Improving Information Retrieval Performance on OCRed Text in the Absence of Clean Text Ground Truth

Kripabandhu Ghosh[a,1,*], Anirban Chakraborty[a,1,*], Swapan Kumar Parui[a,1,*], Prasenjit Majumder[b,2,*]

[a] *Indian Statistical Institute, 203 Barrackpore Trunk Road, Kolkata 700108, West Bengal, India*
[b] *Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Near Indroda Circle, Pin-382007, Gujarat, India*

Abstract

OCR errors in text harm information retrieval performance. Much research has been reported on modelling and correction of Optical Character Recognition (OCR) errors. Most of the prior work employ language dependent resources or training texts in studying the nature of errors. However, not much research has been reported that focuses on improving retrieval performance from erroneous text in the absence of training data. We propose a novel approach for detecting OCR errors and improving retrieval performance from the erroneous corpus in a situation where training samples are not available to model errors. In this paper we propose a method that automatically identifies erroneous term variants in the noisy corpus, which are used for query expansion, in the absence of clean text. We employ an effective combination of contextual information and string matching techniques. Our proposed approach automatically identifies the erroneous variants of query terms and consequently leads to improvement in retrieval performance through query expansion. Our proposed approach does not use any training data or any language specific resources like thesaurus for identification of error variants. It also does not expend any knowledge about the language except that the word delimiter is blank space. We have tested our approach on erroneous Bangla (Bengali in English) and Hindi FIRE collections, and also on TREC Legal IIT CDIP and TREC 5 Confusion track English corpora. Our proposed approach has achieved statistically significant improvements over the state-of-the-art baselines on most of the datasets.

Keywords: Information Retrieval; OCR error; Word co-occurrence

[*] Email addresses (in the same order as names): kripa.ghosh@gmail.com, chakraborty.abhi89@gmail.com, swapan.parui@gmail.com, prasenjit.majumder@gmail.com,

[1] Tel : +91-9831704289, Fax : +91-325773035

[2] Tel : +91-9712660746, Fax : +91-07930520010

1. Introduction

Text collections containing OCRed errors have presented challenges to researchers in Information Retrieval. Here, we term these collections as erroneous text collections. Researchers have applied different error modelling and correcting techniques on such collections. These techniques comprise training models on sample pairs of correct and erroneous variants. But such an exercise is possible only when the training samples are available. There are many text collections which are created directly by scanning hard copies and OCRing them. We are in an age of digitization. A large number of hard-copied documents have been scanned and archived online. The Million Book Project[3] [4] [5] was a book digitization project led by Carnegie Mellon University School of Computer Science and University Library. It was designed on scanning and OCRing books of different languages. By December 2007, more than 1.5 million books were scanned in 20 languages; mostly in Chinese, English, Telugu and Arabic. Another class of vital documents comprise the legal documents. Millions of court documents, defense documents, proprietary and legacy documents are in hard-copy format. The number of such documents is alarming in countries like India where they are in several Indian languages. Scanning, OCRing and archiving such volumes of documents are a great challenge itself. Information retrieval from such collections often offers further challenges since OCRs in Indian languages are not well-developed. Moreover, the print quality, font diversity and several other features of the hard-copies contribute heavily to the low quality of the scanned documents. Therefore, the clean, error-free version of such a collection is not available to be used for training purpose. Hence, error modelling on such datasets would require manual creation of the error-free version. ACM SIGIR Digital Museum[6] has archived lecture notes of IR stalwarts like Cyril W. Cleverdon, Gerard Salton, Joseph John Rocchio, K. Sparck Jones, et al. as pdf versions created by scanning the original hard-copies of the same. This collection also lacks the original text version. OCRing of this collection is likely to generate erroneous texts which have to be corrected without the error-free version being available. The Illinois Institute of Technology Complex Document Information Processing Test Collection (Oard et al. (2010)) (Grossman & Cormack (2011)), version 1.0, referred to here as "IIT CDIP" and informally in the TREC community as the "tobacco collection" was created for the TREC Legal task. IIT CDIP consists of 6,910,192 document records in the form of XML elements. The text version was created by OCR from the original document images. However, it contains wide-spread OCR errors which made it very difficult for the participants to achieve a meaningful performance on it. The size and the presence of OCR errors discouraged participation in the Legal task to an extent that the organizers decided not to use it further. The IIT CDIP corpus is another such erroneous document collection for which the clean error-free version is not available for error modelling.

[3] `http://en.wikipedia.org/wiki/Million\_Book\_Project` as seen on 28th June, 2015

[4] http://medlibrary.org/medwiki/Million_Book_Project

[5] `http://www.absoluteastronomy.com/topics/Indian_Institute_of_Information_Technology,_Allahabad`

[6] www.sigir.org/museum/allcontents.html

The absence of training data presents a different problem premise, namely, improving the information retrieval performance on OCRed text in the absence of clean text ground truth. To the best of our knowledge, the first such endeavour to address the problem was done by Ghosh et al. (Ghosh & Chakraborty (2012)) in FIRE 2012 RISOT track. They proposed an algorithm based on word similarity and contextual information. A string matching technique (e.g., edit distance, n-gram overlaps, etc.) alone is not sufficient in locating the erroneous variants of an error-free word due to homonymy. For example, word pairs like *industrial* and *industrious*, *kashmir* (place) and *kashmira* (name), etc. have very high string similarity and yet they are semantically unrelated. Such mistakes are even more likely when we do not have a parallel error-free text collection to match the erroneous variants with the correct ones using the common context. However, contextual information can be used to get more reliable groups of erroneous variants. Contextual information can be harnessed effectively by word co-occurrence. We say that two words co-occur if they occur in a window of some size. Word co-occurrence has been used successfully in identifying better stems (Paik et al. (2011a), Paik et al. (2011b)) than methods that use string similarity alone (Majumder et al. (2007)).

