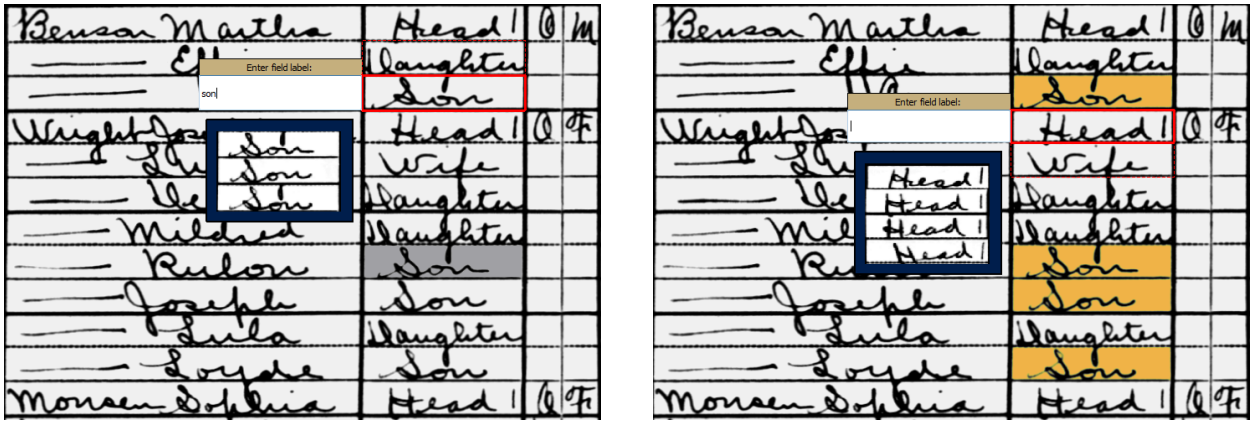
Intelligent Indexing: A Semi-Automated, Trainable System for Field Labeling

Robert Clawson, William Barrett

Brigham Young University, Provo, UT, USA

ABSTRACT

Historical documents without an index are both plentiful and valuable. Solutions like FamilySearch Indexing and Amazon Mechanical Turk provide a way to index these documents through manual effort. We present Intelligent Indexing as a methodology for both maximizing the return on the indexer's effort, and so that the system can improve over time. The methodology is defined by three guiding principles: build an enabling user interface, minimize context switching, and learn from every user interaction. We describe these principles generally, and show their application to the 1920 Utah census. A user study will be performed for quantitative and qualitative evaluation of our implementation of the methodology.



(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 1. Before and after the indexer labels a field.

1. INTRODUCTION

We present a remarkable new paradigm for field labeling which is faster, easier, more enjoyable, and that learns and improves over time. Intelligent Indexing works so well because it brings together both the intelligence of the indexer and the computing power of the machine in a highly coupled system. Previous work[1–4] in semi-automated labeling focuses more on the recognition system and less on the interaction with the indexer. Generally speaking, the paradigm in these papers has not shifted much from the standard machine learning model of automating as much as possible, and leaving human interaction to correct mistakes and label fields rejected from the model. The rest of this paper is focused on methods with a brief conclusion at the end.

2. RESEARCH DESCRIPTION

2.1 Preprocessing

Document images in a collection are scaled, translated, binarized, and zoned. A template for the collection is used in these processes. After preprocessing, each field in the table can be isolated. Isolating the handwriting

