Indexing On-Line Handwritten Texts Using Word Confusion Networks

Sebastián Peña Saldarriaga, Mohamed Cheriet
Synchromedia - École de technologie supérieure
1100, rue Notre-Dame Ouest
Montréal (Québec) H3C 1K3, Canada
spena@synchromedia.ca, mohamed.cheriet@etsmtl.ca

***Abstract*—In the context of handwriting recognition, word confusion networks (WCN) are convenient representations of alternative recognition candidates. They provide alignment for mutually exclusive words along with the posterior probability of each word. In this paper, we present a method for indexing on-line handwriting based on WCN. The proposed method exploits the information provided by WCN in order to enhance relevant keyword extraction. In addition, querying the index for a given keyword has worst case complexity $O(\log n)$, as compared to usual keyword spotting algorithms which run in $O(n)$. Experiments show promising results in keyword retrieval effectiveness by using WCN when compared to keyword search over 1-best recognition results.**

***Keywords*-document retrieval; on-line handwriting; word confusion networks; keyword spotting;**

I. Introduction

Documents which are captured at the time of writing, and encode the dynamics of handwriting, are referred to as on-line documents or digital ink. With the recent advances in pen computers and digital pens, such documents are gaining popularity. As a result, reliable methods allowing to retrieve quickly and accurately such documents are increasingly demanded.

Our proposal aims at providing a robust and efficient data structure, i.e. an index, to support more sophisticated text search engines over digital ink. We make the distinction between *data retrieval* which consists in providing facts like word frequencies, and *information retrieval* which consists in processing such facts in order to meet a specific information requirement [1].

Typically, our index structure can be used to support state-of-the-art information retrieval approaches. However, it can also be used as a keyword spotting engine where the word or phrase queries are provided in some text encoding, e.g. UTF-8.

In this paper, we propose a novel framework for handwriting indexing relying on word confusion networks built from the output of a recognition engine. The remaining of this paper is organized as follows: Section II gives an overview of existing handwriting retrieval methods. Then we present our indexing approach in Section III. Section IV outlines the corpus and queries used in our experiments. Experimental results are presented and discussed in Section V, then conclusions are drawn in the final section.

II. Related work

Searching unconstrained handwriting is a challenging task that has been subject to research for years now. Several approaches have been proposed in the past, in the offline [2], [3], [4], [5], [6] as well as in the on-line domain [7], [8], [9], [10], [11], [12]. In general, existing methods for handwritten document retrieval can be divided into recognition-free and recognition-based approaches.

Usually, recognition-free approaches match words using low level features, are language-independent but writer-dependent. For instance, in [2] dynamic time warping is used to match pixel-based features of word images. In on-line handwriting, ink queries can be matched either at the stroke [7] or at the point level [9], [10].

On the other hand, retrieval can be performed over noisy texts as output by a handwriting recognition engine [8], [4], [6], [11]. As far as the underlying recognition engine is writer independent, these methods work in a writer-independent fashion. However, noisy text output is the major pitfall of these approaches. It has been argued that recognition errors can have little impact on search experiences [4] as far as the documents are not too short and that redundancy can cope with recognition errors. Attempts have been made though to counter the impacts of recognition errors on retrieval, including techniques based on approximate string matching [7], n-best recognition candidates [8], [6], posterior lattices and Viterbi search [11], and combination of word spotting and noisy text retrieval [12].

Our contribution clearly makes use of a handwriting recognition engine. However, it differs from the above mentioned works in several aspects. Firstly, because these works provide all-in-one solutions to handwriting retrieval while our aim is to provide robust data structures to store the document database underlying a retrieval system. Secondly, because we give theoretical guarantees in time complexity in order to ensure scalability. It is worth to note that most keyword searching algorithms run in $O(n)$ time, where n can be either the number of documents [11] or the vocabulary size [9]. Moreover, the matching process involves computationally expensive dynamic programming algorithms. For such reasons, these approaches will hardly scale to large or even moderately volumes of data.

III. Indexing handwriting via confusion networks

In this section we describe our approach to indexing. First we begin by introducing inverted files in Subsection III-A, then we describe word confusion networks (WCN) and an algorithm to create them in Subsection III-B. The remaining subsections will consider the index construction and querying, and running time analysis.

A. Inverted files for handwritten documents

Like in database management systems, fast resolution of queries requires the use of an *index*, that is to say, a data structure that maps words to the documents that contain them. Inverted files are considered to be the most efficient index structures for text query evaluation [13]. Figure 1 shows a schematic of view of an inverted index.

