Performing Information Extraction to Improve OCR Error Detection in Semi-structured Historical Documents

Thomas L. Packer
Department of Computer Science
Brigham Young University
Provo, Utah, USA
tpacker@byu.net

ABSTRACT

Optical character recognition (OCR) produces transcriptions of document images. These transcriptions often contain incorrectly recognized characters which we must avoid or correct downstream. An ability to both identify OCR errors and extract information from OCR output would allow us to extract and index only correct information and to post-process specific parts of the OCR output with targeted resources (e.g. re-OCR using specialized dictionaries). We present a general approach to OCR error detection that uses a hidden Markov model trained to simultaneously detect OCR errors and extract information. We evaluate this approach in two information extraction settings and on semi-structured text from two machine-printed family history documents. We show this joint approach to OCR error detection to be an improvement over two alternative approaches, one based on dictionary matching and the other using a hidden Markov model trained only to detect OCR errors. In particular, we report an average of 8% increase in macro-averaged F-measure between the dictionary approach and our best HMM. Our contribution is to show how an OCR error detection approach based on a word model can be improved by joining this task with an information extraction task, and that an improvement in OCR error detection is achieved regardless of the information extraction task.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language parsing and understanding*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Experimentation, Performance

Keywords

information extraction, optical character recognition, OCR, semi-structured text, error detection, hidden Markov model

HIP '11, September 16 - September 17 2011, Beijing, China
Copyright 2011 ACM 978-1-4503-0916-5/11/09 ...$10.00.

1. INTRODUCTION

From scanned images of historical documents to document images taken by modern smart phone applications, there is much information found in images of text that would be more useful if extracted accurately. A typical approach to extracting information from document images is to first run a third-party OCR engine. Valuable information can then be extracted from OCR text if we either *omit* or *correct* the errors produced by the OCR engine. In either case, we avoid the situation of consuming inaccurate data—a situation that sometimes may be worse than extracting no information at all (e.g. in knowledge engineering to construct dictionaries, gazetteers and name authorities). The processes of omitting and correcting errors both rely on the important step of error detection. However OCR error detection has rarely been looked at closely outside of one process: automatic OCR error correction. We argue that there is value in doing OCR error detection separately from error correction for the sake of modularity, flexibility, and targeted analysis.

With this in mind, we address the challenge of OCR error detection in the context of extracting information from OCR text. The joint task of OCR error detection and information extraction (IE) is itself a valuable combination for the following reasons. If we wish to manually correct the OCR errors, classifying sections of text using IE will allow us to be more discriminating in what parts of the text to devote our resources to. For example, if we wish to index text for a search engine for finding named entities in historical documents, we may not need to correct any OCR output except for the names to be indexed. Alternately, if we wish to correct OCR errors automatically or to re-OCR the text, we can do so more accurately using dictionaries and other resources customized for the semantic categories assigned to the extracted text.

There is previous research connecting OCR with information extraction, including [16] and [11] who demonstrate that the quality of information extraction is reduced in the presence of OCR errors. Work involving the extraction of named entities from OCR output include [12, 8].

Our joint learning of OCR error detection and information extraction is an instance of multi-task learning [4, 3]. We are not aware of any existing research taking a multi-task learning approach to OCR error detection.

Traditionally, there are two main ways to detect word errors in OCR text: dictionary lookup and character n-gram matching [10, 9]. Most recent approaches to OCR error detection are part of an error correction process. Chen et al. [6] describe a method for identifying erroneous words that relies on the OCR engine's estimation of low confidence characters and lack of a dictionary match, followed by other steps that are dependent on their generating and testing replacements words from partial matches in a lexicon. They report a verification (error detection) accuracy of 92.41%. Pal, Chaudhuri, and Kundu [14, 5] detect OCR errors in a highly inflectional Indian language, Bangla, in two steps. Words containing characters that could not be recognized by the OCR engine are immediately considered erroneous. They then actively detect other errors by matching word roots and suffixes to entries in corresponding dictionaries and then checking for grammatical agreement between the matched roots and suffixes. They do not report error detection accuracies.

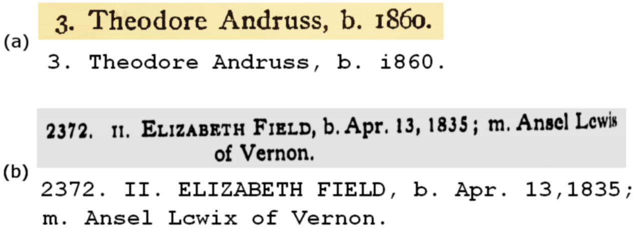
In the present research we combine the task of detecting word-level OCR errors with the task of extracting information from historical documents. We show that this joint approach identifies OCR errors well in text taken from two family history books (achieving 96% overall accuracy in error detection, averaged over four experiments), that it is an improvement over two related approaches, and that improvement occurs for two different IE tasks, namely field segmentation and named entity recognition.

The rest of this paper is organized as follows. In Section 2, we describe our corpus of historical text and how we used it in our experiments. In Section 3, we present the methods we used to detect OCR errors, including a dictionary matching approach and three variations of a hidden Markov model. In Section 4, we report and compare the results of each method applied to our corpus. Finally, in Section 5, we draw conclusions and list future work.

2. DATA AND EXPERIMENTAL SETUP

Our corpus[1] consists of entries from the lists of children in family descriptions. We have taken these list entries (or child records) from two family history books: *The Ely Ancestry* published in 1902 [1], and *The Barber Genealogy* published in 1908 [17]. Almost all of these child records contain the child's birth order, name, and birth date; most contain additional information. Figure 1 shows an example page from each book and Figure 2 shows images and OCR output for one record from each book.

