

Computer Assisted Handwritten Document Transcription Through
Subword Spotting

by
Brian L. Davis

A thesis proposal submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science
Brigham Young University

October 2015

ABSTRACT

Computer Assisted Handwritten Document Transcription Through Subword Spotting

Brian L. Davis

Department of Computer Science, BYU

Master of Science

Computer assisted transcription (CAT) for handwritten documents has emerged as a way to reduce man-hours spent transcribing documents by leveraging handwriting recognition technology in concert with humans' efforts. We propose that a highly modular and parallelizable CAT method can be created by leveraging subword spotting. Rather than attempting to recognize the text as a whole, we will spot common character n-grams in the document images. From this partial transcription information and user oversight the system will begin to infer transcriptions, using the relative placement of spotted n-grams to constrain matching with and on-line Lexicon. We will validate the proposed system with a series of user studies.

1 Introduction

At the current rate of FamilySearch’s volunteer indexing program, FamilySearch collects documents faster than they can transcribe them. While OCR technology has become accurate enough for tasks in printed document transcription, handwriting recognition has not reached the same level of performance. Accurately transcribing historical documents require a burdensome number of man-hours. This is a problem which faces genealogical record providers, archivists of general historical documents, and applies to a number of tasks related to offline handwriting transcription.

A fully automated, unconstrained handwriting recognition system with an acceptable level of accuracy on historical documents is not possible with the current level of technology. During a recent competition for handwriting recognition on historical documents, the top method had a word error rate above 25%[16]. Historical documents have noise, degradation, and other issues which cause accuracies to be lower than other documents. This has left most handwriting transcription work to humans. However, there are still ways to leverage recognition techniques to speed up the transcription process, while allowing human oversight and guidance to maintain accuracy. These we will refer to as computer assisted transcription (CAT) methods.

The transcription of handwritten historical documents is a service needed by organizations such as genealogical resource providers (FamilySearch, Ancestry, MyHeritage, etc.) and digital archives (the Internet Archive, libraries, etc.). Additionally, the general problem of offline handwriting transcription is highly demanded, evidenced by the numerous businesses offering this service. Those using these services expect human level transcription, but using a CAT method can cut the number of man-hours needed without losing the accuracy that comes from human oversight.

We have viewed this particularly from the perspective of FamilySearch’s volunteer indexing program. Having a better method of transcribing, which is more effective and pleasant to use, would draw more volunteers to the program while increasing the productivity of the volunteers. Our transcription paradigm reduces transcription to simple short tasks that could easily be completed by a casual user. It additionally makes it easier to bypass difficult parts of a document, allowing someone else to decipher it. Our paradigm also lends itself to a mobile environment, as it is very adaptable to a small screen and doesn’t require typing.

We propose a recognition technique which relies on the key idea that if we can recognize only parts of text, the missing information can be implied by use of a lexicon and language model. This can be compared to the well known game “Wheel of Fortune” where players are given a handful of letters, their placement in a phrase/word and a category to aid their guessing. See Fig. 1 for an example. Can you fill in the blanks? Is it easier if you know they are names?

In our method, a partial recognition (the known letters of the “game”) would be achieved by spotting frequently occurring character n-grams in handwritten documents. Character n-grams are simply groups of n consecutive characters, such as “ar” or “ing” (our work will be particularly using bigrams and possibly trigrams); all words can be viewed as being made of a set of character n-grams. Character n-grams are useful as they occur with far greater frequency than individual words, and are more distinguishable than individual letters. With this technique, a parallel CAT system could be constructed which allows a computer and multiple humans to work in concert to transcribe a corpus. This system particularly relies on very minimalistic human input (typically selecting things).

2 Related Work

Segmentation is a difficult part of handwriting recognition. While line segmentation techniques are able to accurately extract lines of text and with good success extract words, letter segmentation is

A n t h o _ _
_ _ c h a e _

Figure 1: These are two words with a handful of letters left out. It is not hard to guess the correct words, especially if told these are names. A computer, with access to a lexicon, can be far more effective at this guessing game than a human.

an open problem (in the 2013 ICDAR handwriting segmentation contest, multiple entries achieved close to 99% line segmentation detection and accuracy, the best word segmentation method achieved only 91% accuracy[?]).

Whole word recognition is infeasible for large vocabularies, so we would prefer to recognize individual characters. But to properly segment the letters from a handwritten word (particularly if it is written in a cursive script) is infeasible without having the transcription first. This is Sayre’s paradox, where the segmentation and recognition are dependent on one another.[22] To solve this, the state-of-the-art approaches have tried to recognize and segment simultaneously.

Hidden Markov models (HMM) were borrowed from speech recognition to achieve this task, and worked reasonably well[11]. However, HMMs have their own limitations, particularly the Markov assumption and the fact that they are most effective with windows less than the length of a letter, meaning context is often lost.