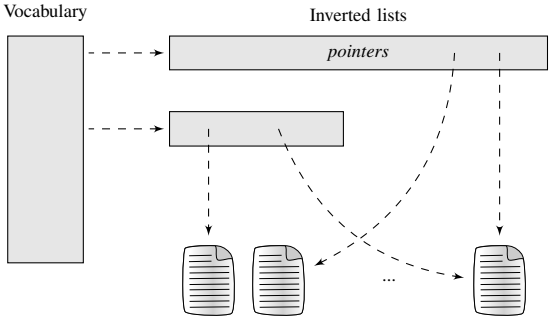

Figure 1. Schematic view of an inverted index.

Our index should provide the usual simple statistics that underly similarity measures in search engines: *a*) the frequency of term t in a document; *b*) the number of documents containing t; *c*) the number of occurrences of t in the collection; *d*) the number of documents in the collection; and *e*) the number of terms in the collection.

Besides this information, each occurrence of a term is indexed along with its posterior probability, the stroke offset (i.e., the number of the first stroke in the word) and the length of the term in strokes.

B. Word Confusion Networks

Word confusion networks [14] are compact representations of candidate word lattices. WCNs provide an alignment for each word in the lattice, and at each alignment position a set of mutually exclusive word hypothesis called a confusion set. Each word in a confusion set is associated with its posterior probability, a stroke offset and length. Figure 2 shows a sample confusion network and the handwritten sentence corresponding to it. Documents' transcriptions can be expanded with alternative recognition candidates that may have been written but were not in the top choice, while the use of posteriors avoid overestimation of spurious alternatives.

For instance, in Figure 2 the top recognition candidate does not match the expected result. However, dashed paths corresponding to subsequent recognition hypothesis allow to find the correct recognition result.

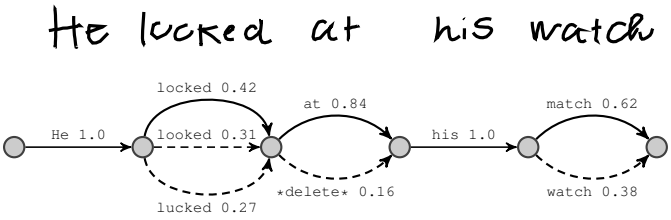

Figure 2. The handwritten sentence `He looked at his watch` and its corresponding confusion network. Solid arcs represent the top recognition result provided by the recognizer, while dashed ones represent alternative recognition candidates.

The construction of a confusion network follows three main steps:

1) Compute the posterior probabilities for all candidates in the word lattice.
2) Set the 1-best path as the pivot path of the network.
3) Align the remaining paths with the pivot, merging the transitions that correspond to the same word and occur at the same stroke offset by summing their posterior probabilities.

The last step implies aligning N sequences of arbitrary length which is known to be an NP-complete problem [15]. By using the 1-best path as the pivot of the network, the problem can be reduced to aligning N times two sequences. The asymptotic computational complexity is then $O(N \times m^2)$, where m is the average length of the sequences, assuming that for a given handwritten text, alternative recognition candidates have similar lengths.

C. Index construction and querying

We use a simple in-memory algorithm to build our inverted index. Traversal of each WCN is performed and edges of each confusion set are added to the vocabulary (except edges accounting for word deletions), updating inverted lists as necessary. Finally, we iterate through the in-memory index that has been constructed, and store the inverted lists and frequency data in separate files. A vocabulary file is created then, containing for each term t, pointers providing random access to the former files.

At querying time, the vocabulary file is entirely loaded on memory, and inverted lists only accessed on demand. A cache policy is defined in order to avoid recurrent disk-access for frequently requested terms.

D. Complexity and running time analysis

The index vocabulary is stored in a dynamic sorted array, in which insertion time is $O(\log n)$, unless the array is to

be resized, in which case insertion is $O(n)$. Assuming each term appears one time in the collection, for n terms to be indexed, construction time is $O(n \log n)$ in the number of terms and linear in the number of confusion networks. Obviously, indexing is a time consuming process, but we didn't focus in optimising the index construction process, since it is not supposed to happen on-the-fly and it can be easily parallelized.