We manually extracted and labeled 300 records from each book. Then we divided each of these two sets of records evenly into training, development test, and blind test sets, for a total of six sets of 100 records each. Table 1 gives other statistics for these six sets of records. We selected the records from consecutive pages from the middle of each books. We did not omit any child records from this contiguous sequence of pages, however we did manually remove text outside of child records and line breaks from any record that spanned multiple text lines. When evaluating our approaches to OCR error detection, we performed parameter estimation using the training sets, we selected variations of each approach (e.g. hyper-parameters) based on their relative performance on the development test sets, and we computed final evaluation metrics for the selected approaches using the blind test sets. We ran each approach once on each blind test set after choosing hyper-parameters for each book. For example, one hyper-parameter for the HMMs was the decision to use the state transition model during Viterbi decoding or not. We report results using a blind test set to reduce the chance that our reported accuracies are dependent on over-tuning our system to a particular test set during development and hyper-parameter selection.


(a)

3. Theodore Andruss, b. 1860.

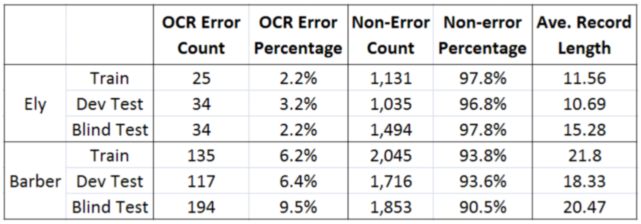
```
3. Theodore Andruss, b. i860.
```

(b)

2372. II. ELIZABETH FIELD, b. Apr. 13, 1835; m. Ansel Lcwis of Vernon.

```
2372. II. ELIZABETH FIELD, b. Apr. 13,1835;
m. Ansel Lcwix of Vernon.
```

Figure 2: Example records with OCR errors taken from Ely (a) and Barber (b), including corresponding image and OCR output.



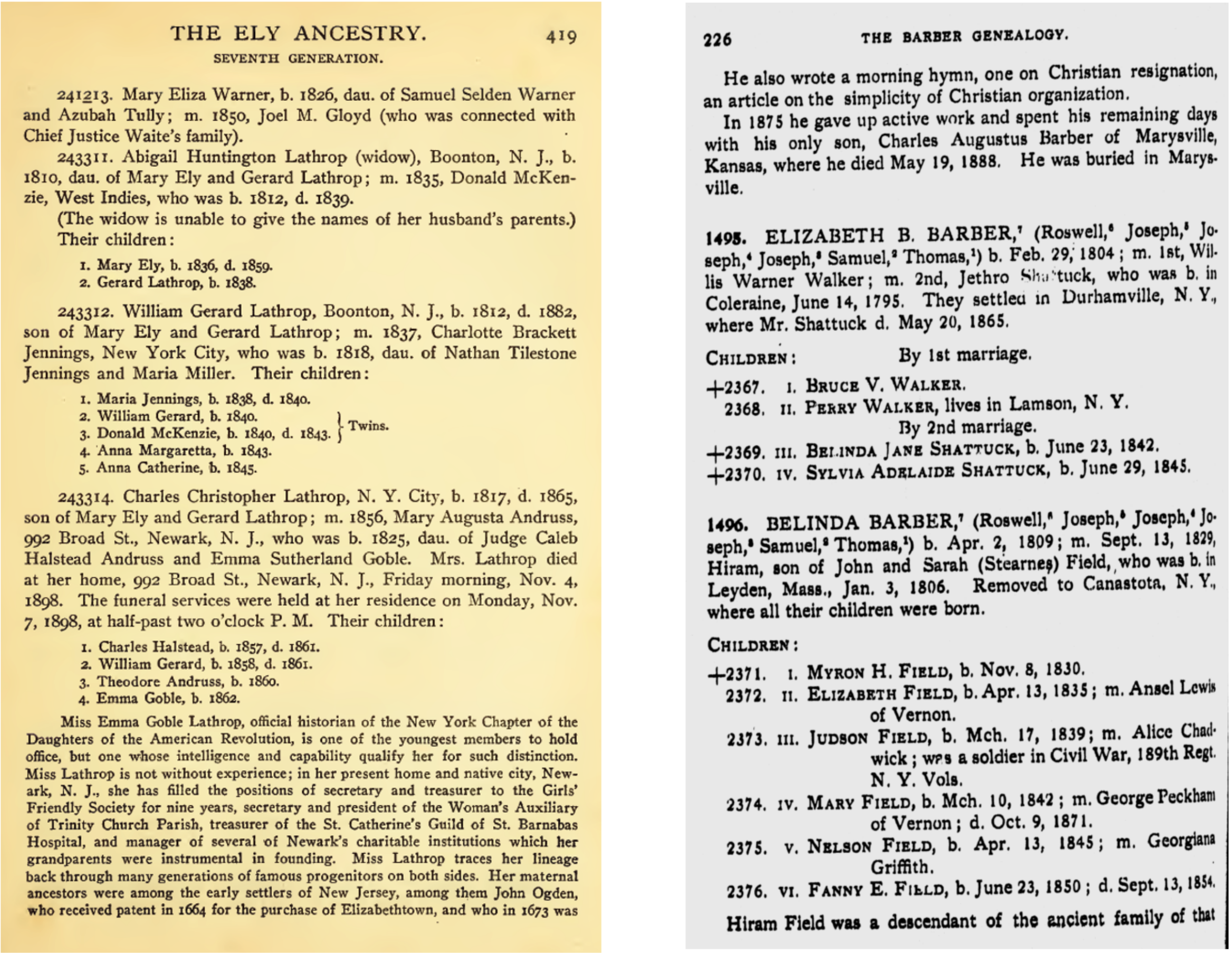
		OCR Error Count	OCR Error Percentage	Non-Error Count	Non-error Percentage	Ave. Record Length
Ely	Train	25	2.2%	1,131	97.8%	11.56
	Dev Test	34	3.2%	1,035	96.8%	10.69
	Blind Test	34	2.2%	1,494	97.8%	15.28
Barber	Train	135	6.2%	2,045	93.8%	21.8
	Dev Test	117	6.4%	1,716	93.6%	18.33
	Blind Test	194	9.5%	1,853	90.5%	20.47

Table 1: Statistics computed over the manually-labeled word tokens in the training, development test, and blind test sets of each book. Each of the six rows above represents 100 records.

