Probabilistic Retrieval of OCR Degraded Text Using N-Grams

S.M. Harding[1], W.B. Croft[1] and C. Weir[2]

[1] CIIR, University of Massachusetts, Amherst MA 01003, USA
[2] Lockheed Martin C2 Systems, Frazer, PA 19355, USA

Abstract. The retrieval of OCR degraded text using n-gram formulations within a probabilistic retrieval system is examined in this paper. Direct retrieval of documents using n-gram databases of 2 and 3-grams or 2, 3, 4 and 5-grams resulted in improved retrieval performance over standard (word based) queries on the same data when a level of 10 percent degradation or worse was achieved. A second method of using n-grams to identify appropriate matching and near matching terms for query expansion which also performed better than using standard queries is also described. This method was less effective than direct n-gram query formulations but can likely be improved with alternative query component weighting schemes and measures of term similarity. Finally, a web based retrieval application using n-gram retrieval of OCR text and display, with query term highlighting, of the source document image is described.

1 Introduction

A major problem with retrieval of OCR text from image data is the inevitable corruption of characters that result from even the best image analysis system. Although OCR errors have little effect on retrieval with good quality input, effectiveness can be significantly reduced in short texts with poor image or scanning quality [7, 4].

In an effort to reduce these losses, we have incorporated the use of n-grams for the representation of document words and user query terms using a probabilistic retrieval system with no modification to the evaluation process. N-grams have been frequently used for word representation to address issues such as multiple character sets, language independence, and spell correction [5].

We have incorporated this research in retrieval and highlighting of image query text using an Intelligent Document Understanding System (IDUS) developed at Lockheed-Martin Corporation [8] and INQUERY, a full text probabilistic retrieval system developed at the University of Massachusetts [9], [1]. The system allows retrieval of text images based on automatic OCR and logical text analysis of image content [11].

In this paper, we describe our efforts in using n-grams in query formulations and expansions with the goal of obtaining enhanced retrieval performance on OCR data. The following sections contain descriptions of retrieval performance experiments using various n-gram query formulations, descriptions of the use of

n-grams in finding terms for query expansion without directly evaluating the n-grams themselves, a brief description of the web application that uses n-gram retrieval of images and finally, some conclusions and future directions for this work.

2 Retrieval Using N-grams

2.1 Indexing N-grams

Our n-gram approach to retrieval of OCR degraded text was to develop a database indexing and query formulation method that produced the best combined 2 and 3-gram indexing and 2, 3, 4 and 5-gram indexing, where each word in the database was represented by both the full word and a combination of its n-gram constituents. Thus the word "Mexican" would be indexed to an INQUERY database as shown in Table 1.

The n-grams are then combined to capture word representation using various INQUERY proximity operators [1]. The key indexing requirement for effective use of these structure operators is forcing all n-grams contained within a word token to have the same document word position as its encompassing word. Thus if the word "Mexico" occupied location 5 in the document text, then all n-grams extracted and saved to the database from this word would also occupy position 5.

2.2 N-gram Binding Operators

At query time, user query terms are broken into their constituent n-gram components and bound together using an INQUERY proximity operator before being submitted to the system for evaluation. The goal is to precisely define a concept (or word) using the n-gram structure without over specifying it and possibly losing relevant documents in the returned rankings.

INQUERY proximity operators were tested to determine the best operator to use as a binding operation for word constituent n-grams. The best binding operator should be strict enough that it not allow extraneous "noise" terms into the best ranked returned documents, yet "loose" enough that a missing n-gram component would not cause the elimination of relevant texts.

INQUERY structure operators considered for n-gram binding operations included the following:

- Strict proximity is represented by a #0 operation (meaning all n-gram components must occur in the same word location) and results in relevant documents being eliminated from the top rankings if the query formulation is not precisely specified.
- A probabilistic AND operation, #and, allows various n-gram components to be missing from a sought document, but downweights that document's ranking accordingly.

- #passage3 or #passage5 operations rank documents containing the n-gram components within windows of three or five word positions. The document is assigned a belief according to its best ranked passage. A passage operation also accounts for n-gram components that may reside across word boundaries. This is useful for OCR text because spaces are very commonly added to text creating two or more words where originally there was only one. N-grams combined with this operation can capture relevant documents whose OCR text have experienced this type of error.

Experiments were run to test the effectiveness of various n-gram binding operators in maximizing retrieval performance.