Recurrent neural networks (RNN) do not share the same limitations as HMMs, and have passed HMMs in performance[7]. However, they are still lacking the accuracy that is needed for a reliable automatic transcription system; the state-of-the-art results for RNNs (employing long short-term memory blocks) still may have above a 25% word error rate when transcribing a historical corpus[16].

Due to the limitations of current handwriting recognition methods, we believe that a fully automated handwriting transcription is out of reach with the present of technology. We thus turn our attention to semi-automated recognition approaches, where we are still using a human to transcribe, but also some intelligent recognition technology. A semi-automated solution will outperform a fully automated system in terms of accuracy and is still far more effective than completely manual transcription. Standard CAT approaches use human feedback to correct automatic transcription in an intelligent way, such that it learns something from the corrections. An active learning approach to handwriting transcription is similar to CAT, but is not concerned about achieving human level accuracy, just better accuracy, and thus requires less human involvement [17]. Since we wish to have a system with accuracy comparable to human performance, thus we will focus on CAT approaches.

Toselli, et al[19] have explored the realm of CAT using the idea of user-verified prefixes. They use a fairly standard HMM recognition model as the backbone of their approach, and take advantage of the incremental nature of the Viterbi decoding algorithm. The recognition is done for a line of text and the user corrects the first error. The Viterbi is then run again, but this time using the assumption that everything occurring before the correction is correct, and thus reusing the computation up to that point. They have also explored slight variations of the same approach that enable more fluent user input with a touchpad[21], mouse[14], or multimodal[20] which speed up the transcription process by allowing more intuitive user interaction (Fig. 2). Their approach relies on a language model to make corrections on a line when a supervision is made. This means the cannot be used to effectively transcribe documents containing non-sentence writing, such as tables and lists, with this method. Serrano, et al also have pursued a similar approach, where the

user corrects the n words in which the recognition model had the least confidence[18].

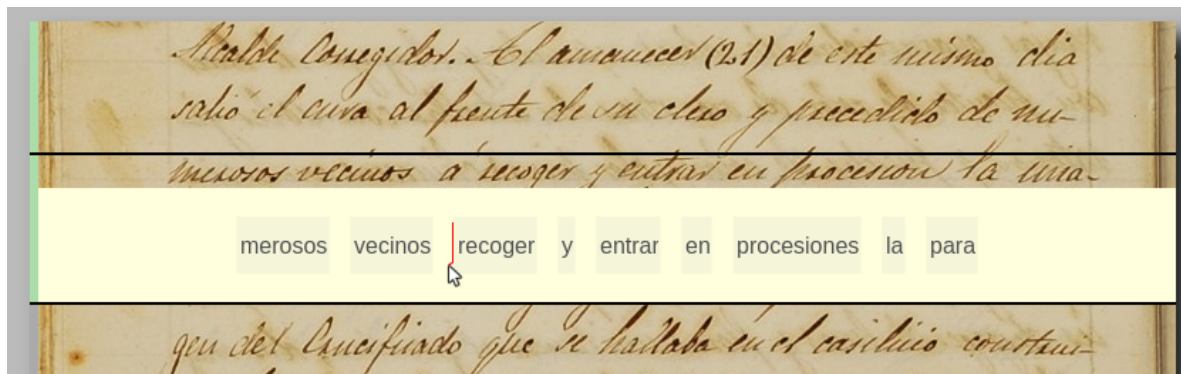


Figure 2: A screenshot of a demo of Toselli et al’s multimodal CAT system. The red line is drawn by the user to indicate the need to insert a word into the automatically obtained transcription.

While it is useful to employ an automatic handwriting recognition method as part of a CAT approach to reduce the human burden, we cannot use the same type of recognition used in most CAT systems, where there is a reliance on the predictive capacity of a language model. Some documents are not comprised of only sentences, and thus cannot be transcribed by these methods. However, many documents are structured such that a pattern can be learned to assist in transcription.

Robert Clawson[5]¹ designed a CAT system for tabular documents, which have a clear pattern. His approach relied on simply finding matches in the document column and assigning them the same user-specified label, as seen in Fig. 3. This provides an accurate CAT system where the user oversees all transcription. The user oversight of matches was accomplished by showing a list of matches to the user (with an adjustable threshold for sensitivity) from which the user removed the false-positive matches. This leveraged the human user’s natural ability to discriminate. However, the approach as a whole is limited in that it requires frequent word repetition to be effective. While some tabular documents have this, it is not true for most documents. Additionally, this method required a very computation preprocessing stage in which a similarity score of all words in a column with all other words in the column had to be computed.

