Intelligent Indexing: A Semi-Automated, Trainable System for Field Labeling

Robert Clawson, William Barrett

Brigham Young University, Provo, UT, USA

ABSTRACT

We present Intelligent Indexing: a general, scalable, collaborative approach to indexing and transcription of non-machine-readable documents that exploits visual consensus and group labeling while harnessing human recognition and domain expertise. In our system, indexers work directly on the page, and with minimal context switching can navigate the page, enter labels, and interact with the recognition engine. Interaction with the recognition engine occurs through preview windows that allow the indexer to quickly verify and correct recommendations. This interaction is far superior to conventional, tedious, inefficient post-correction and editing. Intelligent Indexing is a trainable system that improves over time and can provide benefit even without prior knowledge. A user study was performed to compare Intelligent Indexing to a basic, manual indexing system. Volunteers report that using Intelligent Indexing is less mentally fatiguing and more enjoyable than the manual indexing system. Their results also show that it reduces significantly (30.2%) the time required to index census records, while maintaining comparable accuracy. (a video demonstration is available at http://youtube.com/gqdVzEPnBEw)

1. INTRODUCTION

The problem we have approached in this paper is to reduce the tedium and increase the efficiency of indexing and transcription in handwriting recognition and field discrimination through interactive training and incremental learning. The application domain is structured documents (like Census records) where sufficient repetition exists to justify the use of a learning system to semi-automate the process. In this context, the indexer performs manual annotation (training), selects (deselects) matching (nonmatching) candidate word glyphs through visual consensus and group labeling, adjusting threshold(s) interactively, as needed.

FamilySearch Indexing[1] is an ambitious crowdsourcing project where volunteers index historical records one field at a time. They record information such as name, age, gender, marital status, and place of birth. These indexes allow genealogists to find information about ancestors contained in many types of documents with simple queries. FamilySearch's vast and growing collection of indexed records is possible only through the combined labor of the tens of thousands of volunteers who give many hours of time to the effort.

Repetitive tasks, like indexing, are strong candidates for automation. Thousands of man hours could be saved, and more work accomplished, if indexing could somehow be automated. Unfortunately, there are many image processing challenges to overcome in an end-to-end automated indexing system, and one of the most challenging is automated handwriting recognition. Recognizing handwriting automatically has been studied for many years, and yet continues to be impracticable except in constrained circumstances.

However, if we don't learn from user input, improvements in the efficiency of computerized indexing will be incremental at best, asymptoting in spite of improved user interfaces. Adding learning and assisted automated labeling can provide quantum leaps in throughput. However, our community is still trying to discover what that learning looks like.

Despite the challenges, some work has already been done in leveraging handwriting recognition to increase the rate of document indexing. However, these methods tend to silo the automated recognition and the manual recognition into two separated processes. Some fields are automatically labeled and the rest are manually indexed. Rather than take this approach, we present a semi-automated learning system which uses both the strengths of human indexers and handwritten word recognition technology.

Further author information: (Send correspondence to Bill Barrett)
Bill Barrett: E-mail: barrett@cs.byu.edu
Robert Clawson: E-mail: rtclawson@gmail.com

Document Recognition and Retrieval XXII, edited by Eric K. Ringger, Bart Lamiroy
Proc. of SPIE-IS&T Electronic Imaging, Vol. 9402, 94020A © 2015 SPIE-IS&T
CCC code: 0277-786X/15/$18 · doi: 10.1117/12.2076862

2. RELATED WORK

Closer integration and interaction of humans and computers is currently a highly researched area in the machine learning and pattern recognition communities.[2] Recently developing fields of study in this vein include active learning, semi-supervised learning, incremental learning, and interactive learning. We present here how these research areas are related to and differ from the work in this paper.

Previous work by the authors is also used throughout this work.[3–5] Doug Kennard's word morphing algorithm is used to generate similarity scores between fields of handwriting.[6–9]

2.1 Document Preprocessing

The author's previous work is used to perform image preprocessing.[10] Also, the Fourier-Mellin transform has been shown to be effective in determining the transform parameters to rectify a document image with strong delineated structure.[11] Consensus based techniques can help in discovering cell boundaries.[12]

2.2 Interactive Learning

Though in a different domain, the key contribution in Intelligent Scissors[13] is an excellent metaphor for the contribution of our research. Though graph search methods had been in use for many years in image segmentation, the addition of a human providing simple guidance in Intelligent Scissors made human guided segmentation a reality as a real time tool. Likewise, though handwriting recognition has been studied for many years, by pairing it with real-time human guidance, an efficient and improved indexing system is the result of our research.

We also consider the face labeling portion of Google Picasa™ to be very similar in concept to our research. Picasa automatically detects faces in images, and presents the cropped face images to the user to be labeled. As labels are paired with faces, Picasa will try to match labeled faces with unlabeled faces and automatically propagate the label to the other images with the same face, inviting user feedback and correction in the process. Thus, with minimal effort, an entire image collection can be indexed by a single person.

Doug Kennard also did work to demonstrate interactive training of the word warping algorithm for document annotation.[14]

Research has been done to provide a methodology for evaluating machine assisted annotation techniques.[15] The tool is employed to show that machine assistance can be used to increase both annotator speed and accuracy. The tool can also reveal the level of accuracy required by the machine assistance to produce these positive gains.