Example word token: Mexican

Ordered sequence of two, three, four and five-character n-grams:

me	mex	mexi	mexic	ex	exi	exic	exica	xi	xic	xica	xican	ic	ica	ican	ca	can	an
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

N-grams retained for indexing:

Leading three n-grams at positions 0, 1 and 2	me mex mexi
Trailing two n-grams at positions 16 and 17	can an
First of middle three n-grams at position $((N-4)/3)+2=7$ where N=18 and division of 14 by 3 is rounded up	exica
Second of "middle three" n-grams at position $((N-4)/2)+2=9$	xic
Third of "middle three" n-grams at position $(((N-4)/3)+2)x2=14$	ica

Table 1. Method for determining which n-grams of a word token to select for indexing.

2.3 N-gram Sampling Within Words

N-gram databases are much larger than those containing words alone and the proximity operations performed to reconstitute search concepts may be costly in terms of processing time. In an effort to reduce the size of the n-gram database

Plain Text Query: Mexican environmental newsletters

Translated Query: #wsum(10 9 #sum(Mexican environmental newsletters)

5 #sum(#passage5(me mex mexi exica xic ican can an)
#passage5(en env envi ironm onm ment tal al)
#passage5(ne new news sl let tt ers rs)
))

Table 2. Query formulation using original users query combined with an n-gram component. The user's query terms contribute approximately twice the weight of the n-gram component in the formulation.

and the number of n-grams within a proximity operation, a sampling of at most eight n-grams from a word were used rather than the full n-gram complement. If a word contained less than eight 2, 3, 4 and 5-grams (for a 2-5 n-gram database) or less than eight 2 and 3-grams (for a 2-3 n-gram database), then all the constituent n-grams were used.

The sample included the first three leading and two trailing n-grams. A further three n-grams were taken from a dispersion of n-grams in the "middle" of the word. It is important that the sampling be consistent between the indexing process adding n-gram samples to the database, and the sampling used to transform a user's terms into an n-gram formulation at query time, or less likely matching may occur. Table 1 shows examples of the word n-gram sampling process.

2.4 N-gram Query Formulation

A user's query is automatically reformulated using the appropriate n-gram samples from query terms and a binding operator, and submitted to the system for retrieval. The final representation of the transformed query is in the form of a weighted sum operation (#wsum) in which the original user's query terms are highly weighted while the n-gram formulations derived from them are weighted at approximately half that of the full terms. If a full query term occurs in a document, it should have a higher concept belief than a form consisting of constituent n-grams.

A weighted sum operation allows constituent parts of a query to have varying weights, reflecting the importance of those concepts in retrieval of the documents. Table 2 illustrates the transformation of a user's query into the form actually submitted to the system for retrieval. The two components of the #wsum operation have weights of 9 and 5 for user and n-gram representations. These numbers simply reflect relative importance of the query components and were

decided upon somewhat arbitrarily, but reflecting the belief that user provided concepts are more important than those represented by the n-grams.

2.5 N-gram Retrieval Performance

A series of experiments was run to determine retrieval effectiveness of various n-gram query formulations and database indexing methods. The measurement criterion was the best performance improvement of a technique as measured by the highest percent improvement in average precision over all recall levels when compared to a baseline (non n-gram) database and queries.

These experiments were performed using four different databases that were randomly degraded using data developed by the University of Nevada at Los Vegas (UNLV) [6]. Higher word error rates were used to further degrade the database when more severe corruption was to be tested. Words were corrupted using character recognition confusions typical of OCR systems, including the inclusion or deletion of spaces in words. Precise error rates for the databases is unknown and is measured only by how much worse (percentage) standard query sets performed on the degraded database versus the non-degraded data.

Test collections with small individual document sizes are more easily degraded than collections with larger sized documents [4] since large documents are more likely to have other words representing the search concept that have not been corrupted. The larger document length collections had to be degraded over several iterations before performance differences were measurable. Test collection characteristics are summarized in Table 3. In general, once a collection achieved a 10% level of degradation (that is, queries on the degraded database produced 10% lower average precision over all recall levels than the same query set on the non-degraded collection), n-gram formulations began to achieve higher performance than non n-gram queries. N-gram retrieval always performed worse on degraded collections than standard queries did on non-degraded data, as would be expected. A summary of the best retrieval performance is provided in Table 4.

	CACM–6	NPL–6	TIME–7	WSJ89–6
Docs	3204	11429	423	12380
Queries	50	93	83	49
Avg. Words/Doc	64	42	591	512

Table 3. Summary of test collection characteristics.