The goal of this paper is to propose a novel method to identify erroneous variants in the noisy corpus, to be used for query expansion, in the absence of clean text. Our proposed method uses string similarity measures (*Longest Common Subsequence*, *edit distance*, *Jaccard similarity* of overlapping word n-grams) and contextual information of terms – a combination that effectively produces a performance superior to the state-of-the-art baselines.

The rest of the paper is organized as follows:

In section 2, we discuss the related works. In section 3, we describe our method. We present the results in section 4 and conclude in section 5.

2. Related Work

Studies on retrieval from OCRed text are available in the literature. Among the earliest works, Taghva et al. (1994) applied probabilistic IR on OCRed text. Here, error correction was done using a domain-specific dictionary. The misspelt words were clustered around correctly spelt words which were identified using the dictionary. If the misspelt words in a cluster were close to more than one correctly-spelt word, the error patterns of OCR used were analyzed. A. Singhal et al. (Singhal et al. (1996)) showed that linear document normalization models were better suited to collections containing OCR errors than the quadratic (cosine normalization) models. TREC made a significant effort on the study and effect of OCR errors in retrieval in their two tasks : the Confusion Track and the Legal Track. The TREC Confusion track was a part of TREC 4 (1995) (Harman (1995)) and TREC 5 (1996) (Kantor & Voorhees (1996a)). In TREC 4 Confusion Track, random character insertions, deletions and substitutions were used to model degradations. Such degradations were done on 260,000 English electronic text documents from multiple sources. For the TREC 5 Confusion Track, 55,000 government announcement documents were printed, scanned, OCRed and were then used. Electronic text for the same documents was available for comparison. Participants experimented with techniques that used error modelling to alleviate OCR errors using character n-gram matches. Parapar et al. (Parapar et al. (2009)) combined the results of 5-, 4-, 3- and 2-grams with the actual terms. Four indices were created corresponding to each of the n-grams. One index was created for the actual term. The final score was a weighted linear combination of the scores obtained from these five indices separately. The parameters were trained on the TREC Confusion Track collection and tested on the TREC legal IIT CDIP 1.0 dataset. The method failed to produce significant improvements over the baseline in MAP. Moreover, they did not consider any recall specific evaluation measure which is vital for legal IR. Also, a major drawback with the n-gram is the notable inflation in the size of the inverted index (Paik & Parui (2011)). A passage of k characters contains $(k - n + 1)$ n-grams of length n, but only approximately $(k + 1)/(l + 1)$ words, where l is the average word length for the language. Consequently, there is a marked increase in query processing time when n-grams are used: retrieval with 4-gram is 10 times slower than plain word retrieval. Moreover, for a large and noisy collection like TREC legal IIT CDIP 1.0, the inflation of the whole index can be of concern, when the n-gram version of the whole collection is to be produced. This poses a great resource crisis in the storage of and retrieval from the resulting collections.

A similar track, RISOT (Garain et al. (2013)), was offered in Forum for Information Retrieval Evaluation[7] (FIRE) 2011. This was aimed at improving retrieval performance from OCRed text in Indic script. In 2011, a FIRE Bangla collection of 62,825 documents was available as the "TEXT" or "clean" collection from a leading Bangla newspaper, Anandabazar Patrika. Each document of the collection was scanned at a resolution of 300 dots per inch. Then, each scanned document was converted to electronic text using a Bangla OCR system that had about 92.5% accuracy. Ghosh et al. (Ghosh & Parui (2013)) performed a two-fold error modelling technique for OCR errors in Bangla script. In 2012 RISOT track, in addition to the Bangla collection pair, a Hindi collection pair was also offered. The error-free Hindi document collection was created from a leading Hindi newspaper Dainik Jagaran. The OCRed Hindi collection was created using a Hindi OCR system. Ghosh et al. (Ghosh & Chakraborty (2012)) tried to address the problem of improving retrieval effectiveness from noisy collection in the absence of the clean text version in FIRE RISOT task 2012 on Bangla dataset. Recently, Chakraborty et al. (2014) produced further improvement in performance on this problem. They applied query-specific clustering of the erroneous term variants based on co-occurrence and Pointwise Mutual Information (Kang & Choi (1997)). More recently, Chakraborty et al. (2015) produced a significant improvement in Recall over Chakraborty et al. (2014) on the TREC legal IIT CDIP 1.0 dataset.