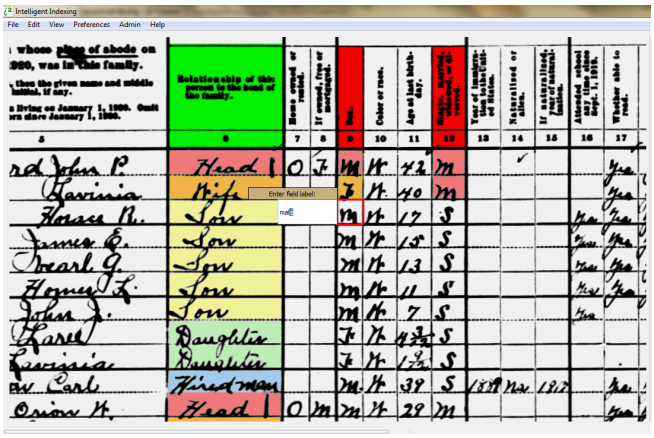


Figure 2. Column header coloring indicates progress through the page (green means the column is done). Cell coloring helps the indexer quickly see which fields have been indexed. Also, each label is assigned a different color, so fields that should have the same label, but have a different color, can be quickly identified.

in the document allows the system to compare them to each other and suggest labels. The preprocessing step is nontrivial, but also is not the focus of this research, so existing technology will be used "out of the box".[5]

To avoid the risk of having an unresponsive UI, or to be unable to perform the level of field recognition we would like in real time, all field comparisons are precomputed. This can be done with any handwriting recognition system; we have chosen to use a whole word approach[6,7] that generates a similarity score, where a low score represents a closer match. Fields are compared across multiple documents so that the system can build a growing labeled corpus, but we stratified the 1920 Utah census by enumerator and only compare fields within documents prepared by the same enumerator. The ramifications of this decision was that learning starts over for each enumerator. There are 42 enumerators for the over 1100 document images in the census. However, it is not clear that having a growing example set across multiple authors is even desirable, since handwriting recognition accuracy decreases significantly when comparing across multiple authors.

2.2 Build An Enabling User Interface

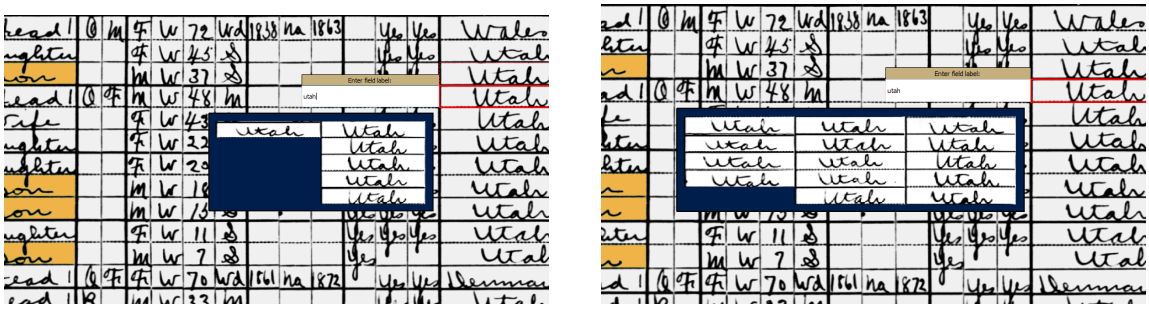
For most categories, the same words comes up again and again. Instead of requiring the user to type these words in over and over, auto-complete will finish what the user is typing as long as it is a label that has been previously seen.

We also wanted some way to quickly spot check that fields have been labeled correctly. One way to do this is to color the fields according to their label. For example, in the marital status category, all fields labeled "married" are the same color, and all fields labeled "single" are another color.

Outlining fields in red as they are moused over is a subtle UI element that reduces confusion about which fields need to be indexed on the page.

Auto advancing the selected field saves dozens of mouse clicks or key presses for each image indexed, and allows the indexer's hands to stay on the keys.

Making the indexer scroll the page is a waste of time when it can be prevented. The display should automatically adjust itself so that everything that needs to be displayed can be displayed. Since the selected field is automatically advanced, this will sometimes move it off of the displayed portion of the image. Keeping it on screen is performed by automatically scrolling the image.



(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 allows for more matches to be made.

Figure 3. Before and after the indexer adjusts the threshold.

2.3 Minimize Context Switching

FamilySearch Indexing current displays the document image and below it a form or spreadsheet for entering the labels. The 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 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. 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.