We labeled the three sets of records for each book using the following three labeling schemes: *error*, *field-error*, and *entity-error*. We trained each approach on one of these labeling schemes. Figure 3 and Figure 4 illustrate the three labeling schemes as an XML file format and as an aligned diagram, respectively. In *error* labeling, we marked each word token with a boolean flag (e or n) indicating whether it contains an OCR error or not. During both development testing and final testing, we computed evaluation metrics by comparing the automatically predicted *error* labels of each word token against the manually labeled word tokens for each of the four *error*-labeled test sets. The approaches trained on the other two labeling schemes produce compound labels. We therefore had to separate the *error* portion of these labels from the *semantic* portion during evaluation. We represent the information extraction tasks using the semantic portion of these other two labeling schemes, described next.

In *field-error* labeling, we marked each word token with a pair containing both the error flag (described above) and a

[1] All the data used in this paper is available by email from the author and from our wiki `https://facwiki.cs.byu.edu/Ancestrycorp/index.php/Main_Page` under the heading "Family History Children Lists".



THE ELY ANCESTRY. 419

SEVENTH GENERATION.

241213. Mary Eliza Warner, b. 1826, dau. of Samuel Selden Warner and Azubah Tully; m. 1850, Joel M. Gloyd (who was connected with Chief Justice Waite's family).

243311. Abigail Huntington Lathrop (widow), Boonton, N. J., b. 1810, dau. of Mary Ely and Gerard Lathrop; m. 1835, Donald McKenzie, West Indies, who was b. 1812, d. 1839.

(The widow is unable to give the names of her husband's parents.)

Their children:

1. Mary Ely, b. 1836, d. 1859.
2. Gerard Lathrop, b. 1838.

243312. William Gerard Lathrop, Boonton, N. J., b. 1812, d. 1882, son of Mary Ely and Gerard Lathrop; m. 1837, Charlotte Brackett Jennings, New York City, who was b. 1818, dau. of Nathan Tilestone Jennings and Maria Miller. Their children:

1. Maria Jennings, b. 1838, d. 1840.
2. William Gerard, b. 1840. } Twins.
3. Donald McKenzie, b. 1840, d. 1843. }
4. Anna Margaretta, b. 1843.
5. Anna Catherine, b. 1845.

243314. Charles Christopher Lathrop, N. Y. City, b. 1817, d. 1865, son of Mary Ely and Gerard Lathrop; m. 1856, Mary Augusta Andruss, 992 Broad St., Newark, N. J., who was b. 1825, dau. of Judge Caleb Halstead Andruss and Emma Sutherland Goble. Mrs. Lathrop died at her home, 992 Broad St., Newark, N. J., Friday morning, Nov. 4, 1898. The funeral services were held at her residence on Monday, Nov. 7, 1898, at half-past two o'clock P. M. Their children:

1. Charles Halstead, b. 1857, d. 1861.
2. William Gerard, b. 1858, d. 1861.
3. Theodore Andruss, b. 1860.
4. Emma Goble, b. 1862.

Miss Emma Goble Lathrop, official historian of the New York Chapter of the Daughters of the American Revolution, is one of the youngest members to hold office, but one whose intelligence and capability qualify her for such distinction. Miss Lathrop is not without experience; in her present home and native city, Newark, N. J., she has filled the positions of secretary and treasurer to the Girls' Friendly Society for nine years, secretary and president of the Woman's Auxiliary of Trinity Church Parish, treasurer of the St. Catherine's Guild of St. Barnabas Hospital, and manager of several of Newark's charitable institutions which her grandparents were instrumental in founding. Miss Lathrop traces her lineage back through many generations of famous progenitors on both sides. Her maternal ancestors were among the early settlers of New Jersey, among them John Ogden, who received patent in 1664 for the purchase of Elizabethtown, and who in 1673 was

226 THE BARBER GENEALOGY.

He also wrote a morning hymn, one on Christian resignation, an article on the simplicity of Christian organization.

In 1875 he gave up active work and spent his remaining days with his only son, Charles Augustus Barber of Marysville, Kansas, where he died May 19, 1888. He was buried in Marysville.

1495. ELIZABETH B. BARBER,^{7} (Roswell,^{6} Joseph,^{5} Joseph,^{4} Joseph,^{3} Samuel,^{2} Thomas,^{1}) b. Feb. 29, 1804; m. 1st, Willis Warner Walker; m. 2nd, Jethro Shattuck, who was b. in Coleraine, June 14, 1795. They settled in Durhamville, N. Y., where Mr. Shattuck d. May 20, 1865.

CHILDREN: By 1st marriage.

+2367. I. BRUCE V. WALKER.
2368. II. PERRY WALKER, lives in Lamson, N. Y.

By 2nd marriage.

+2369. III. BELINDA JANE SHATTUCK, b. June 23, 1842.
+2370. IV. SYLVIA ADELAIDE SHATTUCK, b. June 29, 1845.

1496. BELINDA BARBER,^{7} (Roswell,^{6} Joseph,^{5} Joseph,^{4} Joseph,^{3} Samuel,^{2} Thomas,^{1}) b. Apr. 2, 1809; m. Sept. 13, 1829, Hiram, son of John and Sarah (Stearnes) Field, who was b. in Leyden, Mass., Jan. 3, 1806. Removed to Canastota, N. Y., where all their children were born.

CHILDREN:

+2371. I. MYRON H. FIELD, b. Nov. 8, 1830.
2372. II. ELIZABETH FIELD, b. Apr. 13, 1835; m. Ansel Lewis of Vernon.
2373. III. JUDSON FIELD, b. Mch. 17, 1839; m. Alice Chadwick; was a soldier in Civil War, 189th Regt. N. Y. Vols.
2374. IV. MARY FIELD, b. Mch. 10, 1842; m. George Peckham of Vernon; d. Oct. 9, 1871.
2375. V. NELSON FIELD, b. Apr. 13, 1845; m. Georgiana Griffith.
2376. VI. FANNY E. FIELD, b. June 23, 1850; d. Sept. 13, 1854.