Over all collections, the #passage5 operator generally performed best at binding the n-grams together during query formulation, although not always significantly so. The #and operator was not strict enough in binding n-grams together to represent query concepts, resulting in more non-relevant documents

	CACM–6	NPL–6	TIME–7	WSJ89–6
Degradation	−26.6	−52.9	−10.9	−19.6
2-3 gram	+8.4	+36.0	+3.9*	+ 4.5
2-5 gram	+11.1	+35.9	+3.0	+11.5*
Restricted	+14.7	+38.8	+3.8*	+11.9
Weighted	+10.8	+38.1	+5.1*	+11.7*

Table 4. Summary of retrieval performance experiments using assorted n-gram query formulations. The reported values are percent increases in average precision over all recall levels compared to standard queries (no n-grams). Entries marked with asterisk indicate an operator other than #passage5 produced the indicated performance.

being retrieved in the high rankings. The #0 operator was conversely too strict and removed relevant documents from high rank because of missing n-gram components. As database degradation increased, n-gram retrieval performance improved over standard queries.

Marginal performance improvement occurred in the TIME collection. This collection is the most difficult to degrade because of the larger number of words in each article and unusually high baseline retrieval performance characteristic of this collection. The best n-gram binding operator to use with this collection was also unclear, probably also due to the low level of collection corruption achieved.

Performance of n-gram databases that included 2 and 3-grams versus those including 2, 3, 4 and 5-grams was also not clearly shown. However, when the databases were restricted to contain only the maximum eight n-grams samples per word, the 2-5 n-gram "restricted" databases out-performed "restricted" 2-3 n-gram databases, making these the best overall in retrieval performance.

Query formulations using weighted n-grams based on their positions in the constituent word did not perform significantly better than those ignoring n-gram position. The position weighted n-gram query forms gave half as much importance (weight) to the middle three n-grams as to the leading three, and half again less importance to the trailing two n-grams as to the middle three. This formulation was based on the assumption that leading n-grams might be more important in defining a concept than trailing n-grams, as has been the case with phonetic representation of words.

3 N-grams Based Query Term Expansion

Another n-gram query technique was tested in an effort to reduce storage and evaluation time and improve retrieval performance. This approach uses n-grams to discover words that match and "nearly" match target terms, then add these additional terms to the original query. This approach has wide appeal since it could be largely language independent and could be applied to various concept

(word) representations such as phonemes, soundex codes [14, 13] or for spelling correction [12], using differing retrieval engines [2], or as a means to summarize the content of a document [3].

3.1 An N-gram to Term Database

Expanding user provided query terms with matching and closely matching words is accomplished by converting the query terms to their 2-gram components and querying a collection word database, containing all the words, in lower case form, found in the text collection.

This collection word database is created by forming SGML "pseudo-documents" where a collection word is the pseudo-document title. The collection of pseudo-documents is then indexed to a 2-gram INQUERY database. Table 5 illustrates the conversion of a an SGML document into one of it's corresponding SGML tagged pseudo-documents that will be indexed into the collection term database. The size of this database will be very much smaller than an n-gram database for a source text collection because each word in the collection is represented only once. Further reductions are made by eliminated number strings as terms needing expansion. A document format other than SGML could be used to reduce the size of the raw pseudo-document collection.

After the pseudo-documents are indexed, 2-gram queries to this database will produce a listing of pseudo-document "titles", that actually represent candidate terms that match or closely match the user's query term. A list of the top 20 ranked terms are further restricted through the use of a Qgram Distance measure described below. The final expanded query contains the sum of all #syn operations for each user provided query term.

3.2 Determining Nearness of Match

The problem of determining a measure of closeness of match has been widely studied. The principle measure used for filtering candidate terms was the edit distance, or number of single insertions, deletions, or additions needed to make one string the same as another. Further complexity may be added to this measure by applying a "cost" of additional operations, such as character transposition, or special substitutions, as with common OCR errors [13].

We chose to keep the method simple and use Ukkonen's [10] Qgram Distance measure, which counts the number of n-grams contained in two words versus the number they share. The simplest form of this measure is

$$QD(s,t) = |G(s)| + |G(t)| - 2 * |G(s) AND G(t)|$$

where s and t are two strings to be matched and G(x) is the set of 2-grams in string x.