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 of an 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 the context switch between categories, we advance the field after a labeling event 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.

Originally we had the label editor on the side bar of the application. However, this was a violation of the 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. We believe this is not only faster and easier for the user, but will reduce the number of mistakes that are made.

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 the 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 sticks out from the page. An example of how this looks in the software can been seen in Figure 1, where the labels "Son" and "Head" are previewed in figures (a) and (b) respectively.

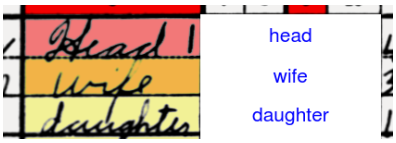


Figure 4. Labels are displayed beside the fields.

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 for 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. These 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. An example of these labels is shown in Figure 4.

2.4 Learning From Every Interaction

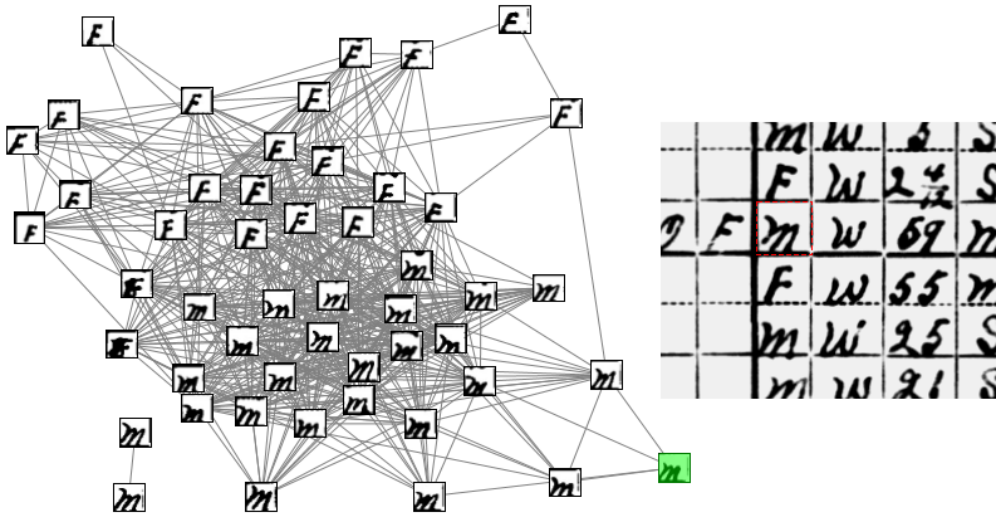
Perhaps the most straightforward interaction to learn from is when a field is selected and labeled. We learn that the selected field should have that label, and that other fields with a small morph cost to that field should likely also have the same label. Our current implementation is to have a threshold under which all fields receive the entered label. The user interface presents these fields to the indexer so that they can be confirmed. Finally, all of the labeled fields are added to the training set.

When a labeled field is corrected, it is either because the indexer changed their mind about a field, mistyped, or failed to prevent an automatic labeling from applying a label erroneously. In any of these cases, it is possible that other labeled fields in the column have the same error, so we learn from this interaction by considering all labeled fields and comparing them to the corrected field. Any fields under the current threshold are prompted as candidates for also having their label changed.