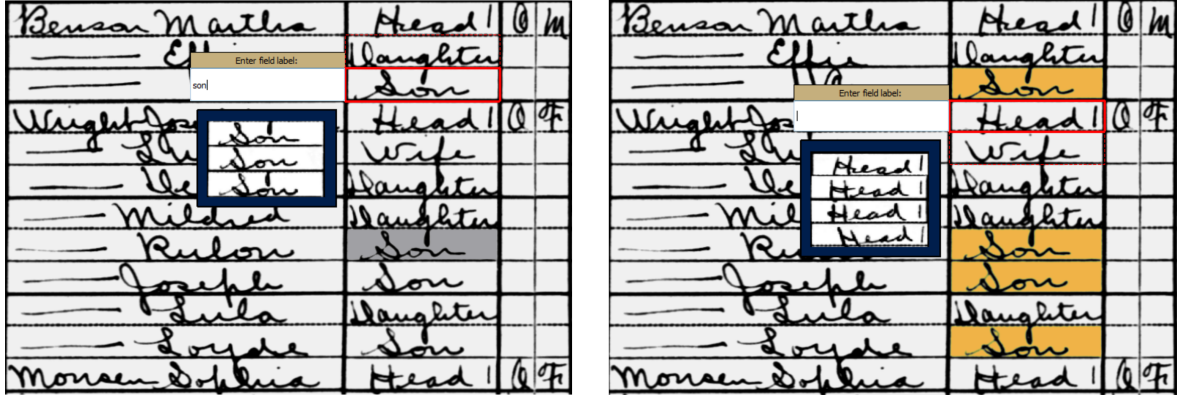
TODO: "A framework" also provides a similar method.

TODO: carpet approach and clustering

We will show that an alternative to word matching, or word spotting, is character n-gram spotting, as n-grams have more frequent repetition than words do. This will provide the backbone of our CAT system.

Word spotting was first proposed as an alternative to transcribing a corpus. Rather than digitizing the document text image so standard text searches can be run, the document is searched using the images themselves, either with a keyword string or a keyword image (exemplar image). There are two primary approaches to featurizing the images when word spotting: holistic features which capture information about a whole image (word) and local or sequential features[13]. Holistic features have one description for an entire word image (such as a bag-of-visual-words), whereas local features have descriptor for a small portion, or window, of a word image. As local features provide much more detail and are more flexible than holistic features, they have been the most actively pursued in research.

¹You can view a short demo and explanation of his approach at <https://www.youtube.com/watch?v=gqdVzEPnBEw>



(a) A small window shows the user matching words later in the column. The user can get rid of bad matches by either clicking on them or adjusting a threshold.

(b) The matched words are given the same label when entered.

Figure 3: Clawson’s CAT system for tabular documents.

Most of these locally directed feature approaches share the common theme of taking features from vertical slice windows (usually only one or a handful of pixels wide). They compare an exemplar using either dynamic time warping (DTW) or HMMs, relying on a prior line segmentation. The variance between the methods largely lies in the features used. Some use small square windows to allow totally segmentation-free word spotting using a sliding window[15].

Rodriguez et al[13] proposed a simple histogram gradient feature which demonstrated improved performance when compared with (1) the popular profile and transition based features of Rath and Manmantha[12], (2) the gradient and transition focused features of Marti and Bunke[11] and (3) the simple histogram features of Bunke et al[4]. They also showed that a HMM worked better than DTW, for their features. However, their method was not an exemplar based approach.

Recently, Aldavert et al[1] and Almazan et al[3] have presented superior word spotting methods which rely on heuristic descriptions. Aldavert et al use the well known bag-of-visual-words method, including recent improvements from the Computer Vision community. This is a very simple, yet effective, exemplar spotting approach. Aldavert et al use Fischer vectors, which are similar to bag-of-visual-words, with spatial pyramids in addition to a special character histogram pyramids. These are used to find a new space in training in which both strings and word images can be embedded. This allows it to preform both string and exemplar queries as well as hybrid queries which yield excellent results.

We are interested in spotting character n-grams (bi- and trigrams) rather than words. While this hasn’t explicitly been done, we feel examining the performance of other methods on short words (2 to 3 letters) is instructive. For Rothacker et al’s[15] their segmentation free HMM based method, they report ~46% and ~55% mean Average Precision (mAP) for two and three lettered words respectively, a drop by ~15% and ~6% from the mAP for all words they tested (61%). Fischer et al[6] report ~70% and ~83% mAP for two and three lettered words, compared to ~90% mAP for words of length 5 or longer, for their line segmentation dependent, character HMM based method. While neither of these numbers are very promising, Almazan et al[2] observe that sliding window approaches can frequently find false positives of short words inside other words (e.g. finding the word “the” inside the word “weather”). As this is precisely what we want to happen in n-gram spotting (this is, we *want* to find groups of letters in the middle of words), we

expect we should have more success spotting character n-grams than other methods in spotting short words. However, part of the poor accuracy in spotting short words is the fact that there is less information with which to discriminate; this is an obstacle we still must overcome.