However, one can find substantial work in the literature on OCR error modelling and correction. Kolak and Resnik (Kolak & Resnik (2002)) applied a pattern recognition approach to detecting OCR errors. Magdy & Darwish (2006) used Character Segment Correction, Language modelling, and Shallow Morphology techniques in error correction on OCRed Arabic texts. On error detection and correction of Indic scripts, B.B. Chaudhuri and U. Pal produced the very first report in 1996 (Chaudhuri & Pal (1996)). This paper used mor-

[7] www.isical.ac.in/∼fire

phological parsing to detect and correct OCR errors. Separate lexicons of root-words and suffixes were used. Fataicha et al. (Fataicha et al. (2006)) (Fataicha et al. (2002)) located confused characters in erroneous words and created a collection of erroneous error-grams. Finally, they generated additional query terms, identified appropriate matching terms, and determined the degree of relevance of the retrieved document images to the user's query, based on a vector space IR model. Confused characters in erroneous words were also used by Marinai (2009). Reynaert (Reynaert (2014)) developed an online processing system for post-processing of OCR errors. It first derives the alphabet for the language from an appropriate source. Then, the valid characters for a given language are retained. Then, the list of all possible character confusions are produced according to a given threshold for Levenshtein distance. Choudhury et al. (2007) explored the challenges in developing a spell-checker orthographic proximity between two words for Bengali, English and Hindi.

3. Our approach

In this section we describe our proposed approach. Most of the prior work on the identification of erroneous variants of a word have relied on string similarity alone. In this work, we have combined contextual information with a string similarity measure to get more reliable erroneous variants. Before going into the approach, we describe the key concepts used in the approach in the following subsection. Then, we describe the approach in detail.

3.1. Key Terms

3.1.1. Word Cooccurrence

We say that two words w_1 and w_2 co-occur if they appear in a window of size s ($s > 0$) words in the same document d. Suppose, we say that the words w_1 and w_2 co-occur in a window of size 5 in a document d. This means that there is at least one instance in the document where w_1 is followed by at most 4 words (distinct from w_1 and w_2) and then followed by w_2, or w_2 is followed by at most 4 words (distinct from w_1 and w_2) and then followed by w_1. Let $cooccurFreq_{(d,s)}(w_1, w_2)$ denote the number of instances w_1 and w_2 co-occur in d in a window of size s. Then, we call $cooccurFreq_{(d,s)}(w_1, w_2)$ the co-occurrence frequency of w_1 and w_2 in document d for a window of size s. However, it is a common practice to compute $cooccurFreq_{(d,s)}(w_1, w_2)$ over all the documents in a collection. This is likely to give a more robust measure of co-location of the words w_1 and w_2.

Word co-occurrence gives a reliable measure of association between words as it reflects the degree of context match between the words. Usually, the total co-occurrence between word pairs is calculated over a collection of documents by summing up the document-wise co-occurrence frequencies. High co-occurrence of a pair of words is an indicator of high degree of relatedness of the two words. This association measure gets more strength when it is used in conjunction with a string matching measure. For example, two words sharing a long stem (prefix) is likely to be variants of each other if they share the same context as indicated by a high co-occurrence value between them. The word *industrious* shares a stem "industri" with the word *industrial*. But, they are not variants of each other. They can be easily segregated by examining their context match as they are unlikely to have a high co-occurrence frequency. In this paper, we have used co-occurrence information along with a string similarity measure (LCS, ES and overlapping n-gram based Jaccard, discussed in the following subsection) to identify erroneous variants of query terms.

3.1.2. Word Similarity Measures

We have considered three string similarity measures for our problem. They are described below.

- **Longest Common Subsequence (LCS) similarity** :

  Given a sequence $X = \langle x_1, x_2,....,x_m\rangle$, a sequence $Z = \langle z_1, z_2,....,z_k\rangle$ $(k \leq m)$ is called a *subsequence* of X if there exists a strictly increasing sequence $\langle i_1, i_2,....,i_k\rangle$ of indices of X such that for all j = 1,2,...,k, we have $x_{i_j} = z_j$. Now, given two sequences X and Y, we say that Z is a *common subsequence* of X and Y if Z is a subsequence of both X and Y. A *common subsequence* of X and Y that has the longest possible length is called a *longest common subsequence* or LCS of X and Y. For example, let $X = \langle A, B, C, B, D, A, B\rangle$ and $Y = \langle B, D, C, A, B, A\rangle$. Then, the sequence $\langle B, D, A, B\rangle$ is an LCS of X and Y (Cormen et al. (2009)). Note that LCS of X and Y is not in general unique.

  In our problem, we consider sequences of characters, or strings. For strings *industry* and *industrial*, the LCS is *industr*. Now, we define a similarity measure between two words as follows :

  $$LCS\_similarity(w_1, w_2) = \frac{StringLength(LCS(w_1,w_2))}{Maximum(StringLength(w_1),StringLength(w_2))}$$

  So, $LCS\_similarity(industry, industrial)$
  $$= \frac{StringLength(industr)}{Maximum(8,10)} = 0.7$$

  Note that the value of LCS_similarity lies in the interval [0,1].

- **Edit Distance based similarity** :

  Edit distance is a popular measure for measuring distance between two strings. Edit distance between two words is the minimum number of single character edits, i.e., insertions, deletions or substitutions, required to change one word into the other[8].

[8] http://en.wikipedia.org/wiki/Levenshtein_distance as seen on 25th June, 2015

Let us denote edit distance by ED. Edit similarity (ES) between two words w_1 and w_2 is defined as

$ES(w_1, w_2) = 1$ - $\frac{ED(w_1,w_2)}{Maximum(StringLength(w_1),StringLength(w_2)}$.