Accessing a term in the index is done in $O(\log n)$, which is the time complexity of a standard binary search. The index also supports "starts with" queries in order to mimmick the behavior of stemming algorithms. This operation is performed in $O(\log n + k)$ time, where k is the number of terms that starts with the query. For phrase queries, the processing time is slower since inverted lists of each term are fetched and intersected. The cost of retrieving a phrase query is dominated by the cost of accessing inverted lists for common words like "in" or "the". However, we cannot ignore them since they play important semantic roles in phrase query constructions [13].

IV. Data

Experiments were conducted on the IAM On-Line Handwriting Database (IAM-OnDB) [16]. The IAM-OnDB database contains 1 560 forms of on-line handwritten English text acquired on a whiteboard. This database has been extensively used as a benchmark for handwritten text recognizers and writer identification methods, this to best of our knowledge the first time it is used in indexing and retrieval experiments.

The set of keywords used in our experiments were obtained by taking nouns that appear the most in the ground-truth data. Noun phrases were also extracted from the ground-truth data with a terminology extraction tool [17]. The idea is to extract corpus-specific vocabulary based on statistical as well as linguistic term properties. A set of 50 terms, including several noun phrases, were extracted to be used as queries. Table I provides the set of queries along with their number of occurrences in the corpus.

V. Experiments

Microsoft Tablet PC SDK 1.7 [18] was used to recognize documents from the IAM-On database. Our goal is to show that our approach can be used along with a "black box" recognizer which only provides recognition candidates. Since posterior probabilities are not provided by the SDK, we approximated a confidence value for each word w as follows [19]:

$$P(w) = \frac{2}{N(N+1)} \sum_{k=1}^{N} \delta(w, w_k) \times (N + 1 - k) \qquad (1)$$

where k is the rank of a given word hypothesis, N is the size of the N-best list and $\delta(w, w_k) = 1$ if $w = w_k$ or 0 otherwise.

Table I
Set of queries used in our experiments.

Term	Count				
work	44	film	19	general	11
book	38	week	18	hair	11
life	38	british	17	picture	11
house	36	war	17	german	10
home	35	light	16	autumn	9
world	35	food	15	bed	9
room	31	speech	15	research	9
young	30	state	15	doctor	8
mother	28	earth	13	operation	6
year	26	hour	13	true world	6
days	24	method	13	night club	5
table	24	body	12	wood	5
boy	23	child	12	lead chromate	4
government	23	family	12	prayer book	4
church	21	order	12	proof texts	3
figure	20	policy	12	sodium circuit	3
party	20	situation	12		

A. Indexing time

Our experiments were run on a 1.8 Ghz AMD Athlon™ processor with 2 GB RAM, which represents a standard configuration nowadays. Table II shows indexing time for several runs of our method. It can be seen from the table, that running time is overwhelmingly dominated by the cost of building the confusion networks, except for the trivial 1-best case. For 5-best hypothesis building the word confusion networks accounts for 53% of the indexing time, while at 30-best candidates it accounts for 88% of the running time, which is not surprising given the complexity of the alignment algorithm (see Subsection III-B).

It is worth noting that indexing the IAM-OnDB collection in five hours means that each document is processed in 11 seconds. This seems to be a fair processing time for incremental indexing, i.e. one document at time. For large volumes of data, the algorithm can be parallelized or the WCN construction approximated if large values of N are to be used. If a handwriting recognition engine can output confusion networks directly, the cost of indexing is nearly non-existent.

Table II
Processing time as function of the N-best list size.

	N-best list size				
	1	5	10	20	30
Recognition	0:35:02	0:34:42	0:34:33	0:34:38	0:34:42
WCN building	0:00:22	0:44:49	1:25:18	2:54:44	4:26:28
Indexing	0:00:01	0:00:01	0:00:02	0:00:01	0:00:01
Total	0:35:25	1:19:32	1:59:55	3:29:23	5:01:11

B. Index size

Table III gives some statistics showing the storage overhead induced by our indexing scheme. The collection size

is 390 MB in its original format and 60.9 MB compressed. The size of the final indices is very small compared to the size of the input data. The average storage overhead is 2% of the compressed data. The small size of the indices is achievied by using an efficient index representation based on variable-length encoded integers. For larger datasets, further compression techniques could be used in order to reduce space consumption and disk accesses.

Table III
INDEX SIZE AS FUNCTION OF THE N-BEST LIST SIZE.

