Using Character Shape Coding for Information Retrieval

A.F. Smeaton
School of Computer Applications
Dublin City University
Glasnevin, Dublin 9, Ireland

A.L. Spitz
Daimler Benz Research
& Technology Center
1510 Page Mill Road
Palo Alto, CA 94304 USA

Abstract

In conventional information retrieval the task of finding users' search terms in a document is simple. When the document is not available in machine-readable format, optical character recognition (OCR) can usually be performed. We have developed a technique for performing information retrieval on document images in such a manner that the accuracy has great utility. The method makes generalisations about the images of characters, then performs classification of these and agglomerates the resulting character shape codes into word tokens based on character shape coding. These are sufficiently specific in their representation of the underlying words to allow reasonable performance of retrieval. Using a collection of over 250 Mbytes of document texts and queries with known relevance assessments, we present a series of experiments to determine how various parameters in the retrieval strategy affect retrieval performance and we obtain a surprisingly good results.

1 Introduction

Information retrieval (IR) finds documents from a collection which are relevant to a user's query. Normally a query is a collection of words or search terms and in conventional IR where the machine-readable version of documents is available, the task of finding users' search terms, in a document is a simple match, though the overall IR task involves much more.

IR is an inexact operation where a user's search term can appear in a document as some morphological variant of a word. For example a search for *human computer interfaces* should match against a document containing *computers interfacing with people* even accounting for the pluralisation of *computer* and the noun vs. verb forms of *interface*. This example illustrates another feature of IR that makes it an inexact and non-trivial operation; the choice of words to represent concepts has a huge variance among users as the example above illustrates. In addition, the relationship between concepts, a fundamental part of any information or knowledge representation, can have enormous ambiguity, a feature inherited from the underlying natural language we use in text. As a final complication, many words are polysemous, meaning they can have more than one meaning, such as the word *bar* meaning a place for refreshments, a material restriction as on windows, etc. [1].

In order to perform an IR match between a query and a set of documents, the most common approach is to count the number of query terms present in each document and use this, along with other inputs such as document length and term frequency, to calculate a score by which documents are ranked. If a document is not available in such form then optical character recognition may be used to interpret a scanned document image. In practice, though, the OCR operation itself is not an entirely accurate process, especially if the documents are of poor quality. As an alternative to full-scale OCR we propose a technique known as character shape coding (CSC), effectively OCR with a much reduced alphabet and perform retrieval on the CSC representation of documents.

In this paper we explore the retrieval effectiveness of an IR system based on representing documents by their CSC tokens. In our experiments, these could have been assigned as the result of an image scanning and analysis technique but they are not as our CSC representations of document texts are generated directly from a machine-readable version. Although there is likely to be some mis-recognition of search terms in documents when using a CSC representation, the noise this generates would not contribute to diminishing the effectiveness of retrieval too much. In the next section of this paper we look briefly at using OCR for IR and we present the technique of character shape coding. Section 3 explores the pitfalls and merits of using a CSC representation of documents. In section 4 we present our experimental environment, in section 5 we show our results and in the final section we present conclusions.

		CSC V_0	CSC V_2
A	←	A-Zbdfhklt	A-Zbdfhklt
x	←	acemnorsuvwxz	amorsuvwxz
g	←	gpqy	gpqy
i	←	i	i
j	←	j	j
e	←		ec
n	←		n

Table 1: CSC Mappings for V_0 and V_2

2 OCR and Character Shape Coding for Information Retrieval

It is generally acknowledged that the optical character recognition (OCR) accuracy requirements for IR are considerably lower than for many other document processing applications [2, 3]. This is based on the premise that important words in a query tend to occur frequently in relevant documents and therefore the probability that an important word will be correctly recognised at least once in a document is elevated above the overall accuracy that is characteristic of OCR performance. Several researchers have independently developed codes that capture the gross features of individual characters without the computational overhead, and susceptibility to noise, of OCR. Tanaka and Torii [4] developed a two bit code, the high order bit of which encoded x-height and non-x-height characters. The low order bit indicated whether the character form has exactly one, or more than one, crossing of a horizontal line midway in elevation between the baseline and the x-line, though this is very font dependent. Schürmann et al. [5] and Sinha [6] simply differentiate ascender and descender characters from x-height characters. In order to make the coding process more tolerant of broken and touching characters, Sinha then collapses this list of codes by reducing runs of similar codes to a single occurrence. Spitz, in his original work [7] (V_0), developed codes similar to Schürmann and Sinha but further distinguishing **i** and **j**. Spitz's later versions measured eastward [8] (V_1) and then southward [9] (V_2) concavity in order to separately identify **c** and **e**, and also **n**. V_0 and V_2 are the coding methods we use here and they are outlined below. In performing CSC recognition of input documents, the CSCs are aggregated at the word level into Word Shape Tokens (WSTs).