Hiram Field was a descendant of the ancient family of that

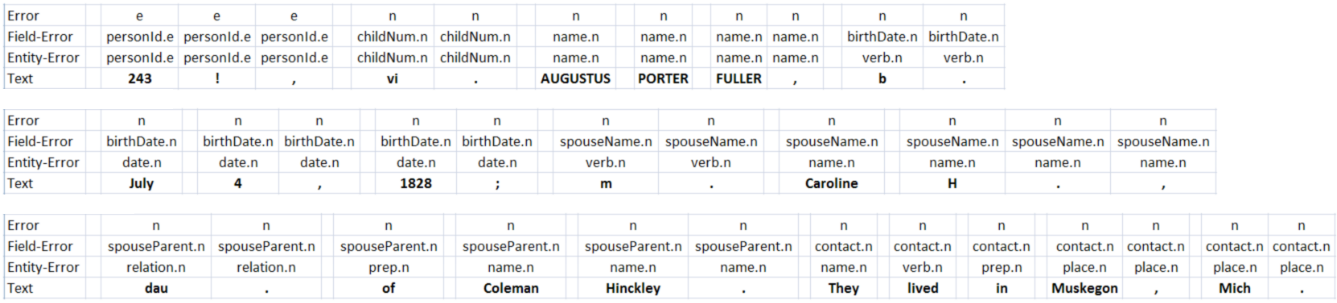
Figure 1: An example page of each of the two family history books from which we took text for our corpus.

```
(a) <e>243!,</e> <n>vi.</n> <n>AUGUSTUS PORTER FULLER,</n> <n>b. July 4, 1828;</n> <n>m. Caroline H.,</n>
    <n>dau. of Coleman Hinckley.</n> <n>They lived in Muskegon, Mich.</n>

    <personId.e>243!,</personId.e> <childNum.n>vi.</childNum.n> <name.n>AUGUSTUS PORTER FULLER,</name.n>
    <birthDate.n>b. July 4, 1828;</birthDate.n> <spouseName.n>m. Caroline H.,</spouseName.n>
(b) <spouseParent.n>dau. of Coleman Hinckley.</spouseParent.n>
    <contact.n>They lived in Muskegon, Mich.</contact.n>

    <personId.e>243!,</personId.e> <childNum.n>vi.</childNum.n> <name.n>AUGUSTUS PORTER FULLER,</name.n>
    <verb.n>b.</verb.n> <date.n>July 4, 1828;</date.n> <verb.n>m.</verb.n> <name.n>Caroline H.,</name.n>
(c) <relation.n>dau.</relation.n> <prep.n>of</prep.n> <name.n>Coleman Hinckley.</name.n>
    <name.n>They</name.n> <verb.n>lived</verb.n> <prep.n>in</prep.n> <place.n>Muskegon, Mich.</place.n>
```

Figure 3: An example record from the Barber portion of our corpus labeled in an XML style with each labeling scheme: "Error" (a), "Field-Error" (b), and "Entity-Error" (c).



Error	e	e	e	n	n	n	n	n	n	n	n
Field-Error	personId.e	personId.e	personId.e	childNum.n	childNum.n	name.n	name.n	name.n	name.n	birthDate.n	birthDate.n
Entity-Error	personId.e	personId.e	personId.e	childNum.n	childNum.n	name.n	name.n	name.n	name.n	verb.n	verb.n
Text	**243**	**!**	**,**	**vi**	**.**	**AUGUSTUS**	**PORTER**	**FULLER**	**,**	**b**	**.**

Error	n	n	n	n	n	n	n	n	n	n	n
Field-Error	birthDate.n	birthDate.n	birthDate.n	birthDate.n	birthDate.n	spouseName.n	spouseName.n	spouseName.n	spouseName.n	spouseName.n	spouseName.n
Entity-Error	date.n	date.n	date.n	date.n	date.n	verb.n	verb.n	name.n	name.n	name.n	name.n
Text	**July**	**4**	**,**	**1828**	**;**	**m**	**.**	**Caroline**	**H**	**.**	**,**

Error	n	n	n	n	n	n	n	n	n	n	n	n	n
Field-Error	spouseParent.n	spouseParent.n	spouseParent.n	spouseParent.n	spouseParent.n	spouseParent.n	contact.n	contact.n	contact.n	contact.n	contact.n	contact.n	contact.n
Entity-Error	relation.n	relation.n	prep.n	name.n	name.n	name.n	name.n	verb.n	prep.n	place.n	place.n	place.n	place.n
Text	**dau**	**.**	**of**	**Coleman**	**Hinckley**	**.**	**They**	**lived**	**in**	**Muskegon**	**,**	**Mich**	**.**

Figure 4: The example record from Figure 3 after tokenization with word tokens and corresponding labels aligned.

Ely Fields (8)	Barber Fields (21)	Ely Entities (6)	Barber Entities (12)
birthDate	birthDate	childNum	bracket
childNum	birthPlace	date	childNum
contact	bracket	name	conj
deathDate	child	other	date
marriageDate	childNum	place	individualEntry
name	contact	verb	name
other	deathDate		other
spouseName	deathPlace		personId
	individualEntry		place
	marriageDate		prep
	marriagePlace		relation
	name		verb
	other		
	personId		
	sibling		
	spouseBirthDate		
	spouseDeathDate		
	spouseFamily		
	spouseName		
	spouseParent		
	spouseResidence		

Table 2: The complete set of labels used in annotating the text in each book for each scheme.

label for the field segment of the child record that the token falls within. For example, both the “b.”, and the year that follows, belong to the *birthDate* field. This labeling scheme is similar to that applied to the Cora bibliography data set which is used in a number of field segmentation papers (e.g. [7]). In this paper, we refer to extracting fields, including the determination of field segment boundaries and labels, as “field segmentation”, consistent with the language used by [2] and [7].