3 Thesis Statement

We will construct a computer assisted transcription system for handwritten documents which will partially transcribe words by completing the easier task of spotting common character n-grams. These partial transcriptions permit a constrained lexicon look-up from which a user can select the correct transcription.

4 Project Description

We will develop a character n-gram spotter, experimenting with modifying existing methods as well as trying ones based on object detection strategies from computer vision. This performance will be empirically measured. Using the results of the performance of the n-gram spotting, we will evaluate the size of lexicon for which our approach will be feasible.

We then will implement a demonstrative CAT tool using the flexibility of n-gram spotting. This will largely be a proof-of-concept work to show how one might use character n-gram spotting. We anticipate its performance to be good, but it may be difficult to compare to similar tools as ours will be strongest in an application that is slightly different from how most others have been tested.

4.1 Spotting

We require a spotting method that which can learn incrementally. Image exemplar spotting approaches are an easy way of doing this as new examples can simply become new queries. However, other trained methods may be able to be updated. There are some spotting methods which are able to do both image and string based queries[3] which would be useful for leveraging both offline training and new exemplars.

Given our specific need in spotting character n-grams, we will likely have to adapt the current word spotting technology to get the performance we need. This will be a preliminary task that will provide the backbone for our CAT system.

4.2 Preprocessing

All documents will be have preprocessing done to reduce noise in the image, deskew the image, segment lines, have the text lines deslanted and have the words segmented. Word segmentation will likely have errors, but these can be corrected by users.

4.3 Lexicon

Lexicon look-ups with wildcards is a well studied problem. An approach which would be well suited to our needs is n-gram indexing, in which each n-gram has all words which contain it indexed to it. Then a search is simply an intersection of all the indexes for the n-grams in the query.

4.4 Tool

The system is designed to leverage the flexibility of character n-gram spotting to create partial transcriptions. This flexibility largely comes from it's parallelizable nature; individual spotting tasks, and transcription tasks, are not dependant on one another. This allows an interactive design with multiple asynchronous users. Additionally, we can exploit user feedback to improve spotting.

We will begin by training our spotting method so that it can spot n-grams in most types of handwriting with a reasonable accuracy. This will be used to begin spotting the documents. Depending on this character n-gram spotter's performance, a training step may need to precede the main execution, allowing the spotting method to gather context/author specific n-gram exemplars.

The system is broken into tasks, some of which are completed by the user, some of which are completed by a server doing the computation/spotting. By including a task manager which distributes tasks, we can allow for multiple users to be working on the same set of documents simultaneously. See Fig. 4 for a brief outline of the work-flow. The tasks comprising the work done by the computation server and the user will be the following subsections.

This method is unique in that the different user's progress will help produce more tasks, making the effort very cooperative. This happens as each completed n-gram verification task will provide partial transcriptions for words in the corpus. This allows more possible complete word transcriptions to be found. Each spotting and full word transcription that is done provides more n-gram image examples to the system from which it can improve spotting, creating more tasks for the server and then other users. This incremental learning from new n-gram example images can be done in two ways: using them as new image exemplars for queries directly or attempt to learn features from them to improve following queries. Additionally, we can use them as a refining tool to prune images returned by previous spotting tasks (that haven't been user-reviewed yet).

This prototype system will be developed as a in-browser app, thus making it immediately cross-compatible with all devices with a modern browser. A Node.js server will be created to communicate with the clients and the C++ program which will be responsible for performing the actual word spotting.

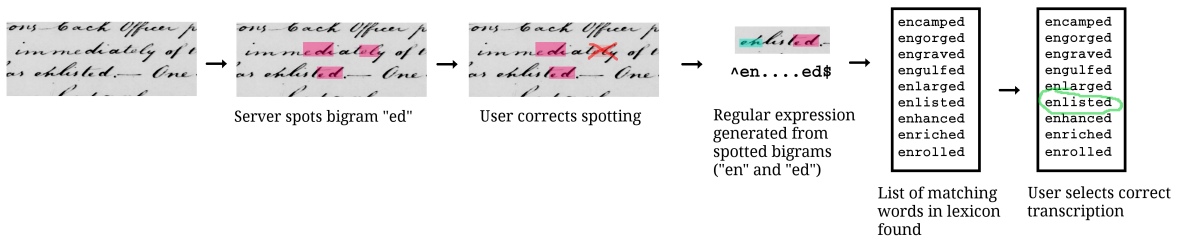


Figure 4: The work-flow of the proposed CAT system. We show a portion of a document image as an example.