George Nagy is one of the foremost pioneers and proponents of interactive learning and *green* technology. He built the CAVIAR (Computer Assisted Visual Interactive Recognition) system, which is used for recognizing faces and flowers. CAVIAR has shown that interactive recognition is more than twice as fast as the unaided human, and yields an error rate ten times lower than state-of-the-art automated classifiers. Nagy states in his paper that "whether such an approach can be equally effective in the domain of documents as it is for flowers and faces is unproven, and adapting CAVIAR to document analysis requires further research."[16] Nagy later approaches interactive learning in documents and provides a good summary of existing methodologies.[17] Our research also addresses this subject.

3. INTELLIGENT INDEXING

Intelligent Indexing is a new methodology for annotation tasks that establishes a framework for learning from human interactions and improving over time. While using Intelligent Indexing, users will spend less time on repetitive actions and energy-wasting context switching and more time on higher-level tasks that require human intelligence.

Intelligent Indexing is suitable in any situation where repetition in the data can be leveraged to provide automation and reduce the load of manual effort. The system looks for similar items in the dataset, and allows the user to label whole sets of items at a time instead of each individually. The user has input into the classification boundary so that an individual's preference for precision versus recall can be satisfied. Should errors be present in the list of matched items, it is simple to fix these errors before they are applied (see Figure 1). Fixing errors in this way taps into a human strength that is reinforced from the beginnings of childhood with the likes of Sesame Street and the activity "which one of these is not like the other"[18] (see youtube link in references). This removes the need for post correction, and does so with surprisingly little cost.

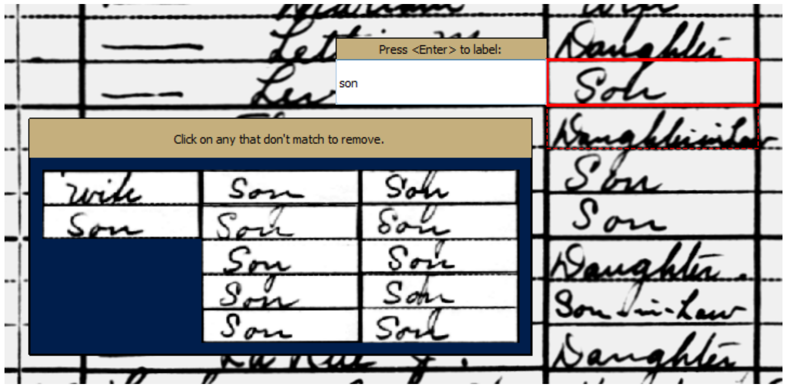


Figure 1. The word glyph “wife,” which is not like the others (“son”), can be deselected by clicking on it, after which the remaining word glyphs can be labeled as “son” with a single key stroke. This is the power of Intelligent Indexing.

3.1 Data

This section describes the data from the 1920 Utah census, used in Intelligent Indexing. The census is composed of a preamble that contains information about the census including the state, county, and enumerator (census taker). The rest of the document is a large table. The table has a heading describing each column, then room for fifty individual records. The columns on the census include name, relationship to head of household, gender, race, whether they can read or write, marital status, place of birth, father's place of birth, mother's place of birth, and occupation. The image preprocessing was performed as has been described in previous work.[10]

For Intelligent Indexing, we chose to focus on six of the possible columns of information. The data encountered in these columns is summarized in Table 1.

Census Category	**Possible values**
Relationship to Head of Household	Relationships like “Wife”, “Daughter”, “Son”, or “Brother”.
Gender	(M)ale, (F)emale
Marital Status	(M)arried, (S)ingle, (W)idowed, (D)ivorced
Place of Birth, Father's Place of Birth, Mother's Place of Birth	Place names like “Utah”, “Wales”, or “New York”

Table 1. Data categories from the 1920 Utah census used for testing Intelligent Indexing.

3.1.1 Metadata

For each image in the document collection, a matching metadata file is created. The metadata stored contains the field location on the page, the category of each field, the enumerator for each page, the ground truth label for each field as provided by FamilySearch Indexing, the label provided by the indexer in the Intelligent Indexing client, whether the label was applied automatically, and the time it took to label the field.

3.1.2 Precomputing Morphing Costs

The similarity scores generated using handwriting recognition are the backbone of the automated labeling system. For our system, these similarity scores were precomputed to guarantee interactive speeds while the client is running. The census collection was split into chunks according to handwriting style. This was possible because the enumerator (the census collector) wrote their name on each page, and because the census had been indexed previously, with the enumerator field included in the index. The similarity scores for a particular category and a particular group are stored in a file we call a cost matrix.

Six of the twenty-seven categories on the 1920 census were chosen to test Intelligent Indexing. These are “Birthplace”, “Marital Status”, “Relationship to Head of Household”, “Father's Birthplace”, “Mother's Birthplace”, and “Gender”. For each of these categories, forty-two groups were created, one for each enumerator. The groups were simply given the name of the enumerator that defined that group.

The reasons for chunking the collection into smaller groups are twofold. First, because each field is compared to each other field in a category, there is an inherent $O(n^2)$ complexity in computing the similarity scores. Anything that can be done to reduce the size of n will reduce the compute time. Second, we have shown in previous work[10] that handwriting recognition accuracy drops significantly when comparing between different handwriting styles and out-of-vocabulary words. The main ramification of the decision to break the collection into groups is that learning in one group does not transfer to the next group.