In *entity-error* labeling, we marked each word token with both the error flag and a label for the kind of named entity it belongs to if part of a noun phrase, or its part-of-speech tag if not part of a noun phrase. For example, we labeled a “b.” as a *verb* and the following year as a *date*. This labeling scheme is similar to that used in named entity recognition tasks plus the addition of three parts of speech (verb, preposition, and conjunction) to maintain a uniform labeling granularity throughout each record.

To give the reader a better sense of how our two labeling schemes differ, here are some consequences. A consequence of *entity* labeling is that a name will be labeled as a name regardless of what relationship or field it is part of, whether the name belongs to the person whom the record is about or to that person’s spouse, child, parent, etc. (This is also true for the other entity types.) Therefore in *entity* labeling, it is possible to model names using all the available given names and surnames found in the data, potentially leading to a simpler emission model of the HMM. In other words, there are fewer states in the HMM with potentially less overlap in their vocabulary. With *field* labeling on the other hand, these names would be given different labels depending on the field they are found in. Also, the words nearby within the same field would be given the same label. For example, the verb, “m.”, next to a spouse name would also be labeled as spouse. This could lead to a simpler transition model in the sense that there are more self-transitions (from a state to itself) and fewer transitions between states. This is not necessarily advantageous when using HMMs which are known to not model field length well using self-transitions. (See Section 4 for specific consequences.)

Our approaches label sequences of atomic segments of text—word tokens—which we first separated from each other. In tokenizing the text, we separated numerals from other text and treated punctuation characters as separate tokens. When hand-labeling the data, we gave each punctuation character the same label as the token immediately preceding it in all cases. We did not read in whitespace except in detecting boundaries between tokens. We also did not hand label missing words, e.g. punctuation characters missed by the OCR engine. There were only a few cases of missing word tokens in this corpus. The complete set of field and entity labels for each book appear in Table 2.

There is a common concern with using small training sets, especially in the case of numerals (e.g. years and child order numbers). To mitigate data sparseness due to a limited training set in learning HMM parameters, and to make it less burdensome to specify numeral patterns in the dictionaries, we replaced all digits in our corpus with a single digit (“8”). Doing this improved accuracies by about 2% on the development test sets of both books. Replacing only certain subsets of digits (e.g. all digits except for “0” or all but “1”) further improved accuracies only for the Barber book, but by less than a percent. To maintain consistency and simplicity, we chose to conflate all digits in both books for all experiments reported below.

The goal of our experiments was to determine which of our approaches produced the highest macro-averaged F-measure for the task of OCR error detection on unseen text in the two documents in our corpus. To compute macro-averaged F-measure, we first computed the F-measure for each label (in this case, *error* and *non-error*). Then we computed the standard mean of these two F-measure values. So, macro-averaged F-measure $= \frac{F_1(e)+F_1(n)}{2}$. The F-measure, $F_1(x)$, for each label x is the harmonic mean of precision, $P(x)$, and recall, $R(x)$, for that label. $P(x) = \frac{C(x)}{B(x)}$, $R(x) = \frac{C(x)}{A(x)}$, $F_1(x) = \frac{2PR}{P+R}$. Here, $A(x) =$ the number of words with actual label x, $B(x) =$ the number of words with predicted label x, and $C(x) =$ the number of words with both predicted label x and actual label x.

We chose to use a macro-average because we are most concerned about the precision and recall in predicting the *error* label and because the proportion of words with OCR errors is relatively small in our corpus, as quantified in Table 1. If we had used a micro-average over the two labels, the performance metrics would have been dominated by the more frequent *non-error* label.

3. APPROACHES

3.1 Dictionary

It is common to use lists of real words to detect spelling and OCR errors [10] [14]. OCR engines themselves often use dictionaries to improve their recognition rate. In constructing our dictionary approach to OCR error detection, we assembled lists of words from various sources that matched our language and domain. Our dictionaries include given names (18,000 instances), surnames (150,000 instances), initials (capital letters A through W), person titles (10 instances including "Mr" and "Jr"), days of the week (16 names and abbreviations), months (26 names and abbreviations), and common English words. For the common English word list, we selected one list from the ten produced by Keith Vertanen.[2] Vertanen produced the 10 lists by taking candidate words from 10 source corpora and then filtering the candidates 10 different times, each time using a different vote threshold. He gave each source corpus one vote. Using each of the 10 lists, we tested performance on the *error*-labeled training set and found that wlist_match1 (the complete union of all ten source lists) performed best. This list contains 1.7 million entries including most of the person and place names in our corpus such as "Muskegon", "Kalamazoo", "Mosher", and "Burnham". We checked for matches in all our dictionaries in a case-insensitive manner.

We constructed two other dictionaries whose entries we selected individually for our OCR data: a list of numeral patterns (1-, 2-, and 4-digit numerals) and a list punctuation characters (. , ; + "). We evaluated performance as we added each numeral pattern and each punctuation character found in the training data one at a time. The tokens whose inclusion improved performance on the training data, we kept in the list; the others, we removed. We conducted the selection process for the general English words list, the punctuations, and the numeral patterns independently for each of the two books. This process selected the same lists for both books.

3.2 Hidden Markov Models

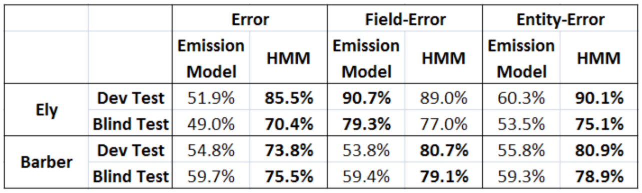
The hidden Markov model [15] or HMM is a statistical model for labeling sequences, e.g. for applying part-of-speech labels to sequences of words. It has traditionally been used in many tasks such as speech recognition [15], information extraction [2], and genomics [18]. HMMs can model sequences of continuous- or discrete-valued observations. Here we apply it to the task of predicting OCR errors from discrete-valued observations (the tokens in the OCR output) with and without joining it to an information extraction task.