4.4.1 Computation Server Tasks

Character n-gram spotting

Select the most promising n-gram exemplar to spot; this is based on the frequency of n-grams and whether we have new information for an n-gram from document/author specific examples extracted from transcribed text. Spot that n-gram in the documents. Pass results to user for n-gram verification task.

Lexicon look-up and scoring

When a word has roughly 50% of its area labelled by n-grams, we create a regular expression representing what we know. This is based on the position of the n-grams in the word, using flexible wildcards to specify possible unrecognized characters and their position (the “Wheel of Fortune” game). If the returned words are small (perhaps less than 40), we performing a scoring of them by running a word spotting algorithm on the word image searching for each of the possibilities. We can then discard any possible transcriptions scoring below a threshold. This threshold can be adaptive to the distribution of the particular set of scores. If few words are above the threshold (10 or less) we pass this to the user for the transcription selection task (where the user selects the correct the transcription from the returned words). If only one word is returned, either from the initial look-up or from the score pruning, we can transcribe the word without human intervention.

Incremental learning from transcription

After the user has confirmed a transcription for a word image, we can then extract new character n-gram exemplars from the image, using the spatial information we have from the n-grams spotted in the image. These can be author specific exemplars which can greatly improve spotting for pages with the same author. How the exemplar is used will depend on the final spotting algorithm selected; it may either be used directly as a query, to modify a query, or as means to learn new character models.

4.4.2 User Tasks

Character n-gram verification (Fig. 5)

The user is shown segments of the image where the spotted n-grams are highlighted. The user deselects/discards any false-positives. The assumption is that most should be correct, thus making it easy to pick out those not belonging to the set.

Select correct transcription (Fig. 6)

The word being transcribed highlighted and is presented to the user with the list of possible transcriptions (ordered by the likelihood of the possible transcriptions and the score generated from word spotting the image). The user either selects the correct transcription, or corrects the search by doing one or more of the following:

- Splitting the image into two words.
- Indicating the image is splicing a word (and which side).
- Indicating if any of the n-gram labels are wrong.
- Manually entering the transcription.

[.5cm]

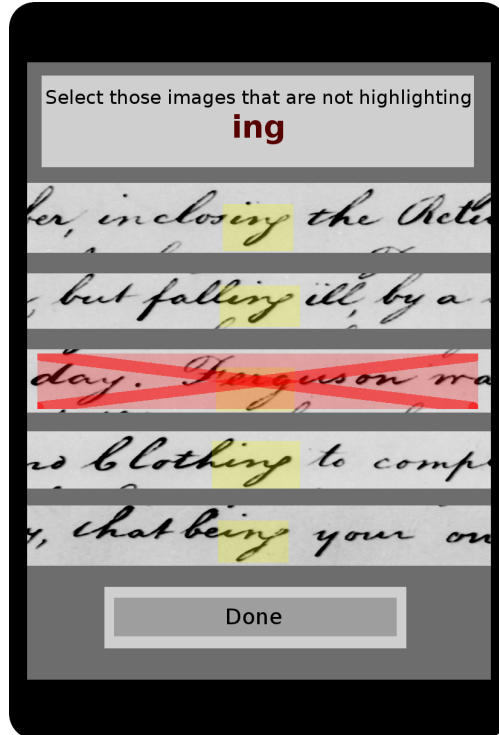


Figure 5: A mock-up of what the user might see when verifying character n-gram spotting. The highlighted images are from the server spotting a particular n-gram. The red-boxed image has been selected by the user as it is a false-positive.

4.5 An Example

To solidify our description of our transcription system, let us examine an example of how it might operate. Suppose we have a corpus of document images that contain a total of 4860 words (the George Washington dataset). We would begin by spotting the most common bigram, “th.” We might spot 30% of the instances of “th” in the document, 178 out of 593, with some number of false-positives. All of our spotting results are presented to users who discard the false-positives, thus leading to very high precision. Of the X instances of “th” we’ve spotted, 135 are at the beginning of the word. TODO

4.6 Preliminary Work

4.6.1 Subword Spotting

We have begun testing state-of-the-art word spotting methods under the application of spotting bigrams and trigrams within words. Our best results so far have been Almazan et al[3], which achieved 49% mAP spotting bigrams and 67% mAP spotting trigrams in the George Washington dataset.