Let us consider the example of *industry* and *industrial.* Note that the minimum number of edits to get *industrial* from *industry* is 3. So, $ED($*industry*, *industrial*$) = 3$ and $ES($*industry*, *industrial*$) = 0.7$. Note that the value of ES also lies in the interval [0,1].

- **Jaccard similarity**

Jaccard similarity (JC) between two finite sets A and B is defined as :

$JC(A, B) = \frac{s(A \cap B)}{s(A \cup B)}$,

where $s(E)$ denotes the number of elements in E. So, JC lies in the interval [0,1].

For two words w_1 and w_2, the sets A and B are formed by the *overlapping n-grams* of the words. The value of n in an n-gram is a positive integer. However, it is chosen to be greater than 1, because 1-gram overlaps fail to capture the order of characters in a word.

Now, for the strings *industry* and *industrial* the overlapping 2-grams are respectively given by $A = \{in, nd, du, us, st, tr, ry\}$ and $B = \{in, nd, du, us, st, tr, ri, ia, al\}$. Then, $JC(A, B) = \frac{s(\{in,nd,du,us,st,tr\})}{s(\{in,nd,du,us,st,tr,ry,ri,ia,al\})} = 0.6$.

For these two strings, the JC value for overlapping 3-grams is 0.556 and the JC value for overlapping 4-grams is 0.5.

The value of n can be difficult to choose especially for very short strings. Consider, two strings *cat* and *cut.* For $n = 1$, JC(*cat*, *cut*) $= 2/4 = 0.5$. However, for $n > 1$, there is no overlap between the strings. LCS_similarity and ES are less sensitive to small differences between the strings than the n-gram overlap measure. n-gram based schemes have been used by the TREC Confusion task participants and in a work by Parapar et al. (Parapar et al. (2009)) discussed in the Related Works section of our paper. These works involve tokenization of the whole document collection into n-grams, indexing and retrieval from the n-gram collections. This inflates the inverted index to a great extent and slows down retrieval considerably. Note that we use n-gram based Jaccard overlap for the *Segregation* sub-step only for filtering likely error variants of a query term; and finally for the *Melding* step (both to be discussed shortly). Our approach does not involve indexing the whole document collection as n-grams or retrieval from n-gram inverted index. We compute the Jaccard overlap between two words whose n-gram representations are created promptly on the fly. For a given pair of words, JC is comparable to LCS and ES in time requirement. This makes our n-gram approach much faster than the previous efforts.

3.2. Our proposed approach

The goal of our approach is to obtain a set of variants of a given query term. Thus, based on this, for a given query one can obtain an expanded query that is expected to produce a better retrieval performance than the original query. Our approach has two major parts :

1. Agglomeration, and
2. Melding

3.2.1. Agglomeration

This part of our approach can be divided into the following sub-parts:

- **Segregation** :

  Let $\mathbb{D}$ denote the document collection and let L be the lexicon or the set of all unique words in the documents in $\mathbb{D}$. Let $q \in \mathbb{Q}$ be a query such that $q = \{w_1, w_2,...,w_n\}$, where w_i, $i \in \{1,2,...,n\}$, is a query term. We construct a set $L^{\alpha}_{w_i} = \{w \in L : string\_similarity(w, w_i) > \alpha\}$ where α is a threshold value lying in the interval $(0, 1)$. In other words, $L^{\alpha}_{w_i}$ contains all the words in L that has *string_similarity* of more than α with the query term w_i. Here *string_similarity* refers one of the aforementioned similarity measures : *LCS_similarity*, ES and JCn $(n > 1)$.

- **Graph formation** :

  We now define a graph G on the set $L^{\alpha}_{w_i}$ of words. Let $G = (V, E)$ be a graph where V, the set of vertices, is $L^{\alpha}_{w_i}$ and E is the set of edges where every pair of words w_1 and w_2 in V that co-occur in a document, defines an edge. The weight of the edge defined by w_1 and w_2, is given by the co-occurrence frequency of w_1 and w_2 in the collection.

- **Pruning** : Let the maximum edge weight of the graph G be max_{ew}. Then, we eliminate those edges in G whose weight is less than β per cent of max_{ew}. Let this new graph be G_r. Then $G_r = (V, E_r)$ where $E_r = \{e \in E :$ weight of $e \geq \beta$ % of $max_{ew}\}$ and β lies in the interval [0,100]. This step is taken to eliminate the chance co-occurrences of words which are otherwise unrelated but happen to occur together in the same document by chance. The frequencies of such co-occurrences are very small. An example of such a situation can be - "a lady named *Kashmira* visited the place *Kashmir* to spend her summer vacation". Note that the words *Kashmira* and *Kashmir* are not semantic variants of each other.