Figure 2 illustrates a very simple cost matrix. The first occurrence of "Head" in Figure 2 has a cost of 0 when compared with itself and a cost of 3.2 when compared with another instance of "Head." But 3.2 is still much smaller than 8.0 ("Wife" compared with the first instance of "Head") or 9.3 ("Wife" compared with the second instance of "Head"). Note the increase in cost the more different the word glyphs are (for example, "Daughter" compared with "Wife", "Head" (1) or "Head" (2)). This also allows candidate word glyphs (see Section 3.2.1) to be previewed in order of increasing cost, putting the most likely matches at the top of the list. Precomputing these Word Morphing costs allows comparisons to be performed at interactive rates using a simple look-up that is performed during interactive cell labeling by the indexer.

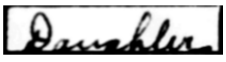
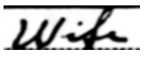
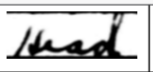
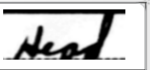
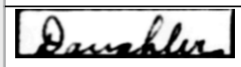
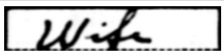
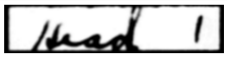
Relationship to head of household				
	0	20.2	15.9	11.4
		0	8.0	9.3
			0	3.2
				0

Figure 2. Example cost matrix.

3.2 Interactive Field Labeling

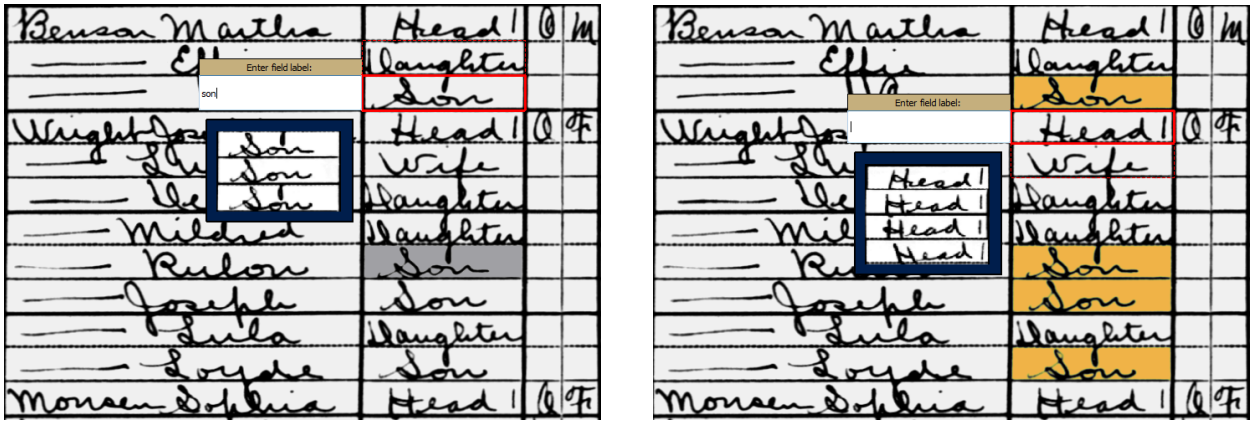
Document indexing is essentially composed of many individual labeling events. This section covers the different moving parts at work during field labeling. The use of training sets is discussed, by which labeled fields in each of the different categories are used to help label future occurrences. Also, thresholds exist for each category of data and can be tuned by the indexer. These thresholds govern how aggressive the recognition engine is. Thresholds are covered in section 3.2.4.

When the recognition engine recommends fields to the indexer, these are displayed in a preview window. The indexer has the opportunity to deselect any fields that don't match, after which the indexer applies the label to all the fields at once (see Figure 3). This mechanism is important because it allows the indexer to make quick adjustments to the automated learning when it happens instead of post correcting later. This is one way in which the intelligence of the indexer is leveraged. The process of approving the fields in the preview window is called *visual consensus* and is covered in section 3.2.2. By these means, labels are applied to fields in groups.

Fields with labels are grouped by color according to their labels. The column headers for the columns that should be indexed are highlighted in red so that indexers can easily tell which fields to index. When columns are completed, this column header indicator changes colors so that indexers can be sure that they have not missed a field. While visual consensus is likely the most important feature for avoiding labeling errors, this field coloration scheme is meant to assist in maintaining accuracy as well.

3.2.1 Recommending Fields for Automated Labeling

There are two opportunities for the recognition engine to recommend fields for automated labeling. The first is when a field is selected. When a field is selected, all unlabeled fields are compared to it to find those that match below the current threshold for that category, as shown in Algorithm 1. Those that do match below the threshold are presented to the indexer, who can remove any that don't match before applying the label (see Algorithm 3). Once the label is applied to the currently selected field, all fields that remain in the preview window are labeled as well.



(a) Indexer selects a field (red box) and types the label ("son") in the input box to the side. Other fields with a similarity score below a threshold are presented to receive the same label.

(b) All the "son"s are labeled, and the next field is selected.

Figure 3. Before and after the indexer labels a field.

The second opportunity is a transitive learning step which follows a labeling event. A visualization of how this transitive labeling works is provided in Figure 4. In 4(a), the indexer selects a field (marked in green). Matches to that field (purple word glyphs) are then presented to the indexer (4(b)). Both green and purple word glyphs are added to the training set. After the indexer approves these fields and labels them, the neighbors of the training set (pink word glyphs) are confirmed as well (4(c)). These neighbors are calculated using Algorithm 2.

This kind of "transitive learning" leverages and propagates the original label onto as many word glyphs as possible without having to re-key the label, thereby increasing efficiency and minimizing keyboard errors.