4.6.2 System Simulation

In an attempt to justify our method, we created a simulation to find how many tasks would need to be completed by users if we were transcribing a given corpus. This simulation takes a number of parameters: a recall rate for a imaginary subword spotting method, a rough confidence level

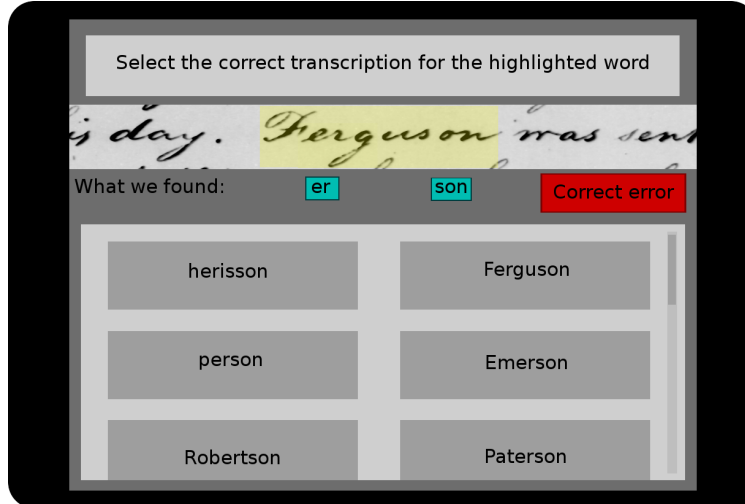


Figure 6: A mock-up of what the user might see when selecting a correct transcription. The transcription possibilities are returned from the server’s lexicon look-up.

for counting the number of unrecognized characters in a word, and a cut off for the number of possible transcriptions a word must be narrowed to before counting the word as transcribed (the real system would send a transcription selection task to the user, but we don’t feel the need to simulate this). We ignore the precision of our imaginary spotting method as all spotting is verified by the user, so given no user error, it will have 100% precision. However, the recall of the system will be balanced with how much work the user has to do in these tasks; this is important in our selection/development of a subword spotting method, but not to our simulation.

We have held the possible transcription cut off at 10, as we feel that is a reasonable number for a user to deal with. We used a recall rate of 49%, based on the mAP mentioned in the previous section (this should roughly correspond to a 50% accuracy rate), although it is likely a better subword spotting method will be used for our actual system. The results presented here are from a simulated transcription of the 20-page George Washington dataset. We used a 32,000 word lexicon and spotted only the 100 most frequent bigrams.

In Fig. 7 we can see the progress of the simulation through the series of tasks. As might be expected, the rate of transcription begins to drop off as most of the words are transcribed. It will be necessary to cut off transcription through n-gram completion at some point and do manual transcription for the remaining words. However, this demonstrates the feasibility of our method. Within 250 spottings, 75% of the corpus is transcribed (including out-of-vocabulary words).

A parameter our simulation is very sensitive to, which we assume the actual system will be sensitive to as well, is the confidence in predicting the number of characters in a word which are not spotted. If given a low confidence, such that the simulation searched more possible transcriptions, the simulation saturated transcription at around 66% transcribed at 350 iterations (spottings) and failed to move past 71% by 1000 iterations. If given perfect knowledge of how many characters are present, but not spotted, the system achieves 66% after only 127 iterations and does not saturate until about 86% has been transcribed. From further analysis, we see that it is caused by many words being transcribed as much as they can be (given the limited n-gram set), but still not returning a possible transcription count below the threshold of 10. If the prediction of the number of unspotted characters proves to be difficult, a loosening of this threshold as the corpus is transcribed may be required.