	N-best list size				
	1	5	10	20	30
Term count	11483	14085	15624	17046	17467
Size (in MB)	0.92	1.10	1.18	1.25	1.27
Overhead	1.49%	1.80%	1.94%	2.05%	2.08%

It is worth noting that the number of distinct terms rises sharply between 1 and 20-best candidates; considering more and more candidates, little increases the number of new terms in the index. This can be a limitation of the lexicon or the language model underlying the recognizer. In such configurations it is not worth taking much candidates since the recognizer will not be able to produce distinct word hypotheses, and the remaining hypotheses at the text level will only be permutations of the first word hypothesis.

C. Retrieval time

Average retrieval time for the whole set of queries is 19 milliseconds, including index loading and disk access. The computational overhead for retrieving simple phrases queries is non-existing, however, this is to be related to the size of the collection and the index vocabulary.

D. Accuracy

We present the performance of our indexing approach using standard information retrieval evaluation metrics: precision, recall and F-measure. Precision (P) is the percentage of keywords retrieved that correspond to a real keyword. Recall (R) is the fraction of all true keywords that are returned by the search. Finally, the F-measure is the harmonic mean of precision and recall:

$$F = 2 \times \frac{P \times R}{P + R} \tag{2}$$

The object of the experiments is to examine the effectiveness of expanding the top recognition result with additional word hypothesis. Table IV shows performances of keyword spotting over our index. The word error rate (WER) obtained with Microsoft's handwriting recognition engine is 28.68%. The following table provides recall, precision, and F-measure for the different N-best considered.

It can be seen from Table IV that using word confusion networks improves recall and F-measure consistently. As expectedly, precision decreases because the risk of considering bad hypothesis is higher when using several recognition candidates.

Table IV
PRECISION, RECALL AND F-MEASURE AS A FUNCTION OF THE N-BEST LIST SIZE. RESULTS CONSIDERING ONLY SINGLE WORD QUERIES ARE GIVEN IN PARENTHESES.

N-best	Recall	Precision	F-measure
1	83.23% (84.42%)	91.73% (91.61%)	0.8727 (0.8786)
5	90.97% (92.39%)	88.48% (88.33%)	0.8971 (0.9031)
10	91.44% (92.87%)	88.13% (87.98%)	0.8975 (0.9036)
20	91.79% (93.23%)	86.61% (86.45%)	0.8912 (0.8971)
30	91.91% (93.35%)	86.34% (86.17%)	0.8904 (0.8962)

By considering the top 5 hypothesis, recall is already improved by more than 7%, at the expense of 3% of precision accounting for the insertion of spurious terms. In our experiments, the optimal tradeoff between precision and recall is obtained by indexing with 10-best hypothesis, which corresponds to the higher F-measure.

We also provided the scores for single word queries alone since it is the usual way of measuring word spotting performances. In this configuration the system behaves in the same way as before, but performances are a little higher. Actually, the retrieval of noun phrases was not improved by indexing with hypothesis below the top recognition result.

Word confusion networks built from bigger sets of recognition candidates offer more potential for expanding the index with competing terms, which results in improving recall as seen in Table IV. However, the risk of including spurious terms is increased along with the computational cost. It is worth noting, however, that as far as an information retrieval system built on top of our index can exploit posterior probabilities, the insertions of false occurrences of terms can be downweighted enough so that retrieval results are not degraded.

VI. CONCLUSION

In this paper, we proposed a scalable approach to index and search keywords in on-line handwritten documents using word confusion networks. Confusion networks offer a convenient representation of competing word recognition candidates. We stored the posterior probability of each competing term, along with segmentation data, i.e. stroke offsets and lengths, in our index.

Keyword search recall is improved in a significant manner by indexing data with more than one recognition candidate. On the other hand, the precision is decreased as expected since the risk of introducing false occurrences of words is higher. In our experiments building confusion networks from top 10 recognition candidates offered the best tradeoff between precision and recall.

Indexing experiments were carried out with different numbers of recognition candidates. They showed that indexing

time is overwhelmingly dominated by the cost of building the confusion networks. We believe that faster algorithms for bulding confusion networks from word lattices will increase scalability of our indexing scheme.

Further experimental validation is still needed. It includes verifying the results with different recognition systems, in order to achieve different levels of word recognition error rate, and using different databases, including databases in other languages than English.