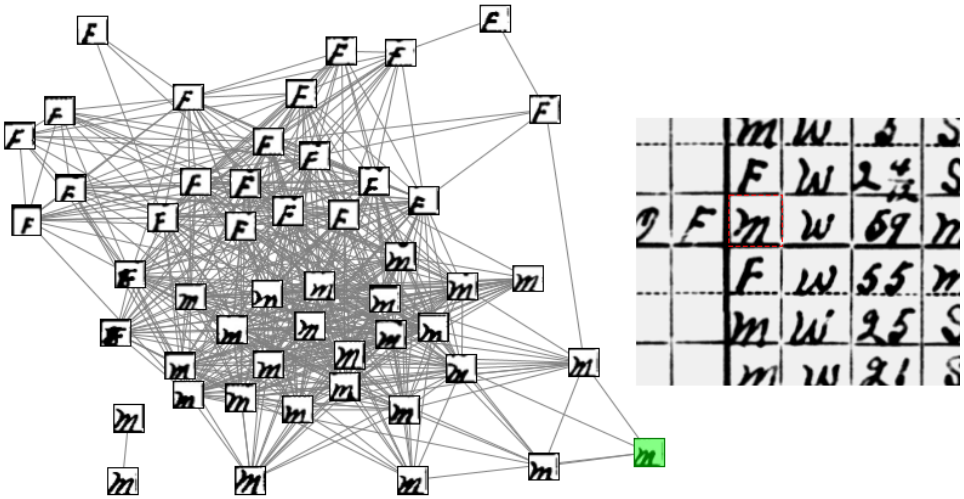
Algorithm 1 Choosing candidate word glyphs (*cwgs*)

1: **Input:**
   f = active field
   ufs = List of unlabeled fields with the same category as f
2: **Output:** List of unlabeled fields to display to the user as *cwgs*
3: $matchingFields \leftarrow \{\}$
4: **for all** *uf* : *ufs* **do**
5: **if** WORDMORPHCOST(*f*,*uf*) $<$ THRESHOLD **then**
6: *matchingFields* += *uf*
7: **end if**
8: **end for**
9: **return** $matchingFields$

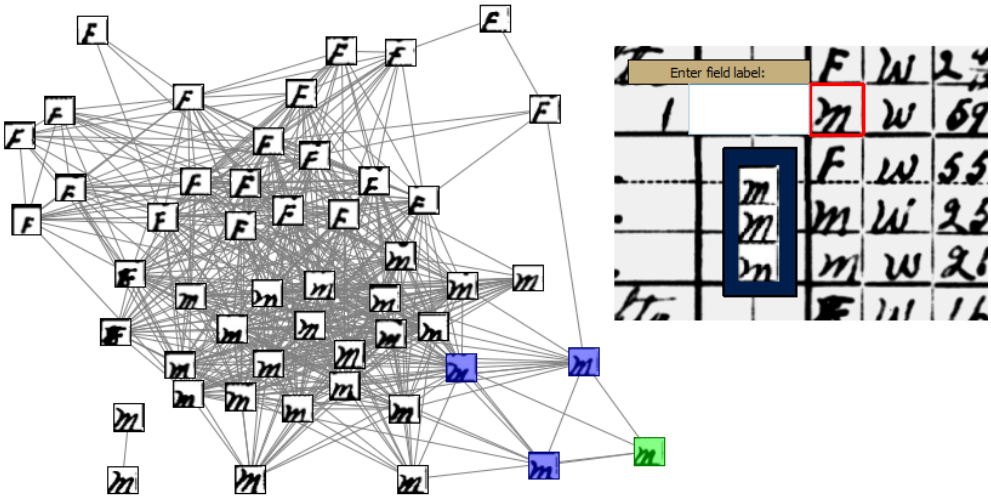
3.2.2 Visual Consensus

When an indexer selects a field, the cost matrix is used to determine which fields on the page are similar to the currently selected field. The threshold for the selected field's category determines how many fields are considered candidate matches. These candidate word glyphs (*cwgs*) are displayed to the indexer so that they can be validated on the spot. This interaction is described in Algorithm 3. The user interface presents the matching fields to the indexer (see Figure 3) so that they can be confirmed (See Algorithm 3, line 2). Fields that don't match are removed by either clicking on them directly or by reducing the threshold (lines 5-7). Finally, all of the labeled fields are added to the training set (line 10).

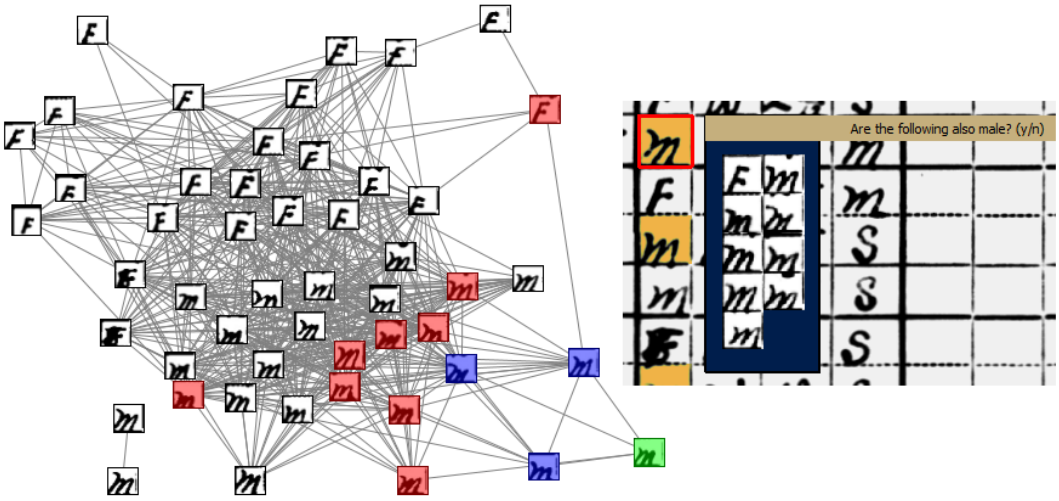
After the interaction described in Algorithm 3, the selected field is considered for transitive learning, where potential matches to the training set for the selected field's category are calculated and presented for transitive labeling. These are validated by the user in a similar manner as the matches to the selected field, with the added advantage that the label does not need to be re-keyed since it is already known.