We identify the vertical positions of the baseline and the x-height of each line of document text. Next we count the number of connected components in each character cell and note their position with respect to the baseline and x-line. Since all character classification is based on connected component vertical location this technique tends to be largely font independent but reliant on roughly consistent point size within each isolated text line. This process is one to two orders of magnitude faster than conventional OCR techniques [10] and, because it does not rely heavily on connectivity and other distortion-vulnerable features, is quite robust [11]. The choice of which CSC version to use is a trade-off; $V_2$is more specific at identifying words because it isolates the character n and the characters c and e but at a cost of more expensive and less robust recognition.

3 Character Shape Coding for IR

The transliteration of the search terms into CSCs is a completely reliable process. The character code input is noise free, and while some information is lost in the transliteration, no noise is introduced. What we seek to do here is to investigate replacing a computationally expensive process (OCR) with an inexpensive one. There are several reasons that one might wish to do so.

As with CSCs, OCR accuracy falls off as image quality degrades. However the accuracy of the CSC process falls off more slowly than OCR since it is independent of the fine structure of letter forms resulting in situations where CSC accuracy may be better than OCR accuracy on poor quality or low resolution images (such as facsimile). There may be a type and range of degradation where OCR still produces useful results in that it correctly recognises some occurrences of search terms but often generates gibberish which is ignored as non-matching with the query, while CSC generates legitimate and matching, but incorrect, WSTs [10]. Acknowledging that the WST recognition process will generate incorrect but legitimate WSTs from noisy documents we believe that the likelihood of this happening for any significant number of terms in a user's search is small. In the limit, document image quality may degrade to the point that no useful information can be derived using either technology but the effectiveness of WST indexing could remain higher than OCR-based, as image quality degrades. In experiments in this paper we simulate rather than implement the CSC encoding process so the results we present are at an upperbound. The question we address is the present work is how effective can the indexing and retrieval of documents based on WSTs, be.

4 Experimental Environment

The TREC conference is an annual benchmarking exercise in which IR systems run the same queries on the same document set and have their top-ranked retrieved documents judged for relevance. In our retrieval experiments we used a collection of 253 Mbytes of text (74,520 unique documents) from the Wall Street Journal and 50 queries or topic statements with an average of 21 relevant documents each. Our retrieval method was to score each document in the collection based on the $tf \times IDF$ weight of a search term in a document, a well-known and well-documented retrieval strategy known to work well when document and queries are indexed by words or word stems [12]. While there are many other *smarts* we could add to this retrieval strategy to improve effectiveness such as query expansion and document length normalisation, we chose to use a plain *vanilla* retrieval strategy so we can concentrate on the WST indexing of documents.

type	types	lost	Export
Exporting	export	exported	exporter
exporters	country	countries	industry
industrial	industrialized	Industry	job
jobs	result	resulted	resulting
resultant			

Table 2: Final Set of Search Terms used in Query 251

We indexed documents by representing them as the set of WST representations for all their surface word forms, i.e. the actual form of word occurrences in document text. We then processed queries by removing stopwords and reverse-stemming remaining words using Porter's stemming algorithm [13] to stem a large portion of text and recording, for each unique word stem, the surface word form occurrence from the text which yielded that stem. This pre-processing stage was used to automatically generate our mapping of word stems to surface form word occurrences. Each word stem in the query was then expanded into the set of surface forms in order to generate morphological and surface variants attributed to pluralisation and verb endings and to various letter capitalisations respectively. The set of expanded terms per query was then pruned manually to eliminate errors due to stemming and unlikely word form occurrences. For example, one query asked for information about lung cancer and the word stem for *lung* is *lung* which is a word stem shared by various forms of the verb *to lunge*. These were eliminated from the query set and the remaining word form occurrences were turned into WSTs for each of the 2 CSC versions and used as query terms for the experiments.