Future research will involve also, investigating more deeply indexing and retrieval of noun phrases, and the problem of handling out of vocabulary queries, which has not been addressed in the present paper. We are currently considering several approaches, including: subword level indexing [20], and noise handling at query time by expanding the query with "erroneous" words that are likely to be recognized [21]. We believe that further improvements could be obtained by combining confusion network indexing and query expansion techniques.

ACKNOWLEDGMENT

This work was supported by the MDEIE of Québec under the program PSR-SIIRI via the ICIOS project grant number PSR-SIIRI-148.

REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: Addison Wesley-ACM Press, 1999.

[2] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *proceedings of the IEEE Conference on Computer Vision & Pattern Recognition (*CVPR *2003)*, 2003, pp. 521–527.

[3] T. M. Rath, R. Manmatha, and V. Lavrenko, "A search engine for historical manuscript images," in *proceedings of the 27th Annual ACM SIGIR Conference on Research & Development in Information Retrieval (*SIGIR *2004)*, 2004, pp. 369–376.

[4] A. Vinciarelli, "Application of information retrieval techniques to single writer documents," *Pattern Recognition Letters*, vol. 26, no. 14, pp. 2262–2271, 2005.

[5] K. Terasawa and Y. Tanaka, "Slit style HOG feature for document image word spotting," in *ICDAR 2009, proceedings of 10th International Conference on Document Analysis & Recognition*, 2009, pp. 116–120.

[6] V. Govindaraju, H. Cao, and A. Bhardwaj, "Handwritten document retrieval strategies," in *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data (*AND *2009)*, 2009, pp. 3–7.

[7] D. Lopresti and A. Tomkins, "On the searchability of electronic ink," in *proceedings of the 4th International Workshop on Frontiers on Handwriting Recognition (*IWFHR *1994)*, 1994, pp. 156–165.

[8] G. Russell, M. P. Perrone, and Y. M. Chee, "Handwritten document retrieval," in *proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (*IWFHR *2002)*, 2002, pp. 233–238.

[9] A. K. Jain and A. M. Namboodiri, "Indexing and retrieval of on-line handwritten documents," in *proceedings of the 7th International Conference on Document Analysis & Recognition (*ICDAR *2003)*, 2003, pp. 655–659.

[10] C. Jawahar, A. Balasubramanian, M. Meshesha, and A. M. Namboodiri, "Retrieval of online handwriting by synthesis and matching," *Pattern Recognition*, vol. 42, no. 7, pp. 1445–1457, 2009.

[11] C. Cheng, B. Zhu, X. Chen, and M. Nakagawa, "Improvements in keyword search within handwritten digital ink," in *proceedings of 10th International Conference on Document Analysis & Recognition (*ICDAR *2009)*, 2009, pp. 863–866.

[12] S. Peña Saldarriaga, E. Morin, and C. Viard-Gaudin, "Ranking fusion methods applied to on-line handwriting information retrieval," in *proceedings of the 32nd European Conference on Information Retrieval (*ECIR *2010)*, 2010, pp. 253–264.

[13] J. Zobel and A. Moffat, "Inverted files for text search engines," *ACM Computing Surveys*, vol. 38, no. 2, 2006.

[14] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of word confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[15] L. Wang and T. Jiang, "On the complexity of multiple sequence alignment," *Journal of Computational Biology*, vol. 1, no. 4, pp. 337–348, 1994.

[16] M. Liwicki and H. Bunke, "Iam-ondb - an on-line english sentence database acquired from handwritten text on a whiteboard," in *proceedings of the 8th International Conference on Document Analysis & Recognition (*ICDAR *2005)*, 2005, pp. 956–961.

[17] P. Drouin, "Term extraction using non-technical corpora as a point of leverage," *Terminology*, vol. 9, no. 1, pp. 99–115, 2003.

[18] *Tablet PC SDK Versions*, Microsoft. [Online]. Available: http://msdn.microsoft.com/en-us/library/ms840463.aspx

[19] N. Ueffin and H. Ney, "Word-level confidence estimation for machine translation," *Computational Linguistics*, vol. 33, no. 1, pp. 9–40, 2007.

[20] K. Ng and V. W. Zue, "Subword-based approaches for spoken document retrieval," *Computer Speech*, vol. 32, no. 3, pp. 157–186, 2000.

[21] Y. Fataicha, M. Cheriet, J. Y. Nie, and C. Y. Suen, "Retrieving poorly degraded OCR documents," *International Journal on Document Analysis and Recognition*, vol. 8, no. 1, pp. 15–26, 2006.