(a) Indexer selects a field (green).



(b) The selected field's neighbors are prompted to the indexer (purple).



(c) After the indexer applies the label to the green (and purple) fields, the neighbors of the neighbors of the selected field are prompted to the indexer (pink).

Figure 4. Interactive, transitive learning from an incremental training set.

Algorithm 2 Calculating training set matches (Transitive Labeling)

Input:
f = last indexed field
ts = Training set corresponding to the category of f
ufs = List of unlabeled fields
Output: List of unlabeled fields to display to the user as matching candidates
$matchingFields \leftarrow \{\}$
for all uf : ufs **do**
    for all $tsField$: ts **do**
        if $tsField$.Label $\neq f$.Label **then**
            continue
        end if
        if WORDMORPHCOST($tsField$,uf) $<$ THRESHOLD **then**
            $matchingFields$ += uf
        end if
    end for
end for
return $matchingFields$

Algorithm 3 Interactive field labeling

Indexer **selects** a cell, C, to label (Ex. cell containing handwriting "Son") - Figure 3(a)
Computer **presents** candidate word glyphs, ($cwgs$) - Algorithm 1
Indexer **keys** in a label, L for C (Ex. Indexer types "Son" in input box) - Figure 3(a)
while $\exists f \in cwgs$ s.t. $Label(f) \neq L$ **do**
    user deselects $f \in cwgs$ that do not match C by
    a. clicking on $f \in cwgs$ that do not match and/or
    b. adjusting threshold slider - Figure 5
end while
all $cwgs$ are assigned label L - Figure 3(b)
all $cwgs$ are added to training set

3.2.3 Removing a field from the preview window

In earlier iterations of Intelligent Indexing, there was no preview window. After a field was labeled, all fields considered matches to that field were automatically labeled, and it was up to the indexer to figure out what mistakes had been made. This was tedious, time-consuming, and frustrated the indexer. The preview window was a critical revelation, as it allows the indexer to proactively remove mistakes before they happen.

To remove a field from the preview window, the indexer either clicks on it or adjusts the threshold until the field is no longer considered a match. The fields in the preview window are sorted by match strength, so the fields most likely to be mismatched are clumped at the end of the list. This makes it easy to either click out all the mistakes, or see how the threshold is affecting the list.

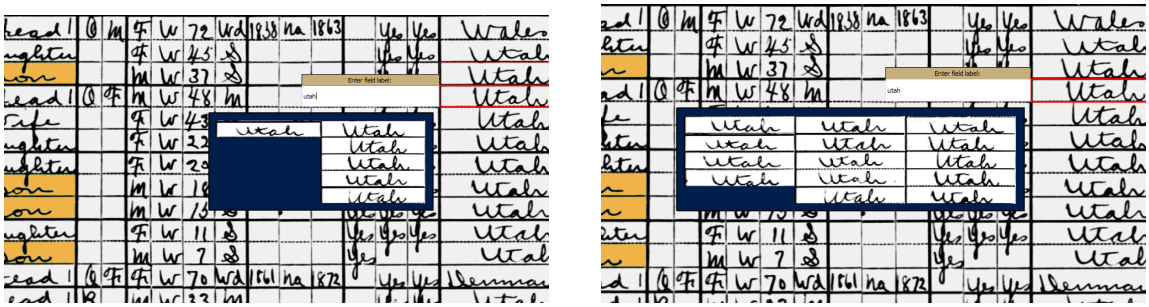
Currently, clicking on a field simply removes it from the match list. However, as described in Future Work, we see opportunity to leverage this information in other ways as well.

3.2.4 Thresholds

A proper setting of the threshold is crucial for Intelligent Indexing to save time for the indexer. A threshold set too low means little or no automated labeling occurs. A threshold set too high results in many spurious fields showing up in the preview window, which will tend to frustrate the indexer.

For low thresholds, the accuracy is very high, but there are also few matches below the threshold. As the threshold increases, the percentage of fields with at least one match below the threshold grows and approaches 100%. However, the accuracy also continues to decrease. The accuracy will plateau when the threshold is such that all fields have a nearest neighbor below the threshold.

In our user interface, the indexer is given a slider that can be used to adjust the threshold to suit their preference. There is also a keyboard shortcut for adjusting the slider. Setting this threshold to suit preference is a task well suited for the indexer. The effect of changing the threshold is immediate, meaning that if a field is selected and the slider is adjusted, the matching fields are reevaluated on the spot. This interaction is shown in Figure 5.



(a) Indexer selects a field (red box). All fields with a similarity score below a threshold are presented to receive the same label.

(b) Increasing the threshold increases the number of matching word glyphs in the preview window, allowing more cells to be labeled.

Figure 5. Before and after the indexer adjusts the threshold.