One need only determine the threshold value of this measure to decide whether the term should be included in the expansion or eliminated. Other

Original Document Content:

⟨*DOC*⟩
⟨*DOCNO*⟩*TIME* – –002⟨*DOC*⟩
⟨*TITLE*⟩
RussiaWho'sInChargeHere?
⟨/*TITLE*⟩
...
⟨/*DOC*⟩

Term List:
Russia
Who's
In
Charge
Here

Pseudo-document for the First Term ("Russia"):

⟨*DOC*⟩
⟨*DOCNO*⟩1⟨/*DOCNO*⟩
⟨*TITLE*⟩
Russia
⟨/*TITLE*⟩
⟨/*DOC*⟩
...
2-GRAM Indexing of the "Russia" Pseudo Document:

Russia – – –⟩ ru us ss si ia

Table 5. Pseudo-document representation of collection terms for a term collection database.

criteria are easily added to this measure such as a collection frequency value to determine if the two comparison strings may be terms in their own right, or simple misspellings (one would keep a misspelled word).

3.3 Query Expansion

Once candidate terms have been selected, they are included in the expanded query formulation and submitted for evaluation. Expansion terms are bound to their respective primary (user supplied) query terms using a synonym operation that treats all its term arguments as similar forms of a term. Table 6 shows

Query Term : Kennedy

2-gram Query: #passage5 (ke en nn ne ed dy)

Top Matching Words and Qgram Distance Measures :

~Kennedy	2
nt~Kennedy	3
Kennedy	0
ennedy	1
Kenn~dy	4
Kennady	4
Kenned~y	3
Kenned~	2
Kennedianai	5
Ken~edy	4
nnedye	3
nnedy	2
K~~nnedy	5
Knnedy	3
Kcnnedy	4
annedy	3
Kenneth	4
kennel	3
drunkennes	8

Expanded Query comprised of candidate terms with Qgram distance measures of 3 or less :

#syn (~Kennedy nt~Kennedy Kennedy ennedy Kenned~y Kenned~
nnedye nnedy Knnedy annedy kennel)

Table 6. Query term expansion using 2-grams indexed from a terms pseudo document collection. A Qgram measure is used to restrict candidate terms and a synonym operation binds the expansion terms together for evaluation. The Qgram threshold value used is 3.

candidate terms from the terms database and their Qgram distance measure, as well as the final query expansion from the original user's query. There are no weighted query components as with the previously described n-gram query formulation.

3.4 Query Expansion Performance

Evaluation of the performance of this method of retrieval on degraded text was identical to that used in evaluating direct n-gram retrieval. Various forms of expanded query were applied to the four degraded databases and the change in average precision over all recall levels was tested. The forms of query expansion tested were those using Qgram distance measure thresholds 1, 2 and 3. A summary of these experiments is given in Table 7.

The best retrieval performance was achieved on all four test collections using a Qgram distance measure of 3 (allowing more expansion terms) as an expansion term candidate cutoff value. Except for the CACM collection using a Qgram distance threshold of 1, retrieval performance using this technique was superior to that using standard queries. However this performance was not generally not as good, especially for the NPL collection, as that achieved using direct n-gram formulated queries.

Reasons for lessened performance of n-gram based query expansion versus direct n-gram query formulation may rest with non-optimal formulation of the expanded query (only one of many possible formulations was tried). Since retrieval performance improved with increasing Qgram distance thresholds, it is possible that still higher cutoff values might continue to improve retrieval performance.

Qgram Distance Threshold	CACM–6	NPL–6	TIME–7	WSJ89–6
1	−2.0	+9.5	+5.5	+4.8
2	+5.6	+14.2	+8.2	+7.9
3	+10.4	+18.4	+9.2	+10.4

Table 7. Summary of retrieval performance experiments using expansion terms derived from an 2-gram term database and three Qgram distance threshold levels. Reported values are changes in average precision over all recall levels compared to that of standard queries.

4 An N-gram Retrieval Application

A web based application using Netscape and a Java applet was developed to demonstrate the direct n-gram retrieval method. The demonstration database was a series of EnviroMexico newsletters in TIFF format covering four years from 1990 through 1993 concerning environmental issues in the US-Mexico border region.

The images were scanned and automatically segmented, OCR processed and logically analyzed to group text blocks into document articles. Articles were not

considered to extend beyond a single page, but article title and body were distinguished when possible and indexed accordingly, using n-gram representations.

Five web pages are presented to the user. The first is the database selection form shown in Figure 1. The databases may be running on different machines and require only that INQUERY servers are running to provide database service.