We perform two operations on the HMMs: training and execution. During training we used maximum likelihood estimation to compute the parameters (or probability distributions) of the HMM from the manually labeled word tokens in the child records in our two training sets. The learned probability distributions include both the state-state transition parameters and state-word emission parameters, as usual. During training we used a one-to-one correspondence between the states in the HMM and the labels manually assigned to the word tokens in the training data. (These labels are shown in Table 2.) Therefore, each HMM contains one state for each type of label found in its training data—the HMM will be incapable of predicting a label that does not appear in its training set.

During execution, the learned parameters are used in the Viterbi algorithm to find, for each record in the test set, the sequence of states (labels) that produces the maximum joint probability for the sequence of word tokens in that record. Out-of-vocabulary (OOV) words are words in the test set that are not in the training set and therefore have zero probability of being emitted by the trained generative model. To prevent these OOV words from producing probabilities of zero for entire records during execution, we smoothed the emission parameters by assigning a probability of $1/N$ to OOV words. Here N equals the number of tokens in the training set. This smoothing method produced higher accuracies in a sample taken from the development test data than Laplace smoothing. We smooth the transition parameters in the same way, meaning that any state may follow any other state with a probability of $1/N$ if that pattern was not seen in the training data. During development we executed a sample of our HMMs without this transition parameter smoothing and saw no change in accuracy.

Our three HMMs differ from each other in the labeling scheme used to train them and therefore also in what kinds of labels they produce during execution. We trained the error HMM using the error labeling scheme, so it produces a binary classification for each word token in the test sets. In other words, we trained this HMM on the same kind of data we tested all the approaches on. On the other hand, we trained the field-error and entity-error HMMs on the training data with corresponding labels. Because of this, they were trained to produce labels that are the conjunction of an OCR error label and a semantic label as described in

[2] Token from `http://www.keithv.com/software/wlist/`



		Error		Field-Error		Entity-Error	
		Emission Model	HMM	Emission Model	HMM	Emission Model	HMM
Ely	Dev Test	51.9%	**85.5%**	**90.7%**	89.0%	60.3%	**90.1%**
	Blind Test	49.0%	**70.4%**	**79.3%**	77.0%	53.5%	**75.1%**
Barber	Dev Test	54.8%	**73.8%**	53.8%	**80.7%**	55.8%	**80.9%**
	Blind Test	59.7%	**75.5%**	59.4%	**79.1%**	59.3%	**78.9%**

Table 3: Macro-averaged F-measure of OCR error detection for all combinations of books (2), test sets (2), labeling schemes (3), and HMM variations (2). This illustrates a strong though inconsistent dependency of our HMM's accuracy on the learned state transition model.

Section 2. Note that the current implementation of our HMMs treat each compound label as completely distinct from any other label. So, even though it may be possible to improve their performance by treating labels like "name.e" and "name.n" as related or similar, our HMMs don't do that. To evaluate these two HMMs, we compared the error label in the hand-labeled test data with the error portion of the conjoined labels that the HMMs produced. Note that we can similarly use these two HMMs to do information extraction (without OCR error detection) by looking at only the semantic portion of the labels they produce.

The HMM is related to the dictionary approach in that both base their labeling decisions, at least in part, on finding a match for a word token in a list of words associated with some state or label. In other words, they both base their decision on whole word *content*. The HMM carries additional information in the form of the two probability models, one of which (the transition model) models a word token's *context*. It may be interesting to know if both probability models are necessary for this task or what the relative value of context vs. content is. We can easily modify an HMM to ignore its transition (context) model and therefore select the label that maximizes each token's probability given only the emission (content) model during execution. We used this technique below to investigate whether our HMMs' levels of performance were due in part to modeling the relative order of states in the input records (the transition model) or whether that level of accuracy could be accounted for merely by modeling the distribution of words associated with each label (the emission model). We selected one of these two HMM variations for execution on each combination of book and labeling scheme in the blind test set based on its performance on the corresponding development test set.

During development, we trained and executed a sample of the HMMs on case-folded text. This consistently reduced the evaluation metrics by less than a percent. So, unlike the dictionary approach, we applied all our HMMs in a case-sensitive manner.

4. RESULTS

As seen in Table 3, the full HMM performed better than the isolated emission model for all combinations of book and labeling scheme except one, namely the field-error labeling scheme applied to the Ely book. In this case, the transition model is likely causing errors because of a well-known problem with the HMM, namely its inability to correctly model the length of a field [13, 19]. There are more states with self-transitions in our field-labeled data than in our entity-labeled data. Based on their relative performance in the development test sets, we chose to execute the isolated emission model for that combination and the full HMM for the other five in the results reported below. Fortunately, the relative performances were consistent between development and blind test sets.

Figure 5 shows the results of the final versions of our four approaches on the four combinations of test set and book. The two HMM approaches that also perform information extraction consistently outperform the two simpler approaches to error detection. To illustrate why, we draw an example from the emission models of the HMMs learned from the Barber training set. A three-digit numeral pattern ("888") appears exactly once in both the "e" state and the "n" state of the HMM trained on the error-labeled data, making this an ambiguous word without additional semantic information. Therefore, the error HMM's emission model is less discriminating than the emission model trained on field-error labeled data. Given this labeling scheme, the *erroneous* three-digit number appears in the field "<personId.e> 238!, </personId.e>" while the *non-erroneous* three-digit number appears in a different type of field, namely "<other.n> a soldier in Civil War, 189th Regt. N. Y. Vols. </other.n>".

We next show that both the field-error and entity-error HMMs consistently perform better than the error HMM regardless of how many records are in the training data beyond just a few records. Figure 6 shows the macro-averaged F-measure computed for both blind test sets when training each of the three HMMs on varying numbers of training examples. The Y-value for each point is the average of 50 runs of training using a different subset of the training set per run, followed by execution on the whole blind test set.