  However, pruning is a risky step that can wrongly separate close members. In addition, it may be more risky if terms are rare in the corpus. For example, let a word w have document frequency (i.e., the number of documents the word has occurred in

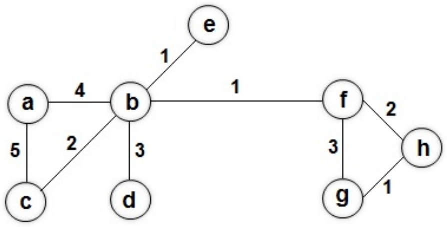

Figure 1: Graph : before clustering

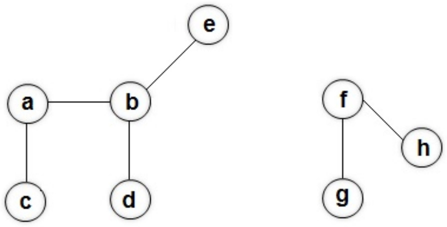

Figure 2: Graph : after clustering

the collection), *df* as 11 and so be the maximum edge-weight in the graph. Then, the nodes with edge weight 1, may actually be semantic variants of w but will be removed if pruning is done at 10%. This may happen for Bangla where rare compound characters may be part of the query. Rare terms and their variants ought to be in the same cluster for effective retrieval. Separation of these variants is not expected. So, pruning is carried out only if the $max(df)$ of the nodes of the connected sub-graph is greater than γ.

- **Congregation** : We now cluster the vertices of graph G_r based on the edge weights. We say that v_1 is the *strongest neighbour* of v_2 if of all the neighbouring (adjacent) vertices of v_2, the weight of the edge joining v_1 and v_2 is the maximum. Our clustering algorithm is as follows :

  Two vertices v_1 and v_2 will belong to the same cluster if

  - either v_1 is the *strongest neighbour* of v_2
  - or v_2 is the *strongest neighbour* of v_1

The clustering algorithm is illustrated in Figures 1 and 2. Figure 1 shows the graph before clustering algorithm is applied. The vertices correspond to words and the edge weights correspond to the co-occurrence values between the words. Figure 2 shows the clustered graph. The connected components of the clustered graph are the resulting clusters. Vertex c is the strongest neighbour of a which is in turn the strongest neighbour of b. b is the only neighbour (and hence the strongest member) of both d and e. But neither b is the strongest neighbour of f nor f is the strongest neighbour of b. Thus, we get a connected component containing the nodes a, b, c, d and e. Similarly, we get the other connected component containing f, g and h.

This clustering algorithm was used in Paik et al. (2011b). It is more convenient to use this algorithm than the popular clustering algorithms like single-linkage, complete-linkage and k-means algorithm since it is parameter-free. In the hierarchical algorithms like single-linkage and complete-linkage algorithms, the clusters depend heavily on the cut-off threshold. On the other hand, in partitional clustering algorithms like k-means algorithm, the value of k plays a pivotal role in clustering. Such parameters have to be chosen judiciously and there is no standard way of choosing them. So, we decided to use this simple and effective clustering method in our work.

3.2.2. Melding

Now, given a query term w_i, we need to find erroneous variants from the OCRed corpus. Let $\mathbb{C} = \{Cl_1, Cl_2,...,Cl_k\}$ be the set of all clusters formed from $L^{\alpha}_{w_i}$ (lexicon of the erroneous corpus) by the clustering algorithm discussed in the last subsection. So, each cluster Cl_j is of the form $\{w_{j_1}, w_{j_2},..,w_{j_m}\}$, where w_{j_t} is a word in cluster Cl_j. In the clusters of $\mathbb{C}$, we look for the word that has the maximum string similarity with w_i. In other words, let $w_{closest}$ in $L^{\alpha}_{w_i}$ be the word such that $string\_similarity(w_{closest}, w_i) > string\_similarity(w_t, w_i)$, for all w_t in $L^{\alpha}_{w_i} - \{w_{closest}\}$. Let $Cl_{closest} \in \mathbb{C}$ be the cluster containing $w_{closest}$. Then, we choose all the words in $Cl_{closest}$ as the erroneous variants of w_i. If there are more than one such $w_{closest}$ having maximum similarity with w_i leading to more than one $Cl_{closest}$, we do not choose any cluster. This is because if the variants of w_i are taken from more than one $Cl_{closest}$, the resulting expanded query may lead to poor retrieval results.

Our proposed method in algorithmic form is shown below. A pictorial view of the proposed method is given in Figure 3. For a *Query Term*, we get a *Subset of Lexicon* after filtering out words from *Lexicon* based on the α threshold. *Co-occurrence Block* represents the repository of all the co-occurrence information for all the word pairs in the document collection. For the *Subset of Lexicon*, the *Co-occurrence Block* is used to read the co-occurrence values for this subset of words and form the *Graph*. Next, the *Graph* is pruned using the β threshold and we get the *Pruned Graph*. Then, *Pruned Graph* is clustered to get the *Clusters*. Now, for the *Query Term*, we choose the appropriate cluster from the *Clusters*. We call this process *melding*. The chosen cluster (if any cluster is chosen), along with the *Query Term*, forms the *Expanded Query*. The *Expanded Query* is then used for retrieval.