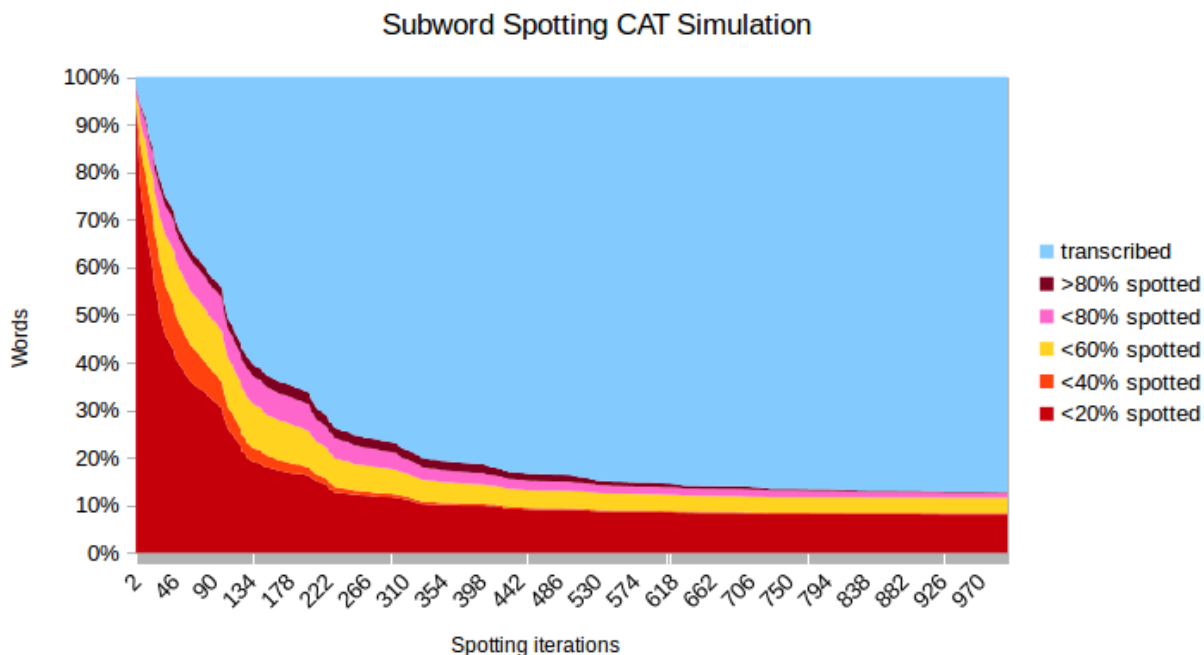


Figure 7: A simulation run with medium confidence on unspotted character count prediction. The chart is drawn so one can observe the progress of spotting as well as transcription, each category indicating the percentage of words which have the given portion of their characters recognized by spottings (or indicating the percentage of words transcribed). One will note that a large number of words never have more than 20% of their characters spotted. This is due to two reasons: we are only using the 100 most frequent bigrams, and a number of words in the corpus are numbers, which we are not spotting or transcribing in this simulation. The group of words who are between more that 40% spotted that are never transcribed are either out-of-vocabulary as well or are simply diabolic cases which have too many possible transcriptions given the limited number of bigrams we can spot.

5 Validation

The n-gram spotting methods will be empirically tested using well known datasets (George Washington[9] and IAM[10]) as well as a dataset closer to the domain of our expected application of the transcription method.

5.1 User Study

Our CAT system must be validated with a user study as it’s goal is to be a more effective crowd-sourcing method for transcription. It is not purely intended to be faster (indeed, it may not be faster than other computer assisted transcription methods), but more modular, allowing for work to be broken up into smaller pieces, and more enjoyable to use. It clearly achieves modularity in design, but we want to see how fast it is and get user feedback to see if it is something people would enjoy using.

Most other computer assisted transcription methods rely on a more fully automated transcription upfront and report performance measures in terms of word error rate or effort reduction,

which are measures of how many words the user had to correct. While our users will primarily be doing corrections as well, they will very different types of corrections (taps and swipes instead of typing). Thus it would be difficult to compare these other methods to our proposed method using the previously authors’ metrics, though they still may prove to be useful.

We propose doing a user study which would involve having a volunteer² use the system for about 5-10 minutes on their own device at their own convenience. This is probably the easiest way to get a large number of participants and a varying degree of experience transcribing documents among the volunteers. To encourage use on more platforms, so we get a broader survey, the study will encourage users to use tablets and phones. We will track what device they use, how many tasks they complete, how long it takes them to complete tasks, and errors they make, in addition to having the volunteers complete a short survey about their experience.

Given the flexibility of the system, we would like to test it on two different document types: a regular written text like the George Washington[9] or the Jennie Leavitt Smith[8] dataset, and a structured dataset such as census or marriage records. This will provide us with some comparison to Toselli’s CAT approaches (which work on corpora similar to the George Washington dataset) and Clawson’s CAT (which works on tabular documents).

It will be difficult to get a synchronized group of testers to explore the parallel nature of our method. Instead, we will complete a number of spotting verification tasks, and possibly transcription selection tasks, prior to having the users begin. These can then be fed to the system during a user test to simulate having other users on the system, although it may suffice to simply feed them the the system prior to the user interacting.

6 Thesis Schedule

Finalization of spotting method	1 February 2016
Complete implementation of UI/backend	1 April 2016
Complete user tests	15 April 2016
Complete revisions	1 May 2016
Complete additional user tests	15 May 2016
Thesis written (all but final revisions)	7 June 2016
Thesis defended	1 July 2016

References

- [1] David Aldavert, Marçal Rusiñol, Ricardo Toledo, and Josep Lladós. A study of bag-of-visual-words representations for handwritten keyword spotting. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(3):223–234, 2015. ISSN 1433-2833. doi: 10.1007/s10032-015-0245-z. URL <http://dx.doi.org/10.1007/s10032-015-0245-z>.
- [2] Jon Almazán, Diego Fernández, Alicia Fornés, Josep Lladós, and Ernest Valveny. A coarse-to-fine approach for handwritten word spotting in large scale historical documents collection. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*. IEEE, 2012.