For example, query 251 is: *Exportation of Industry: documents will report the exportation of some part of U.S. Industry to another country. Relevant documents will identify the type of industry being exported, the country to which it is exported; and as well will reveal the number of jobs lost as a result of that exportation.*

After manually pruning the generated surface word form, the list of search terms used is shown in Table 2. Each of these was turned into a WST for both $V_0$and V_2 CSC mappings and the process described above yielded an average of 24.96 WSTs per query. In terms of retrieval effectiveness, the results obtained are presented in Table 3, along with an upperbound retrieval effectiveness based on conventional IR using word stemming and stopword removal on document texts. This upperbound represents the best theoretically achievable limit for our WST-based retrieval though other IR techniques could be used to improve this upperbound further.

As can be seen, the above results reprinted from [12] are quite disappointing for WST-based retrieval and in this paper we set out to improve them by reducing the amount of noise in the query WSTs.

	Upperbound	V_0 WSTs	V_2 WSTs
P @ 0.0	0.4015	0.0480	0.1261
P @ 0.1	0.3156	0.0371	0.0958
P @ 0.2	0.2365	0.0161	0.0658
P @ 0.3	0.1746	0.0052	0.0363
P @ 0.4	0.1393	0.0027	0.0254
P @ 0.5	0.1185	0.0024	0.0205
P @ 0.6	0.0834	0.0011	0.0161
P @ 0.7	0.0706	0.0010	0.0079
P @ 0.8	0.0493	0.0001	0.0014
P @ 0.9	0.0366	0.0001	0.0007
P @ 1.0	0.0360	0.0001	0.0007
Av. P	0.1365	0.0079	0.0303
P @ 10	0.1500	0.0200	0.0680
P @ 30	0.0900	0.0107	0.0413

Table 3: Performance of Upperbound and Entry-level WST-based retrieval

Word	type	lost	job	industrial
CSC V_0	17	1010	6	2
CSC V_2	5	503	4	1

Table 4: Number of Entries in Comprehensive Lexicon Sharing WST for Some Search Terms

5 Experimental Results

One of the reasons why our initial experiments with WST-based retrieval yielded such poor effectiveness was that each of the WSTs in each CSC mapping maps to multiple surface form occurrences in the document texts, and the distribution of these is very skewed. In ongoing work, Spitz has been collecting a comprehensive lexicon of 318,636 unique surface form word occurrences in text. Using some of the search terms from the TREC query introduced earlier, Table 4 shows the number of surface form occurrences from this lexicon which share the same WST as the search term indicated: Clearly there are some WST search terms which introduce a lot of noise into the retrieval process in that the WST form of that search term will score documents containing each of the word forms sharing that WST, a clearly undesirable feature. Even for the V_2 mapping there are many search terms which introduce noise such as *lost* in the above, and these should be pruned from the query. We ran a series of experiments for V_0 and V_2 of the CSC mapping in which we gradually eliminated WST search terms based on the number of surface forms which shared that WST in our lexicon. We started with no pruning at all ("any") and then pruned out search terms sharing their WST with 20 or more other WSTs, then 15, then 10 and so on. The results of these, along with the unpruned queries are presented below in Table 5 for V_0 and Table 6 for V_2.

Before examining these results more closely it is worth re-capping exactly what we are doing here. For

Prune if WST shared with: Av # terms	any 24.9	< 20 12.83	< 15 11.42	< 10 9.12	< 5 6.43	< 3 4.61	< 2 3.46	unique 2.32
P @ 0.0	0.0480	0.1695	0.1753	0.1875	0.1753	0.1763	0.2209	0.2036
P @ 0.1	0.0371	0.1257	0.1315	0.1457	0.1216	0.1155	0.1414	0.1248
P @ 0.2	0.0161	0.0667	0.0684	0.0818	0.0814	0.0783	0.0901	0.0640
P @ 0.3	0.0052	0.0332	0.0320	0.0438	0.0352	0.0320	0.0311	0.0237
P @ 0.4	0.0027	0.0266	0.0219	0.0307	0.0266	0.0191	0.0206	0.0152
P @ 0.5	0.0024	0.0223	0.0167	0.0267	0.0252	0.0169	0.0181	0.0136
P @ 0.6	0.0011	0.0139	0.0098	0.0207	0.0105	0.0038	0.0040	0.0046
P @ 0.7	0.0010	0.0069	0.0061	0.0155	0.0088	0.0023	0.0027	0.0025
P @ 0.8	0.0001	0.0037	0.0032	0.0089	0.0068	0.0017	0.0011	0.0013
P @ 0.9	0.0001	0.0025	0.0018	0.0073	0.0057	0.0007	0.0004	0.0001
P @ 1.0	0.0001	0.0021	0.0018	0.0073	0.0057	0.0007	0.0004	0.0001
Av. P	0.0079	0.0361	0.0352	0.0442	0.0378	0.0335	0.0380	0.0322
P @ 10	0.0200	0.0571	0.0571	0.0612	0.0542	0.0574	0.0638	0.0619
P @ 30	0.0107	0.0361	0.0374	0.0361	0.0403	0.0376	0.0404	0.0357