4. Results

4.1. Dataset

We tested our approach on FIRE RISOT[9] Bangla and Hindi collections. The collection statistics are given in Table 1. *Bangla original* is the “clean” or error-free version created

[9] http://www.isical.ac.in/∼ fire/data.html

Algorithm 1: : Error correction algorithm

```
Let $\mathbb{Q}$ be the set of all query terms, $\mathbb{D}$ be the collection of all the documents and $\mathbb{L}$ be the lexicon of $\mathbb{D}$
for each word $w$ in $\mathbb{Q}$ do
    /* Segregation */
    Let $L_w^\alpha$ be the set of words in $\mathbb{L}$ whose $string\_similarity$ with $w > \alpha$ $(0 < \alpha < 1)$
    /* Graph formation */
    Let $G = (V, E)$ be a graph defined on $L_w^\alpha$ such that the vertices in $V$ correspond to the words in $L_w^\alpha$, and there is an edge in $E$ between two vertices if the corresponding words in $L_w^\alpha$ co-occur and the weight of the edge is the corresponding co-occurrence frequency
    /* Pruning */
    If $max(df)$ of the nodes in $G > \gamma$, construct a graph $G_r = (V, E_r)$, where $E_r$ contains only those edges in $E$ whose edge weight $> \beta$ per cent of the maximum edge weight in $E$
    /* Congregation */
    Cluster the vertices in $G_r$ such that two vertices $v_1, v_2 \in V$ belong to the same cluster if either $v_1$ is the strongest neighbour of $v_2$ or $v_2$ is the strongest neighbour of $v_1$
    /* Melding */
    Let $\mathbb{C} = \{Cl_1, Cl_2, ..., Cl_k\}$ be the set of clusters formed; choose the cluster that contains the word that has maximum $string\_similarity$ with $w$; let the number of such clusters be $no\_cl$
    if $no\_cl == 1$ then
        choose all the words in the chosen cluster as erroneous variants of $w$
    else
        IGNORE
    end if
end for
```

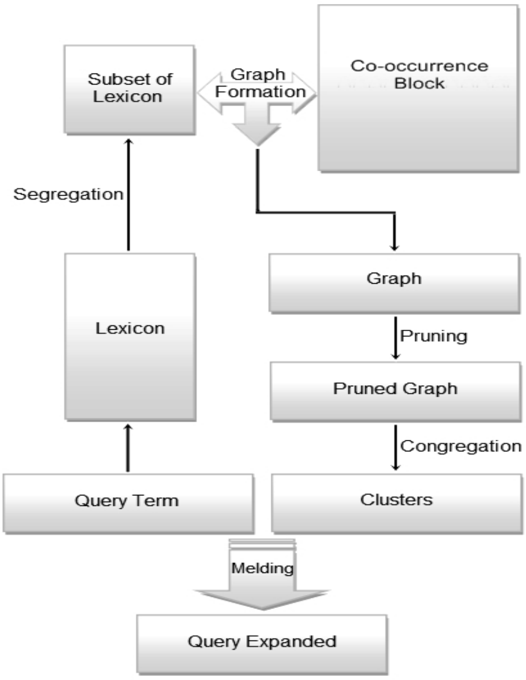

Figure 3: Our proposed method : a pictorial view

from Anandabazar Patrika. *Bangla OCRed* is the scanned-and-OCRed version of the same. Similarly, *Hindi original* is the error-free version and *Hindi OCRed* is its OCRed version. A document in the original version and its OCRed version had the same unique document identification string so that the original-OCRed pairs can be easily identified. We can see that, for both Bangla and Hindi, the original version contains more documents than its OCRed version. So, the extra documents in the original were not used for comparison. But, despite having fewer documents, we can see that the OCRed collections contain more unique terms than their error-free counterparts. The number of unique terms in Bangla original corpus is 396968 while the same number in its OCRed version is 466867. Similarly, the number of unique terms in Hindi original corpus and its OCRed version are 242047 and 264240 respectively. This discrepancy is caused by OCR errors. Most of the inflations are caused by misrecognition (as multiple candidates). We will have a more detailed discussion on this issue in a subsequent section. The Bangla collection has 66 topics and the Hindi collection has 28 topics. These topics were created for previous FIRE Ad Hoc tasks. A subset of the Ad Hoc topics was selected for the RISOT task. The third dataset which we used is the TREC Legal IIT CDIP collection (Oard et al. (2010)). Unlike the Bangla and Hindi collections, the IIT CDIP collection lacks the clean-text original version which is useful in error-modelling. So, this legal collection provided the best use case scenario for our approach. We have also tested our proposed method on TREC 5 Confusion track (Kantor & Voorhees (1996b)) datasets, which consist of three versions – one clean error-free version (*original*) and two erroneous versions with estimated character error rates of 5% and

Dataset	No. of documents	No. of topics	No. of unique terms
Bangla original	62838	66	396968
Bangla OCRed	62825	66	466867
Hindi original	107696	28	242047
Hindi OCRed	94432	28	264240
IIT CDIP	6910192	43	135985661
TREC 5 Confusion (original)	55600	49	262597
TREC 5 Confusion (5%)	55600	49	313558
TREC 5 Confusion (20%)	55600	49	2532386

Table 1: Collection statistics

20%, produced from the clean version. We see that the number of unique terms in the 5% collection is more than that in the *original*, and this number is considerably higher in the 20% version. This is due to the presence of erroneous variants of terms in the two noisy collections.