²Recruiting volunteers can be aided by using Robert Clawson’s volunteer testers list as a starting place. The rest will be gathered via on-campus flyers and personal connections.

- [3] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. Word spotting and recognition with embedded attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2552–2566, 2014.
- [4] Horst Bunke, Samy Bengio, and Alessandro Vinciarelli. Offline recognition of unconstrained handwritten texts using hmms and statistical language models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):709–720, 2004.
- [5] Robert Clawson. Intelligent indexing: A semi-automated, trainable system for field labeling. Master’s thesis, Brigham Young University, 2014. URL scholarsarchive.byu.edu/etd/5307/.
- [6] Andreas Fischer, Andreas Keller, Volkmar Frinken, and Horst Bunke. Lexicon-free handwritten word spotting using character HMMs. *Pattern Recognition Letters*, 33(7):934–942, 2012.
- [7] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2009.
- [8] Douglas J. Kennard, William A. Barrett, and Thomas W. Sederberg. Word warping for offline handwriting recognition. In *Proceedings of ICDAR*, pages 1349–1353. IEEE, 2011.
- [9] V. Lavrenko, T. M. Rath, and R. Manmatha. Holistic word recognition for handwritten historical documents. In *Proc. DIAL 2004*, pages 278–287. IEEE, 2004.
- [10] U. Marti and H. Bunke. The IAM-database: An english sentence database for off-line handwriting recognition. *Int. Journal on Document Analysis and Recognition*, 5:39–46, 2002.
- [11] U.V. Marti and Horst Bunke. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *International journal of Pattern Recognition and Artificial intelligence*, 14(1):65–90, 2001.
- [12] Toni M. Rath and Raghavan Manmatha. Features for word spotting in historical manuscripts. In *Proceedings Document Analysis and Recognition*, 2003.
- [13] José A. Rodriguez. and Florent Perronnin. Local gradient histogram features for word spotting in unconstrained handwritten documents. In *Int. Conf. on Frontiers in Handwriting Recognition*, 2008.
- [14] V. Romero, A.H. Toselli, L. Rodriguez, and E. Vidal. Using mouse feedback in computer assisted transcription of handwritten text images. In *Proceedings of ICDAR*, pages 96–100. IEEE, 2009.
- [15] Leonard Rothacker, Marçal Rusiñol, and Gernot A. Fink. Bag-of-features hmms for segmentation-free word spotting in handwritten documents. In *Proc. ICDAR 2013*, pages 1305–1309. IEEE, 2013.

- [16] Joan Andreu Sánchez, Alejandro H. Toselli, Verónica Romero, and Enrique Vidal. Icdar 2015 competition htrts: Handwritten text recognition on the transcriptorium dataset. In *Proceedings of ICDAR*, 2015. URL icdar.org/proceedings.
- [17] Nicolás Serrano, Adrià Giménez, Albert Sanchis, and Alfons Juan. Active learning strategies for handwritten text transcription. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI '10*, pages 48:1–48:4, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0414-6. doi: 10.1145/1891903.1891962. URL <http://doi.acm.org/10.1145/1891903.1891962>.
- [18] Nicolás Serrano, Adrià Giménez, Jorge Civera, Alberto Sanchis, and Alfons Juan. Interactive handwriting recognition with limited user effort. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(1):47–59, 2014. ISSN 1433-2833. doi: 10.1007/s10032-013-0204-5. URL <http://dx.doi.org/10.1007/s10032-013-0204-5>.
- [19] A.H. Toselli, V. Romero, L. Rodriguez, and E. Vidal. Computer assisted transcription of handwritten text images. In *Proceedings of ICDAR*, pages 944–948. IEEE, 2007.
- [20] Alejandro Toselli, Verónica Romero, Moisés Pastor, , and Enrique Vidal. Multimodal interactive transcription of text images. *Pattern Recognition*, 43(5):1814–1825, 2010.
- [21] AlejandroH. Toselli, Verónica Romero, and Enrique Vidal. Computer assisted transcription of text images and multimodal interaction. In Andrei Popescu-Belis and Rainer Stiefelhagen, editors, *Machine Learning for Multimodal Interaction*, volume 5237 of *Lecture Notes in Computer Science*, pages 296–308. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85852-2. doi: 10.1007/978-3-540-85853-9_27. URL http://dx.doi.org/10.1007/978-3-540-85853-9_27.
- [22] Alessandro Vinciarelli. *Offline Cursive Handwriting: From Word To Text Recognition*. PhD thesis, IDIAP, 2003. URL infoscience.epfl.ch/record/82879.

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis proposal submitted by

Brian L. Davis

This thesis proposal has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date

William A. Barrett, Chair

Date

Thomas W. Sederberg

Date

Daniel Zappala

Date

Quinn Snell
Graduate Coordinator