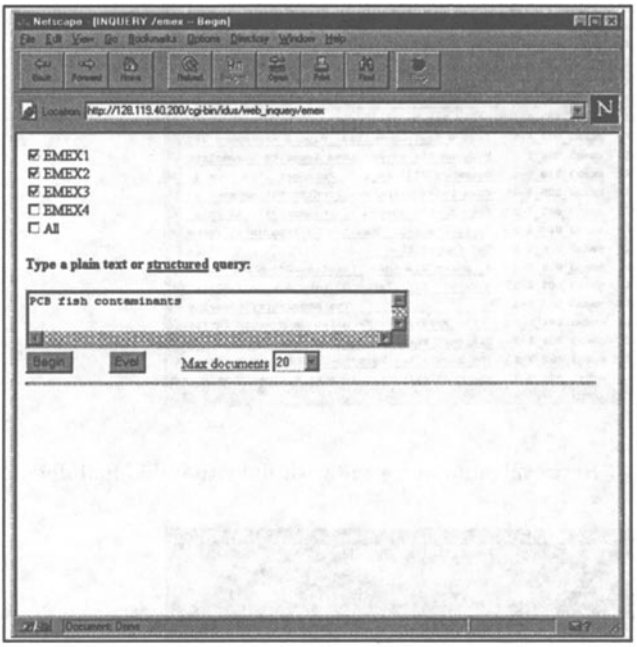

Fig. 1. Skeleton web application database selection and query input page.

The "Begin" button activates the selected databases and brings up the query form. The user types the query in natural language into the text area provided. When complete, the "Eval" button begins query evaluation.

The user's query will be submitted to all active database and the results merged into a single ranked list with best documents at the head of the list. This page (Figure 2) will display a generalized ranking in the form of 0-5 stars indicating document importance relative to all documents retrieved. A document ID and database name from which it came is also displayed. A user selectable link represents the article title.

Also shown on this page is the user's original query and the n-gram formulation derived from it, which was submitted to the system for evaluation.

Selecting a title link brings up the document text page (Figure 3), showing the OCR text from the article's body, with n-gram "hits" boldly highlighted.

This gives the user an idea as to why this document may have been retrieved.

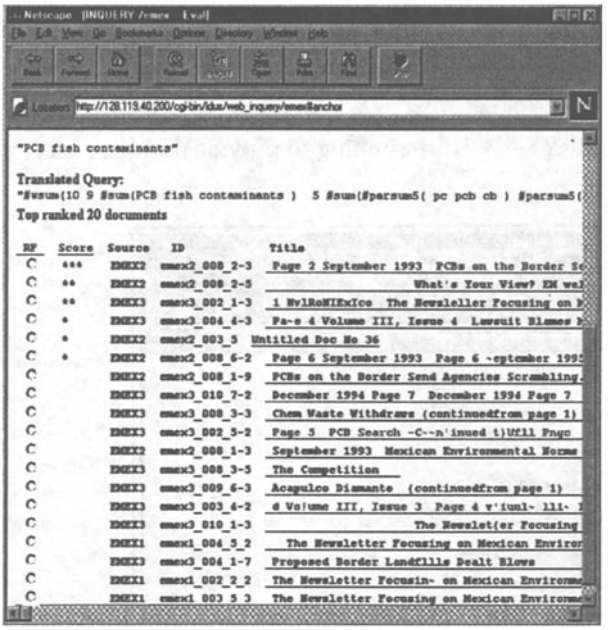

Fig. 2. Retrieval summary page with links to individual documents.

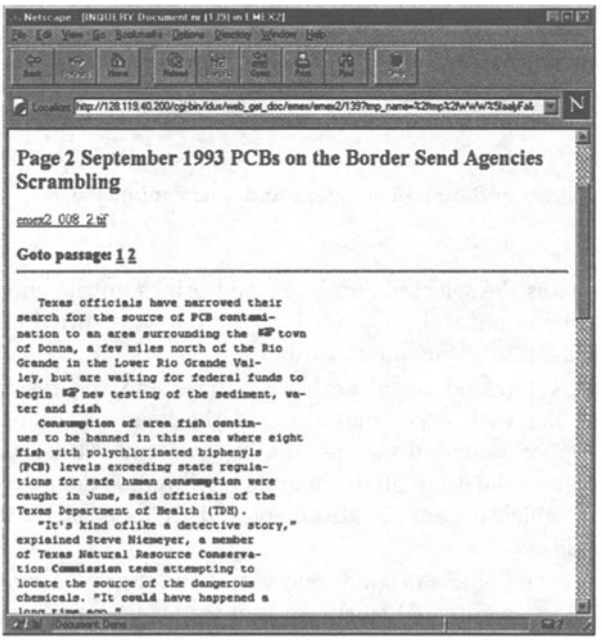

Fig. 3. OCR text of selected document with n-gram based query term highlighting.

