, rBTo” there are 6 errors in the span of just 14 characters. In a state-of-the-art OCR document repository, such as the LSN, the density of errors per token would be less than half that error rate. In these situations, fuzzy information extraction will meet up with fewer unrecoverable errors, and prove more robust than shown here.

Using precise shallow parsing, Pereda and Taghva reported 70.26% recall on a different OCRed collection[7]. The precision in that study was 87.04%. In this study, we successfully prioritized recall at the expense of a significant drop in precision, which we imagine is acceptable in some scenarios.

In future studies, we will develop a systematic approach to setting the edit distance error counts for a given pattern. Navarro outlines many approximate matching techniques suitable for text process[9]. The length of the patterns, the make up of the words within the document set, and the degree of OCR corruption within the collection will play a critical role in determining the optimal error counts. Characterizing the attainable levels of precision and recall will guide IE in an OCR context projects.

ACKNOWLEDGMENT

R.E.P. thanks Otis Barbera and Dino Brule for helping during the proof reading of the document collections. Verifying each hit and miss in both the clean and OCR collection was very tedious.

REFERENCES