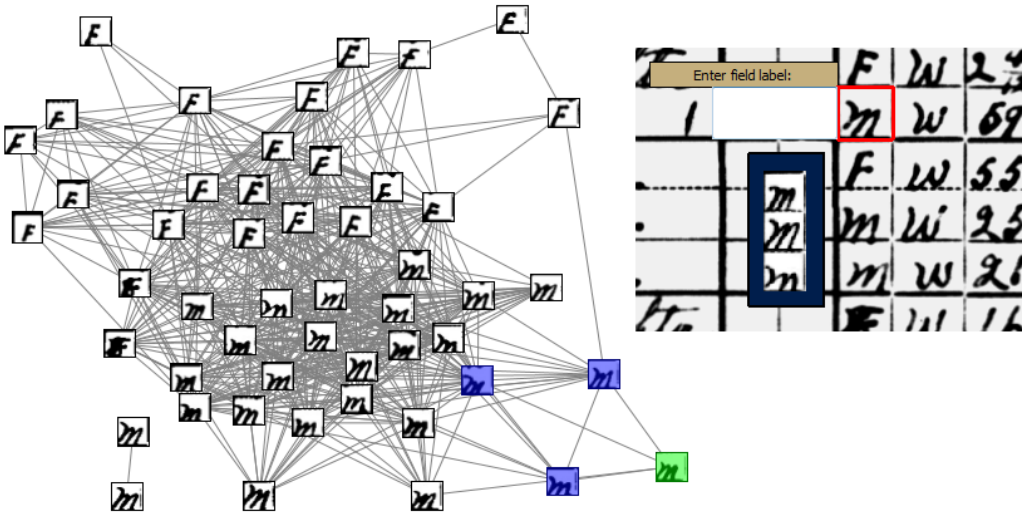
If a candidate field for labeling is removed from the list of matches, either as a match to the current selection or a match to the training set, we have learned two things. First, we have learned that the removed field does NOT have the currently selected field's label. In cases where the number of possibilities for the column is severely constrained, like gender or marital status, the actual label can almost be inferred from knowing what the label is not. However, in general this information is not immediately useful, except that the field can be removed from match lists for the erroneous label. But it can also be marked as not having a particular label, commencing a process of labeling by elimination. Second, we have learned that the removed field is very similar in terms of morph cost to other fields that should have a different label. Conceptually, this information could be used to help differentiate between two sets of fields that cluster close to each other. However, currently we simply use a global threshold and a simple, unweighted nearest neighbor approach for deciding matches, so we are unable to fully capture this information at this time. A simpler use of the information is to mark the removed field so that it cannot participate in automatic labeling, and must be labeled manually. Since it is more like different fields than its own, this can prevent future mislabelings as well.

As the indexer proceeds through the page, and onto other pages, the number of labeled examples in each category increases. As the size of these training sets increases, it becomes reasonable to calculate statistics about the data. For example, the number of fields assigned each of the possible labels in the category could be heavily skewed. This information could translate into a priori probabilities that an unlabeled field is in a given category.

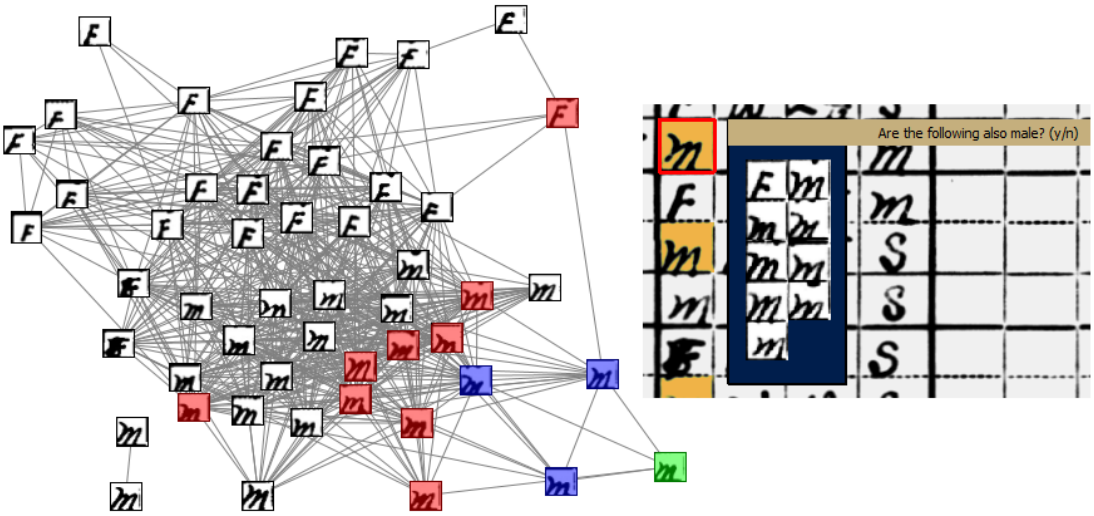
Not only can the neighbors of a selected field be labeled automatically, but potentially the neighbors of those neighbors as well. This transitive learning could get out of hand, ie too much learning from one field. In practice one transitive step often yields a few extra labels without too many mistakes. A visualization of how this transitive labeling works is provided in Figure 5. In 5(a), the indexer selects a field. Matches to that field