Lastly we note that the *micro-averaged* F-measure (and equivalently, overall accuracy) for the multi-task HMMs evaluated on the error detection task using the blind test set fell between 94% and 98%. Accuracy is computed as $C/(C+I)$, where C is the number of correctly labeled tokens and I is the number of incorrectly labeled tokens in the test set.

5. CONCLUSIONS AND FUTURE WORK

The relative performance of the four methods is fairly consistent across books and between development and final testing. Most notably, the two forms of joint IE and error detection consistently outperform the other two methods, with a 5.1% increase in macro-averaged F-measure in blind testing averaged over the four versions of HMM (two labeling schemes and two books) and an 8.1% average increase between the dictionary approach and the best HMM. This supports the idea that applying mutli-task learning to OCR error detection and information extraction is beneficial.

We originally hoped to show that the field-error and entity-error HMMs also perform better at the information extraction tasks compared to an HMM trained on field or entity labels alone. However, the experiments we have performed so far show only inconsistent improvement in the informa-

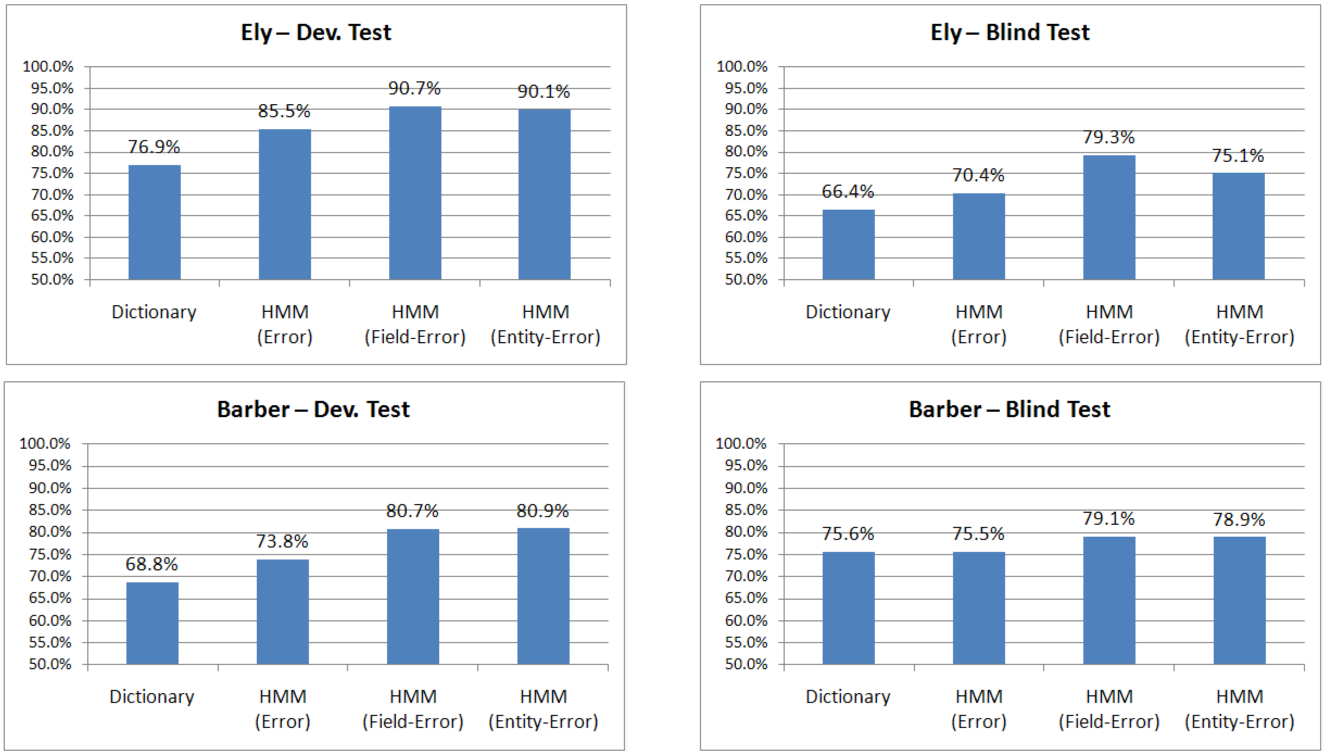
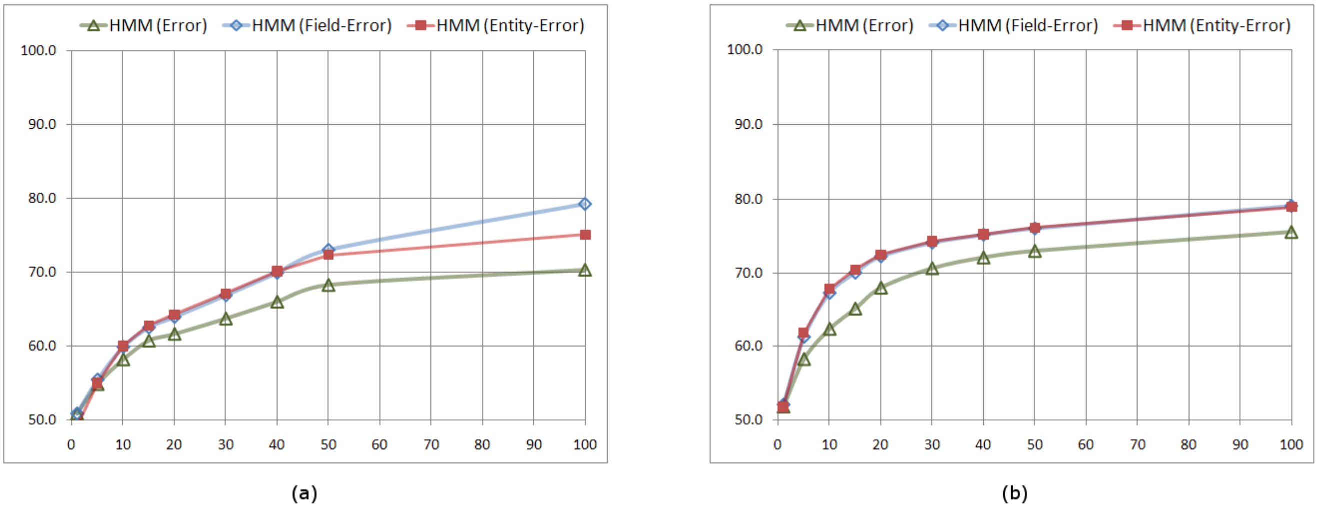


Figure 5: Macro-averaged F-measure of four approaches to OCR error detection computed for four combinations of test set and family history book.



