Subword Spotting for Use in a Computer Assisted Transcription System

Brian Davis, Robert Clawson and William Barrett
Department of Computer Science, Brigham Young University, Provo, Utah
Email: briandavis@byu.net

***Abstract*—Recently, computer assisted transcription (CAT) systems for handwritten documents have been proposed that use word spotting to speed up a human transcriber's work. They are, however, dependant on frequent repetition of words in documents to be effective. We propose that character n-grams could be used in place of words to construct a CAT system as n-grams occur much more frequently. We demonstrate some preliminary results in spotting subword character bigrams and trigrams that are adequate for a feasible CAT system.**

I. Introduction

With current technology, a fully automated, unconstrained handwriting recognition system on historical documents for many applications' required level of accuracy is not possible. During a recent competition for handwriting recognition on historical documents, the top method had a word error rate above 25% [**?**]. 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 are computer assisted transcription (CAT) methods, in which the computer and human user's efforts are coupled.

Robert Clawson designed Intelligent Indexing [**?**][1], a CAT system for tabular documents. Intelligent Indexing relies on finding matching word images in a document column and assigning them the same user-specified label, as seen in Fig. 1. 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 removes the false-positive matches. This leverages the human user's natural ability to discriminate. Zagoris et al. [**?**][2] also proposed a CAT system using word spotting. Rather than focusing on having a user remove bad spots, as the user confirms word images are correct spottings, a relevance feedback loop helps select better results from the word spotting. However, both of these approaches are limited as they require frequent word repetition to be effective.

II. Using Character N-Grams

While many words do not repeat with a high frequency in most documents, character n-grams do. In particular, we focus on bigrams and trigrams. Let us examine the George Washington (GW) dataset [**?**] as an example. The most common bigram in the English language ("th") occurs 622 times, and the 50th most common bigram ("ur") occurs 131 times in this dataset. This is compared to the most common word in the English language ("the") occurring 82 times, and the 50th most common word ("us") occurring 9 times. Additionally, there are many words (such as "river") that occur only once in the dataset, but are composed of character n-grams. How can character n-grams help us transcribe? Given enough n-grams spotted in the same word, the possible transcriptions of that word are quickly reduced by constructing a simple regular expression with which to query the lexicon. To verify that this is feasible, we apply this idea to the GW dataset.

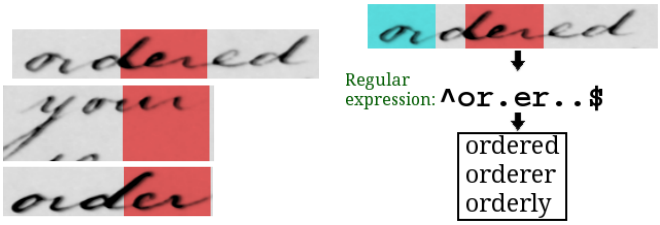
If the 100 most frequent bigrams in the English language are spotted in the GW dataset with 50% recall, 54% of the words in the corpus can be narrowed down to 10 or fewer possible transcriptions, assuming (a) we can correctly guess the correct number of characters we haven't spotted in the words, (b) that user oversight gives us 100% precision, and (c) using a 115,000 word lexicon. From this list of 10 or fewer words a user can easily select the correct transcription. Fig. 2a shows some examples the bigram "er" being spotted in a few words. Fig. 2b shows how the short list of words a user selects from is generated from the n-grams spotted in a word. By learning from correct spottings and transcriptions,

(a) A small window shows matching words from the column. The user can get rid of bad matches (e.g. "Wife") by clicking on them.

(b) The matched words are given the same label, indicated by the highlighting. and the red box is advanced to the next word to be transcribed.

Fig. 1: Clawson's CAT system, Intelligent Indexing.

[1] See http://tiny.cc/intelind for a short demo.
[2] See http://vc.ee.duth.gr/ws/ for an interactive demo.



(a) (b)

Fig. 2: (a) shows results of spotting the bigram "er" with [**?**]. The false positive ("ur" in "your") could be removed with user oversight. (b) shows that from the bigrams "or" and "er," a regular expression can be constructed that returns only three possible transcriptions from a 115,000 word lexicon.

new spotting queries will be able to be created; character n-grams that we missed initially can be spotted with subsequent queries. If we assume they will achieve a 50% recall as well (on the remaining bigrams), by spotting the 100 bigrams again 74% of the corpus will be narrowed down to 10 or fewer possible transcriptions

III. Character N-Gram Spotting Experiment

Spotting subword character n-grams is a largely unexplored area. We have run some initial tests using some word spotting techniques in the application of subword spotting to demonstrate that this is feasible. We evaluated two spotting methods: a part-structured inkball method by Howe [**?**] and an attribute embedding method presented by Almazán et al. [**?**]. Because these are whole word spotting methods, they are not suited to perform subword spotting without some modification. We modified the code provided by Almazán[3] and Howe[4].

We evaluated the subword spotting performance of these methods on two datasets: the George Washington dataset [**?**] and the IAM off-line dataset [**?**]. We evaluated spotting with 10 query images of each of the 20 most frequent bigrams in the English language and with 5 query images of each of the 10 most frequent trigrams. Query images were manually cropped from word images of the datasets. Both methods were tested using the first fold of the testing partitions of the datasets as Almazán et al. (available with the provided code). The GW testing set contains 1215 word images, and the IAM testing set contains 13752 word images. Stop words were not ignored in any way. The attribute embedding method was not trained on character n-gram images but on word images.

A follow-up evaluation was done using the method by Almazán et al. to spot by string the 100 most frequent bigrams using both the GW and IAM datsets.

IV. Results and Conclusion

Tables I and II show the results from the experiments. The mean-average-precision achieved by Almazán et al.'s method is acceptable for a CAT system as a user can discard incorrect

[3]http://github.com/almazan/watts

[4]http://cs.smith.edu/~nhowe/research/code/index.html#psm

TABLE I: Mean-average-precision on 20 most frequent bigrams and 10 most frequent trigrams

	Bigrams		Trigrams	
Method	GW	IAM	GW	IAM
Howe (query by image)	22.11%	9.36%	28.13%	6.25%
Almazán et al. query by image	40.67%	23.10%	69.79%	42.91%
Almazán et al. query by string	64.20%	32.99%	65.05%	48.87%
Almazán et al. hybrid query	64.32%	33.04%	72.38%	49.81%

TABLE II: Subword spotting results using Almazán et al. to query by string on the 100 most frequent bigrams.

Dataset:	GW	IAM
mAP:	45.27%	20.65%

spottings. The difference between the results of spotting the top 20 versus top 100 bigrams is likely because there were fewer instances of the less common bigrams in the training set. For the GW dataset there are on average 98 training instances for each of the top 20 most frequent bigrams and only 33 training instances for the next 80 bigrams; in query by string, the spotting is wholly dependent on these. We are confident better subword spotting accuracy will be able to be achieved with further work.

Clawson and Zagoris et al. showed that word spotting can create an effective CAT system for documents with high word repetition. We have presented an argument and preliminary results indicating that an effective CAT system can be constructed using character n-gram spotting. A user's oversight will allow precise n-gram spotting results. As more n-grams are spotted in a particular word, its list of possible transcriptions is constrained so that a user can easily select the correct one. As spotting results are harvested to create more queries, more n-grams will be spotted, and more of the document will be transcribed.

References