3.3 Minimizing Context Switching

3.3.1 Interacting directly with image

FamilySearch Indexing current displays the document image and below it a form or spreadsheet for entering the labels. One problem with this approach is that it separates spatially the label from its matching field. This makes it difficult for the indexer to be sure that the right information ends up in the right cell. Also, the entry form takes up valuable screen real estate. The result is that the indexer is looking at the page with tunnel vision. Worse, the indexer has to context switch, moving their gaze or focus of attention each and every time a label is entered, making it easy to get off track.

Rather than take this approach, we have the user interact directly with the image. To select a field to label, the indexer simply clicks on it (in Figure 6 part B, the user has clicked on the "Utah" field). A red rectangle marks the place on the page where the indexer is currently working, not unlike a cursor in word processing software. There is also the question of how to indicate to the user what data on the page should be indexed. With the spreadsheet indexing layout, the fields that should be indexed are inherent to the design of the spreadsheet. In our system, however, column headers are colored to specify which columns should be indexed, and only fields that need to be indexed are selectable. A red column header indicates more work is necessary, while a green column header indicates that the column is done.

3.3.2 Working down columns

On a census page, records of an individual are presented across rows. It is perhaps natural then to advance the selected field across rows, allowing the indexer to record the information for one individual at a time. However, the items of information about a person, including the relationship to the head of household, the gender, the marital status, and the birth place, are dissimilar. Horizontal movement across categories causes the indexer to have to bring back to mind the different possibilities and rules associated with that category.

To minimize context switching between categories, we advance the field down the column. In this way, the indexer enters all of the fields in one category first, then moves on to the next category. In the end, this amounts to an assertion that the context of the column is more important than the context of the individual's record. As can be see in Figure 6, the indexer is working down the "Birth Place" column.

3.3.3 Label entry adjacent to field

Originally we had the label editor on the side bar of the application. However, this was a violation of our "minimize context switching" principle. By placing the label editor adjacent to the field, it is now much easier to see both the field and entered text (see part B of Figure 6). We believe this is not only faster and easier for the user, but will reduce the number of mistakes that are made.

3.3.4 Preview matches are localized

Deciding where to put the auto-labeling preview was difficult. On the one hand, overlaying fields on top of the document image risks being too confusing for the indexer. On the other hand, we did not want to break our "minimize context switching" principle and place the preview on the sidebar. In the end, we chose to allow the indexer to keep their focus locally, and to take measures to ensure the preview stands out from the page. An example of how this looks in the software can been seen in part D of Figure 6, where the candidate matches to the currently selected "Utah" field are displayed below the label entry form.

3.3.5 Labels displayed adjacent to fields

Even while using colors to link fields with the same label, indexers will still want to be able to see exactly what they typed to correct any possible mistakes. Displaying these labels all the time for all fields would get distracting, so we display them only for the category of the field currently selected (blue and green machine-printed text spelling "Utah" and "England"). The labels do obscure part of the image, but we consider this to be an acceptable consequence, since it obscures only a part of the image that is not germane to the current task (for an example, see part E of Figure 6). We color the machine-printed text blue for fields the indexer has labeled, and green for fields that have been automatically indexed.

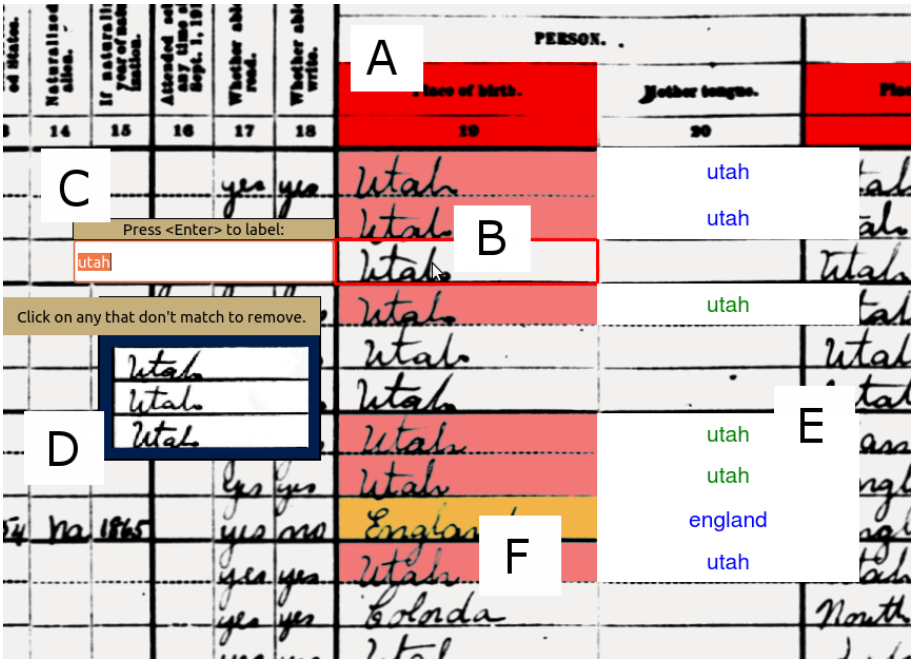


Figure 6. Intelligent Indexing user interface. Notice that everything the indexer needs for the current task is localized, so minimal context switching is required. A) Column headers are red for columns that need to be indexed, and turn green when those columns are completed. B) The active field has a solid red outline. C) The label entry form is placed directly beside the active field. D) Candidate word glyphs from the recommendation engine are presented in a preview window. E) Labels for the fields already indexed are shown. Blue means they have were indexed manually, green for automatically. F) Labeled fields are colored according to their label.