4.2. Evaluation

The results are shown in Tables 2 and 3. *No Expansion* is the retrieval performance on the OCRed collection. *RISOT2012* denotes the results produced by the algorithm reported by Ghosh et al. (Ghosh & Chakraborty (2012)). *KDIR_Cooccurrence* and *KDIR_PMI* denote the results produced by the method proposed by Chakraborty et al. (Chakraborty et al. (2014)) using cooccurrence and PMI respectively. *LCS* is the result when our proposed method is applied when the similarity measure used is *LCS_similarity*. *ES* is the result when the similarity measure used is Edit Similarity. *JC2*, *JC3* and *JC4* are the results when *JC* is used with overlapping 2-, 3- and 4-grams respectively.

SIGIR2015 (Chakraborty et al. (2015)) employed a graph in-degree based weighting scheme on the expanded terms. The weight on a node (i.e., a term) is the sum of its total degree (number of edges connected to the node). The weights were then normalized so that them add up to 1. It used LCS based string similarity measure for selection of potential error variants and for connecting them to the corresponding term. This algorithm was applied on the TREC Legal IIT CDIP dataset (Oard et al. (2010)). We have run this algorithm on Bangla and Hindi datasets also for comparison with our proposed method.

We see that among all the similarity measures used, *ES* produces the best performance for both the data sets. Table 2 shows percentage improvements over *No Expansion*. The numerical difference between our method using *ES* and *No Expansion*, *RISOT2012*, *KDIR_Cooccurrence* and *KDIR_PMI* in Bangla and Hindi was found to be *statistically significant* at 95% confidence level (p-value < 0.05) by Wilcoxon signed-rank test (Siegel (1956)). Our proposed method (using *ES*) is also numerically better than *SIGIR2015* in Bangla and the difference is *statistically significant* at 95% confidence level (p-value < 0.05) by Wilcoxon signed-rank test. However, our proposed method (using *ES*) is numerically comparable with *SIGIR2015* in Hindi and the difference is not *statistically significant* at 95% confidence level (p-value $>$ 0.05) by Wilcoxon signed-rank test.

For comparison on the IIT CDIP dataset, we have reported values of MAP and Recall@k (k = 100, 500 and 1000) for *ES* only. These results are shown in Table 3. Recall values were considered because this is a domain-specific collection, where recall is of primary importance. Our proposed method numerically outperforms all the baselines (*No Expansion*, *RISOT2012*, *KDIR* versions and *SIGIR2015*) on all the evaluation measures. Except in Recall@100 with *SIGIR2015*, these differences are *statistically significant* at 95% confidence level (p-value $<$ 0.05) by Wilcoxon signed-rank test, over all the evaluation measures.

We have also applied our proposed method on TREC 5 Confusion track (Kantor & Voorhees (1996b)) where the approach is tested in the presence of 5% and 20% error. Table 4 shows the results in Mean Reciprocal Rank (MRR) – the official evaluation measure used in the track. Our proposed method numerically outperforms both *No Expansion* and *SIGIR2015* and the differences were found to be *statistically significant* at 95% confidence level (p-value $<$ 0.05) by Wilcoxon signed-rank test. We have compared our proposed method with only *SIGIR2015* since the latter has been the closest competitor of our proposed method. However, *Original*, which is the retrieval performance on the clean text version, has a much superior value. This indicates that there is still room for improvement in our proposed method.

Original shows the performance in Table 2 when retrieved from the error-free version. This value is shown as an upper bound that can be achieved if all the errors are successfully identified. Note that our method does not use the error-free version of the corpus to model the errors in the OCRed version.

However, we have compared our method with the one reported in (Ghosh & Parui (2013)) which used the original error-free corpus for learning the error pattern in the OCRed version. This result is shown in the table as *Expansion Original*. In this method, given an OCRed document, the corresponding original document was considered and one-to-one mapping was determined between the words of these two documents. The mismatching words were stored and this was done for the whole collection. These words were broken down to symbols and a symbol-level mapping was determined between a word in the original document and its probable erroneous variants in the OCRed document. Here, the query words were expanded to include the corresponding variants. Note that the method in (Ghosh & Parui (2013)) uses information about the language like symbol set, nature of compound characters in Bangla script. Our method, on the other hand, uses no such information. Despite this, we see that our best result is numerically comparable with *Expansion Original* for both the datasets and the differences are *not statistically significant* at 95% confidence level (p-value $>$ 0.05) by Wilcoxon signed-rank test.

Note that there are no entries for *Original* and *Expansion Original* in Table 3, since there is no clean-text original collection available for the IIT CDIP collection.

4.2.1. Parameters

Our approach has three parameters α, β and γ. The values of α and β were determined using grid search in the interval (0,1). The values were found at step-lengths of 0.01. The

	Bangla	Hindi
No Expansion	0.1791	0.1468
RISOT2012	0.1974	0.1480
KDIR_Cooccurrence	0.2067	0.1501
KDIR_PMI	0.2060	0.1495
SIGIR2015	0.2141	**0.1685**
LCS	0.2216 (+23.73%)	0.1634 (+11.31%)
ES	**0.2231** (+24.57%)	0.1672 (+13.9%)
JC2	0.2176(+21.5%)	0.1574 (+7.22%)
JC3	0.2154(+20.27%)	0.1552 (+5.72%)
JC4	0.2103(+17.42%)	0.1577 (+7.43%)
Original	0.2567	0.2551
Expansion Original	0.2245	0.1614

Table 2: MAP values on Bangla and Hindi. The best values are shown in bold. The percentage improvements over No Expansion is shown.