Table 5: Retrieval for pruned V_0 WSTs

Prune if WST shared with: Av # terms	any 24.9	< 20 21.86	< 15 21.22	< 10 20.28	< 5 18.58	< 3 16.54	< 2 16.54	unique 10.82
P @ 0.0	0.1261	0.3081	0.3060	0.2865	0.2955	0.2731	0.2731	0.2628
P @ 0.1	0.0958	0.2050	0.2116	0.2100	0.2212	0.2146	0.2146	0.1975
P @ 0.2	0.0658	0.1416	0.1454	0.1612	0.1405	0.1416	0.1416	0.1425
P @ 0.3	0.0363	0.0816	0.0847	0.0789	0.0685	0.0686	0.0686	0.0820
P @ 0.4	0.0254	0.0629	0.0649	0.0610	0.0534	0.0559	0.0559	0.0688
P @ 0.5	0.0205	0.0591	0.0617	0.0574	0.0496	0.0507	0.0507	0.0577
P @ 0.6	0.0161	0.0412	0.0405	0.0417	0.0332	0.0329	0.0329	0.0259
P @ 0.7	0.0079	0.0304	0.0298	0.0312	0.0202	0.0191	0.0191	0.0178
P @ 0.8	0.0014	0.0167	0.0168	0.0174	0.0093	0.0085	0.0085	0.0092
P @ 0.9	0.0007	0.0147	0.0147	0.0146	0.0070	0.0059	0.0059	0.0082
P @ 1.0	0.0007	0.0122	0.0122	0.0122	0.0045	0.0031	0.0031	0.0048
Av. P	0.0303	0.0758	0.0774	0.0756	0.0678	0.0670	0.0670	0.683
P @ 10	0.0680	0.0980	0.1000	0.1120	0.1102	0.1061	0.1061	0.1041
P @ 30	0.0413	0.0753	0.0753	0.0753	0.0748	0.0789	0.0789	0.0694

Table 6: Retrieval for pruned V_2 WSTs

CSC mapping V_0 say, when we filter WST search terms which share their WST with < 20 surface word forms we are eliminating WSTs from our queries which have been generated by search terms like *Bank, Banking, Fossil*, etc. For the most part, the terms removed by this least severe filter all have their first letter capitalised which does cause problems for queries where proper names are an integral part of the query as in *Soviet, German, Europe* and so on. Other words not in this category include terms like *costs* and *hackers* which we know introduce much noise into the retrieval operation. The word hacker for example shares its WST with 1744 other word surface forms with the V_0 CSC mapping and with 92 others when using V_2.

The results in Table 5 show that pruning V_0 WST search terms in the way described above clearly does improve retrieval effectiveness by many times with a frequency threshold of about 10 surface word forms sharing a WST being the best though it is still not as effective as word-based retrieval. When using the V_2 encoding as shown in Table 6, the results we obtain are also an improvement over no search term pruning with the best effectiveness at a pruning threshold of about 15 surface forms. Clearly, eliminating these noisy search terms from the query improves effectiveness.

6 Conclusions and Prospects

If we examine the published TREC-5 proceedings for the performance of the IR systems which took part in the TREC-5 evaluations [14] then we find that 10 groups are represented, including our entry-level WST-based retrieval as shown in Table 3. In our results since then and presented in section 5, our best WST-based performance, using V_2 with a pruning threshold of 15, gives precision-recall figures which are better than some of those other systems. This is a much better level of effectiveness than we expected as we still have further avenues we need to pursue in order to improve retrieval quality.