4. RESULTS

We performed a user study to gather data about indexing times, accuracy, and automation while using Intelligent Indexing. We present those results in this section, as well as provide some visual examples of where we felt the system performed well, and also where it made mistakes.

4.1 Quantitative Results

4.1.1 Indexing Time

The indexing times for individual fields were aggregated to produce an indexing time per page. For the ten indexers who completed both batches, Intelligent Indexing on average decreased the indexing time by an impressive 2 minutes (117

seconds) per page. The average completion time per page across all participants was 7 minutes 30 seconds for basic indexing, and 5 minutes 46 seconds for Intelligent Indexing, which is a 30.22% reduction in time. This decrease in the time it took to index using Intelligent Indexing was statistically significant ($p<0.05$). For all but one experiment, Intelligent Indexing was faster than the basic indexing.

4.1.2 Indexing Accuracy

Through comparison to ground truth data provided by FamilySearch, the volunteers were found to be 98.85% accurate for basic indexing on average. For Intelligent Indexing, the accuracy for manually indexed fields was 96.58%, and the accuracy for automatically indexed fields was 98.46%, with an aggregated accuracy of 98.12% accurate for Intelligent Indexing. The difference in accuracy between Intelligent Indexing and basic indexing was not measured to be significant ($p<0.05$).

Errors came in a couple different forms. A common mistake was to misspell the more difficult state and country names. Another mistake was typing a single letter instead of the full phrase (in cases other than "Gender" or "Marital Status"). Presumably, this occurred at transitions between columns where the indexer expected the auto complete function to work. Another small set of "mistake"s were ones in which the entered label was semantically equivalent to the ground truth, but the two were counted as different because of, for example, abbreviations.

4.1.3 Indexing Automation

Volunteers using Intelligent Indexing had 59.18% of the fields filled out automatically on average per page. This automation rate ranged from 44.33% to 76.33%. In Figure 7 a scatterplot shows the correlation between the time to index a page using Intelligent Indexing and the percent of fields automatically indexed (Automation rate). The R^2 value for the scatterplot is 0.45, showing a strong correlation between automation rate and indexing time. This correlation shows that the difference between the times to index using the basic indexing system and Intelligent Indexing may be attributed to more than just the switching of row-based indexing to column-based indexing. It suggests that a higher automation rate produces faster indexing times.

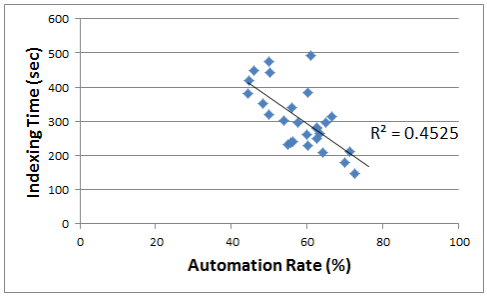


Figure 7. Scatterplot showing correlation between automation rate and time to index a page. Calculated for all pages indexed using Intelligent Indexing, with outliers removed.

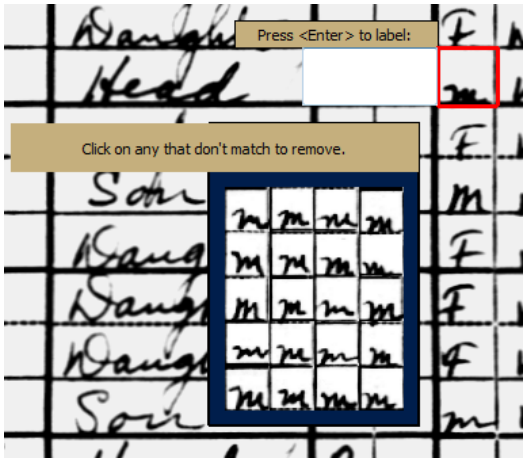
4.2 Qualitative Results

Some volunteers noted that the pages chosen to test Intelligent Indexing were more difficult to index than those chosen to test the basic indexing. The basic indexing pages were hand chosen, while the pages for Intelligent Indexing were pulled randomly from the census. Random pages were chosen so that the results were not dependent on the particular image. The logic at the time for making the basic indexing pages the same was so that the different volunteers could be directly compared. Without making a formal comparison, we are left to speculation. However, it seems likely that the Intelligent Indexing results are negatively biased by the fact that more difficult pages were used to test it.

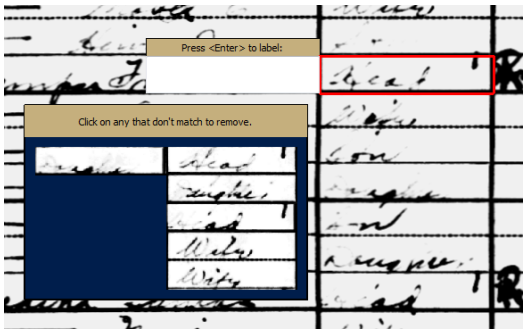
The critical responses to Intelligent Indexing usually had to do with minor bugs or issues with the user interface. Also, some felt that the experiment needed more instructions. For positive responses, a common sentiment was that column-based indexing was much better than row-based indexing. Also, those who figured out how to use the adjustable threshold found it to be effective. Interestingly, not everyone used the threshold in the same way. Some adjusted the threshold as they went and used the adjuster to remove mismatches from the recommendations. Others set a very generous threshold, allowing for several mistakes but capturing more of the correct matches as well. In this case, the work flow was to click to remove the mismatches from the recommendations window.