(a)

(b)

Figure 6: Macro-averaged F-measure of OCR error detection on the blind test set of the Ely (a) and Barber (b) family history books for various numbers of training records (X-axis).

tion extraction task, with improvements of macro-averaged F-measure falling between 0% (due to identical label predictions) and 3%.

We have shown that joining an approach to information extraction with OCR error detection helps improve the performance of OCR error detection. We have also shown that this improvement occurs for more than one form of information extraction, specifically field segmentation and named entity recognition. We believe that similar methods can also be adapted to the task of error detection in other forms of text such as automatic speech recognition and other noisy language. After seeing similar results for two different information extraction tasks and two different qualities of OCR output we believe that the improvement in error detection is not due entirely to a particular labeling scheme or document. We also have no reason to believe that our approaches are dependent on the domain (i.e. specific to family history documents). Performance of our approaches may be affected by the difficulty of the information extraction task, although we have not yet tested this idea.

We expect to see further improvements to OCR error detection in future work. We have already implemented a character n-gram probability model to predict OCR errors, have trained it on the *error*-labeled training set and have evaluated it to a limited extent. It performed well on some of the tests and appears to be less apt to over-fit to the training sets than the HMMs used here. We are currently considering ways of combining the strengths of the character n-gram model with the HMM, i.e. by augmenting or replacing the HMM's emission model with the character n-gram model. We would also like to refine our approach to multi-task learning so that there is a looser coupling between the two tasks, e.g. by tying the parameter of compound labels that share the same error or semantic component. We also expect that utilizing the images of characters and words will also improve OCR error detection.

6. ACKNOWLEDGMENTS

We thank the Internet Archive (http://www.archive.org/), the Allen County Public Library (http://www.acpl.lib.in.us/), and Ancestry.com (http://www.ancestry.com/) for making available the historical documents used in this research.

7. REFERENCES

[1] M. S. Beach, W. Ely, and G. B. Vanderpoel. *The Ely Ancestry*. The Calumet Press, New York, New York, USA, 1902.

[2] V. Borkar, K. Deshmukh, and S. Sarawagi. Automatic segmentation of text into structured records. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, pages 175–186, Santa Barbara, California, USA, 2001.

[3] A. Carlson, J. Betteridge, R. C. Wang, J. Hruschka, and T. M. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 101–110, New York, New York, USA, 2010. ACM. ACM ID: 1718501.

[4] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[5] B. Chaudhuri and U. Pal. OCR error detection and correction of an inflectional indian language script. In *Proceedings of Thirteenth International Conference on Pattern Recognition*, pages 245–249, Vienna, Austria, 1996.

[6] S. Chen, D. Misra, and G. R. Thoma. Efficient automatic OCR word validation using word partial format derivation and language model. In L. Likforman-Sulem and G. Agam, editors, *Document Recognition and Retrieval XVII*, volume 7534, pages 1–10, San Jose, California, USA, 2010. SPIE.

[7] T. Grenager, D. Klein, and C. D. Manning. Unsupervised learning of field segmentation models for information extraction. In *Proceedings of the Forty-third Annual Meeting on Association for Computational Linguistics*, pages 371–378, Ann Arbor, Michigan, USA, 2005.

[8] C. Grover, S. Givon, R. Tobin, and J. Ball. Named entity recognition for digitised historical texts. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, Marrakech, Morocco, 2008.

[9] J. J. Hull and S. N. Srihari. Experiments in text recognition with binary n-Gram and viterbi algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4:520–530, 1982.

[10] K. Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24:377–439, 1992.

[11] D. Miller, S. Boisen, R. Schwartz, R. Stone, and R. Weischedel. Named entity extraction from noisy input: Speech and OCR. In *Proceedings of ANLP-NAACL*, pages 316–324, Seattle, Washington, USA, 2000.

[12] D. Miller, R. Schwartz, R. Weischedel, and R. Stone. Named entity extraction from broadcast news. *Proceedings of the DARPA Broadcast News Workshop*, pages 37–40, 1999.

[13] C. D. Mitchell and L. H. Jamieson. Modeling duration in a hidden markov model with the exponential family. In *International Conference on Acoustics, Speech and Signal Processing*, pages 331–334, Minneapolis, Minnesota, USA, 1993.

[14] U. Pal, P. K. Kundu, and B. B. Chaudhuri. OCR error correction of an inflectional indian language using morphological parsing. *Journal of Information Science and Engineering*, 16:903–922, 2000.

[15] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

[16] K. Taghva, B. Russell, and C. Jeffrey. The effects of OCR error on the extraction of private information. In *Document Analysis Systems VII*, pages 348–357, Nelson, New Zealand, 2006.

[17] J. B. White. *Barber Genealogy*. Press of the Nichols Print, Haverhill, Massachusetts, USA, 1908.

[18] S. Winters-Hilt. Hidden markov model variants and their application. *BMC Bioinformatics*, 7, 2006.

[19] S. Winters-Hilt and C. Baribault. A novel, fast, HMM-with-Duration implementation - for application with a new, pattern recognition informed, nanopore detector. *BMC Bioinformatics*, 8, 2007.