Precision and recall evaluation figures are often difficult to interpret so what do our performance figures actually mean ? The average number of relevant documents for the 50 queries we use is 21 so that at a recall value of 0.1, 2.1 relevant documents will have been retrieved. At this point, a precision value of 0.21, as we get with our best V_2 WST representation, means that those 2.1 relevant documents appear among the first 10 (2.1/0.21) documents retrieved. Our best V_0 WST representation at the same recall point of 0.1 is 0.14 meaning that those 2.1 relevant documents appear among the first 14.48 retrieved documents, so clearly this is not a high precision retrieval strategy.

Because the precision-recall results we have presented are averaged over a set of 50 queries it is difficult to see the effect that our search term pruning has on effectiveness. From an examination of the performance of each query in the set, a failure analysis shows that some queries receive a significant improvement in effectiveness while others are so severely pruned that all the concepts in the query are eliminated and it is effectively random noise we retrieve. These *empty* queries pull down the averaged precision and recall figures we present here. This suggests that we need a query-by-query refinement of which WST search terms to include and constructing a purely algorithmic approach to selecting query WSTs based on their frequencies and the number of surface forms sharing WSTs is not likely to be very profitable if at all.

We believe that a retrieval strategy where the user is actively involved in WST selection where those WSTs are derived from search terms and their variants as we have done, is a useful line of inquiry. We have found that in some queries certain WSTs should be pruned from the search term set leaving enough concept terms to carry the search to fruitful retrieval, while in other queries, WSTs sharing exactly that same number of word surface forms should not be removed. This dependency on the context of the query cannot be programmed algorithmically as it needs to know about query concepts and should be for the user to decide. We are presently building such an interactive system.

References

[1] A.F. Smeaton, "An Overview of Information Retrieval", in *Information Retrieval and Hypertext*, M. Agosti and A.F. Smeaton (Eds.), Kluwer Academic Publishers, 1996.

[2] D.J. Ittner, D.D. Lewis & D.D. Ahn, "Text characterization of low quality images", *Proceedings Symposium on Document Analysis and Information Retrieval*, Las Vegas, pp.301-315, 1994.

[3] K. Tagva, J. Borsack, A. Condit & S. Erva, "The Effects of noisy data on text retrieval", *Journal of the American Society for Information Science*, Vol. 45, No. 1, pp. 50-58, 1994.

[4] Y. Tanaka & H. Torii, "Transmedia machine and its keyword search over image texts", Proceedings of Recherche d'Information Assistee par Ordinateur (RIAO), Cambridge, MA, pp 248-258, 1988.

[5] J. Schürmann, N. Bartneck, T. Bayer, J. Franke, E. Mandler & M. Oberländer, "Document analysis - From pixels to contents", *Proceedings of the IEEE*, 80(7), pp 1101-1119, 1992.

[6] R. M. K. Sinha, "On partitioning a dictionary for visual text recognition", *Pattern Recognition*, 23(5), pp 497-500, 1990.

[7] A.L. Spitz, "Generalized line, word and character finding", *Progress in Image Analysis and Processing III*. S. Impedovo (ed.), pp 377-383, World Scientific, 1993.

[8] A.L. Spitz, "Using character shape codes for word spotting in document images", *Shape and Structure in Pattern Recognition*, D. Dori and A. Bruckstein (eds.), World Scientific, 1995.

[9] J.C. Reynar, A.L. Spitz & P. Sibun, "Document reconstruction: A thousand words from one picture", *Symposium on Document Analysis and Information Retrieval*, Las Vegas, 1995.

[10] T. Nakayama, "Content-oriented categorization of document images", *Proceedings COLING*, Copenhagen, 1996.

[11] A.L. Spitz, "Moby Dick meets GEOCR: Lexical considerations in word recognition", *Proc ICDAR 97* (this volume).

[12] F. Kelledy & A.F. Smeaton, "TREC-5 Experiments at Dublin City University: Query Space Reduction, Spanish & Character Shape Encoding", in [14].

[13] M.F. Porter, "An Algorithm for Suffix Stripping", *Program*, 14(3), 130-137, 1980.

[14] D.K. Harman, (Ed.) "Proceedings of TREC-5" *NIST Special Publication*, (to appear) 1997.