It may also show query words in the document text that would have been missed due to OCR error without the n-gram representation, such as "mcxico" in place of "mexico". A link to an image of the actual document page is also provided so the user may view the original source.

Selecting this link brings up the image from which the article was derived, with the user's query terms underlined in red to more rapidly see why this image was retrieved (Figure 4). The highlighting information is available with the software used for image analysis.

Fig. 4. Selected document image with highlighted query terms.

5 Conclusions and Future Work

The use of n-grams in evaluation of OCR degraded text improves retrieval performance once the level of degradation reaches 10 percent, as measured by reduced average precision over all recall levels of standard, non n-gram queries, applied to the degraded collection. N-gram based query retrieval performance improves over that of standard queries as collection degradation increases.

N-gram representations of collection concepts is costly in terms of space required to store a much enlarged database and the time the retrieval engine requires in evaluating the more complex n-gram queries.

Query term expansion using n-grams and a Qgram distance measure performed better than standard queries, and continued to improve as the Qgram distance cutoff value was increased. However this performance does not yet match that of direct n-gram based queries. It is expected n-gram based query expansion will improve with other query formulation techniques, different query component weighting and other word match measures. Term expansion does considerably reduce the space required for an n-gram database used for query evaluation.

Future work will examine additional methods for reducing the resources needed to derive expansion term candidates, and better closeness measures to eliminate spurious terms in the expansion.

References

1. Callan, J.P., Croft, W.B. and Harding, S.M.: The INQUERY Retrieval System. In Proceedings of the 3rd International Conference on Database and Expert Systems Applications (1992) 78–83.
2. Cavnar, W.: Using an N-Gram-Based Document Representation with a Vector Processing Retrieval Model. In Overview of the Third Text Retrieval Conference (TREC- 3), D.K. Harman, Editor (1994) 269–278.
3. Cohen, D.J.: Highlights: Language and Domain-Independent Automatic Indexing Terms for Abstracting. J. Amer. Soc. Info. Sci. **46** (1995) 162–174.
4. Croft, W.B., Harding, S.M., Taghva, K. and Borsack, J.: An evaluation of Information Retrieval Accuracy with Simulated OCR Output. Symposium of Document Analysis and Information Retrieval (1994).
5. Pierce, C. and Nicholas, C.: TELLTALE: Experiments in a Dynamic Hypertext Environment for Degraded and Multilingual Data. J. Amer. Soc. Info. Sci **47** (1996) 263–275.
6. Rice, S., Kanai, J. and Nartker, T.: An Evaluation of Information Retrieval Accuracy. In UNLV Information Science Research Institute Annual Report (1993) 9–20.
7. Taghva, K., Borsack, J., Condit, A., Erva, S.: The effects of noisy data on text retrieval. In UNLV Information Science Research Institute Annual Report (1993) 71–80.
8. Taylor, S.L., Lipshutz, M., Dahl. D.A. and Weir, C.: An Intelligent Document Understanding System. In Second International Conference on Document Analysis and Recognition (1993) 107–220.
9. Turtle, H. and Croft, W.B.: Evaluation of an Inference Network-Based Retrieval Model. ACM Trans. on Info. Sys. **9** (1991) 187–222.
10. Ukkonen, E.: Approximate String-Matching with Q- grams and Maximal Matches. Theor. Comp. Sci. **92** (1992) 191–211.
11. Weir, C., Taylor, S.L., Harding, S.M. and Croft, W.B.: The Skeleton Document Image Retrieval System. In Symposium on Document Image Understanding Technologies (1997).
12. Zamora, A.: Automatic Detection and Correction of Spelling Errors in a Large Data Base. J. Amer. Soc. Info. Sci. **31** (1980) 51–57.
13. Zobel, J. and Dart, P.: Finding Approximate Matches in Large Lexicons. Soft. Pract. and Exper. **25** (1995) 331–345.

14. Zobel, J. and Dart, P.: Phonetic String Matching: Lessons from Information Retrieval. In Proceedings 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (1996) 166–173.