4.3 Visual Results

This section is composed of screen captures that present the preview window under a variety of conditions. Figure 8(a) shows an example where a high degree of automation is possible. In Figure 8(b), we see the effects of a degraded page on accuracy. Finally, in Figure 1, we show how easy it is to pick out word glyphs that don't belong in the preview window.



(a) The small vocabulary, single character fields worked out very well in general.



(b) A degraded page made recognition difficult.

Figure 8. Visual results of Intelligent Indexing.

5. FUTURE WORK

As we delved into the heart of this research, we soon realized that there was much more to be done than we had time for. Future work could be applied to many avenues. A few might be: handling more difficult categories, harnessing between-category dependencies, automatically learning and adjusting the threshold, finding the optimal order of labeling fields (active learning), improving preprocessing, using weights to fine tune the participation of words in the training set, or incorporating alternative handwriting recognition algorithms (perhaps in a consensus model).

6. CONCLUSION

Intelligent Indexing provides a general, scalable, collaborative approach to indexing and transcription of non-machine-readable documents that exploits visual consensus and group labeling while harnessing human recognition and domain expertise. Previewing and selecting candidate matches is far superior to conventional, tedious, inefficient post-correction and editing. Results show that Intelligent Indexing reduces significantly the time required to index census records, while maintaining comparable accuracy. While the software developed demonstrates "proof of concept" for Intelligent Indexing, we propose this as more of a strategy than a specific solution - an illustration of how the indexer and software can be tightly coupled in a cooperative, symbiotic framework, training each other and harnessing the best of both. Finally, we have discussed possible improvements to Intelligent Indexing that can further enhance recommendation accuracy, make more use of available information, adapt the system to data categories with larger vocabularies, allow Intelligent Indexing to be used on a variety of devices, and improve the user experience.

REFERENCES

[1] [*Fast Facts about FamilySearch Indexing*] (Jan 2012). http://www.mormonnewsroom.org/article/fast-facts-about-familysearch-indexing.

[2] Little, G. and Sun, Y., "Human OCR: Insights from a complex human computation process," in [*Workshop on Crowdsourcing and Human Computation, Services, Studies and Platforms, ACM CHI*], (2011).

[3] Clawson, R. T. and Barrett, W., "Extraction of handwriting in tabular document images," in [*Family History Technology Workshop 2012 (FHTW2012@Rootstech) (http://fht.byu.edu/prev_workshops/workshop12/)*], 76–79 (February 2012).

[4] Clawson, R. and Barrett, W., "Green icr: Semi-automated census record indexing with emphasis on human computer interaction," in [*Family History Technology Workshop 2013 (FHTW2013@Rootstech (http://fht.byu.edu/)*], (March 2013).

[5] Clawson, R. and Barrett, W., "A semi-automated, trainable system for field labeling," in [*Family History Technology Workshop (fht.byu.edu)*], (March 2014). https://www.youtube.com/watch?v=gqdVzEPnBEw.

[6] Kennard, D. J., Barrett, W. A., and Sederberg, T. W., "Word warping for offline handwriting recognition," in [*Document Analysis and Recognition (ICDAR), 2011 International Conference on*], 1349–1353, IEEE (2011).

[7] Kennard, D. J., Barrett, W. A., and Sederberg, T. W., "Many-author offline handwriting recognition using a warping-based approach," in [*Family History Technology Workshop 2013 (FHTW2013@Rootstech) (http://fht.byu.edu/)*], (March 2013).

[8] Kennard, D. J., Barrett, W. A., and Sederberg, T. W., "Offline signature verification and forgery detection using a 2-d geometric warping approach," in [*Pattern Recognition (ICPR), 2012 21st International Conference on*], 3733–3736, IEEE (2012).

[9] [*Warping Based Approach to Handwriting Recognition*] (Mar. 2013). https://www.youtube.com/watch?v=eBQjHgejchA.

[10] Clawson, R., Bauer, K., Chidester, G., Pohontsch, M., Kennard, D., Ryu, J., and Barrett, W., "Automated recognition and extraction of tabular fields for the indexing of census records," in [*Document Recognition and Retrieval XX [8658-17]*], International Society for Optics and Photonics (February 2013).

[11] Hutchison, L. A. and Barrett, W. A., "Fourier–mellin registration of line-delineated tabular document images," *International Journal of Document Analysis and Recognition (IJDAR)* **8**(2-3), 87–110 (2006).

[12] Nielson, H. and Barrett, W., "Consensus-based table form recognition of low-quality historical documents," *International Journal of Document Analysis and Recognition (IJDAR)* **8**(2-3), 183–200 (2006).

[13] Mortensen, E. and Barrett, W., "Intelligent scissors for image composition," in [*Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*], 191–198, ACM (1995).

[14] Kennard, D. J. and Barrett, W. A., "Interactive training for handwriting recognition in historical document collections," in [*Document Recognition and Retrieval XIV*], 65000E–65000E, International Society for Optics and Photonics (January 2007).

[15] Felt, P., *Improving the Effectiveness of Machine-assisted Annotation*, PhD thesis, Brigham Young University. Department of Computer Science (2012).

[16] Nagy, G. and Zou, J., "Interactive visual pattern recognition," in [*Pattern Recognition, 2002. Proceedings. 16th International Conference on*], **2**, 478–481, IEEE (2002).

[17] Nagy, G. and Veeramachaneni, S., "Adaptive and interactive approaches to document analysis," in [*Machine learning in document analysis and recognition*], 221–257, Springer (2008).

[18] [*Sesame Street - "One of these things..." (Bird seed)*] (Nov 2007). https://www.youtube.com/watch?v=ueZ6tvqhk8U.