(a) Indexer selects a field (green).



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



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

Figure 5. Interactive, transitive learning from an incremental training set.

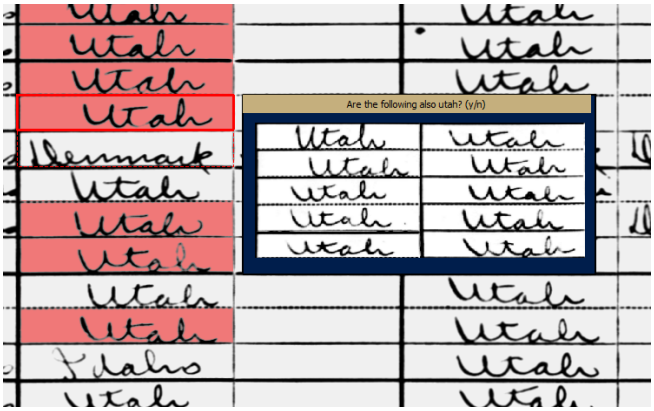


Figure 6. Matches to the training set are a separate prompt that occurs after a labeling event.

are then presented to the indexer (5(b)). After the indexer approves these fields and labels them, the neighbors of the neighbors are confirmed as well (5(c)).

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 automation prompts, which will tend to frustrate the indexer. For the latter reason especially, the indexer is given a slider that can be used to adjust the threshold (there is also a keyboard shortcut for adjusting the slider). Rather than having the indexer set the threshold, it would be better to learn it automatically. Through the course of indexing the fields in a category, the indexer, without ever touching the threshold slider, is implicitly describing where the threshold should be. When the indexer labels a field, then labels another field with the same label, the implicit fact is that the threshold should have been higher, so that the second field would have been automatically labeled with the first. So, when such an event occurs, the threshold is increased so that retrospectively the second field would be automatically labeled by the first. Contrarily, whenever a field is removed from either a list of training set matches or matches to the current selection, this is evidence that the threshold is too high and should be decreased.

3. CONCLUSION

This paper has largely been a discussion of methods. Conspicuously absent is the presence of a results section. We are currently in the process of obtaining results, but anecdotally, the system performs very well. For instance, we were able to label a document in 6 minutes 12 seconds without automation, and in 3 minutes 7 seconds with automation. Being able to double throughput would be a big win, and we feel that the system will do even better on subsequent pages after it has had a chance to learn. Qualitatively, Intelligent Indexing is a more enjoyable experience that is, dare we say it, fun.

REFERENCES

[1] Vajda, S., Junaidi, A., and Fink, G., "A semi-supervised ensemble learning approach for character labeling with minimal human effort," in [*Document Analysis and Recognition (ICDAR), 2011 International Conference on*], 259–263, IEEE (2011).

[2] Sibade, C., Retornaz, T., Nion, T., Lerallut, R., and Kermorvant, C., "Automatic indexing of french handwritten census registers for probate geneaology," in [*Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*], 51–58, ACM (2011).

[3] 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).

[4] Richarz, J., Vajda, S., and Fink, G., "Towards semi-supervised transcription of handwritten historical weather reports," in [*Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*], 180–184, IEEE (2012).

[5] 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 [*IS&T/SPIE Electronic Imaging*], 86580J–86580J, International Society for Optics and Photonics (2013).

[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., "https://www.youtube.com/watch?v=eBQjHgejchA," (2011).