	MAP	Recall@100	Recall@500	Recall@1000
No Expansion	0.0899	0.1574	0.3112	0.3807
RISOT2012	0.0885	0.1615	0.3011	0.3756
KDIR_Cooccurrence	0.0898	0.1658	0.3072	0.3773
KDIR_PMI	0.0897	0.1643	0.3065	0.3721
SIGIR2015	0.0947	0.1741	0.3151	0.3828
Proposed	**0.1011** (+12.46%)	**0.1786** (+10.61%)	**0.3384**(8.74%)	**0.4056**(+6.54%)

Table 3: Results on IIT CDIP 1.0. The best results are shown in bold. The percentage improvements over No Expansion is shown.

value of γ was empirically chosen as 50.

4.3. Analysis

Table 5 shows some query terms and error variants in the Bangla corpus identified by our proposed method as well as the error variants identified by using error modelling in *Expansion Original* (Ghosh & Parui (2013)). The first query term is সিঙ্গুরে, i.e., *singure* (Transliterated) which means "in Singur", Singur being the name of a place. Our proposed method identifies the variants সিঙ্গর (*singor*), সিঙ্গরে (*singore*) and সিঙ্গরের (*singorer*). Error modelling finds the variant সিঙ্গরে (*singore*) only. Here, সিঙ্গরে (*singore*) is an erroneous variant of সিঙ্গুরে (*singure*) whereas সিঙ্গর (*singor*) and সিঙ্গরের (*singorer*) are inflectional variants of সিঙ্গরে (*singore*). This shows that for query term সিঙ্গুরে (*singure*), our proposed method not only finds an erroneous variant but also the inflectional variants of the erroneous variant. So, our proposed method plays a two-fold role in improving the retrieval performance. The next query term is সুনামি,

	5%	20%
No Expansion	0.5955	0.3284
SIGIR2015	0.6198	0.4415
Proposed	**0.6446** (+8.25%)	**0.4619** (+40.65%)
Original	0.7653	0.7653

Table 4: MRR values on TREC 5 Confusion Track dataset. The best results are shown in bold. The percentage improvements over No Expansion are shown. Original is the MRR value when the same set of topics are run on the clean, error-free version, which serves as an upper-bound of performance.

Query term	Proposed	Error modelling
সিঙ্গুরে (*singure*) (in Singur)	সিঙ্গর (*singor*), সিঙ্গরে (*singore*), সিঙ্গরের (*singorer*)	সিঙ্গরে (*singore*)
সুনামি (*sunami*) (Tsunami)	সৃনামি (*srinami*)	2নামি (*2nami*)
ব্রিটেনে (*britaine*) (in Britain)	-	ব্রটেনে (*brotaine*)
বাজেট (*budget*) (budget)	বাজেটে (*budgete*), বাজেঢ (*budgedh*)	-

Table 5: Terms identified (Bangla) : proposed vs error modelling

Query term	Proposed	Error modelling
पदार्थो (*padaartho*) (materials)	पदाथों (*padaathon*), पदार्थ (*padaarth*)	पदर्थो (*padartho*)
अपहरण (*apaharan*) (kidnapping)	अपहरणर् (*apaharanar*), अपहरणोा (*apaharanoa*)	-
साइबर (*saaiber*) (cyber)	साहबर (*saahber*)	सइबर (*saiber*)
मंडल (*mandal*) (Mondal)	मंगल (*mangal*), मंडेल (*mandel*)	हडल (*hadal*)

Table 6: Terms identified (Hindi) : proposed vs error modelling

Query term	Corresponding word in the corpus
खगोलीय (*khagoliya*) (astronomical)	खग (*khag*)
मूल्य (*mulya*) (price)	पूछ (*puuchh*), पूल्स (*puuls*), पूजा (*puuja*)
भ्रुण (*bhrun*) (foetus)	भन (*bhan*), .पूण (*.pun*)

Table 7: Hindi errors

Term	Error variants
potassium	otassium, patassium, polassium, potaasium, potaesium potasaium, potaseium, potasium, potassi, potassiium, potassiu
pathology	aetiology, patholo, patholog, pathologic, pathologie, thology
yield	jield, kield
public	pubiic, publi, publicly, publie, publlc, puhlic, ublic

Table 8: Error variants produced by our proposed method : IIT CDIP 1.0

Term	Error variants
department	bepartmeat, departmeat, nepartmeat, oepartmeat, oepartment
education	educatioa, educationu
indian	fndian, ndian
universities	tniversities, univerdties, univerfities, uuversities
technicians	techacians, techaician, techaicians, techrician, techucians, techuciansg
communities	comm0nities, commanities, commenities, commuaities, communitie, ommunities

Table 9: Error variants produced by our proposed method : TREC-5 confusion track

i.e., *tsunami*, the devastating sea waves Tsunami. Our proposed method identifies the variant সৃনামি (*srinami*). On the other hand, error modelling finds another variant 2নামি (*2nami*). This is an interesting comparison as we see that the two methods identify two different variants of the query term. For the term ব্রিটেনে (*britaine*), meaning "in Britain", our proposed method fails to identify any variants, while the error modelling identifies one variant, viz., ব্রটেনে (*brotaine*). However, for the query term বাজেট (*budget*), the error modelling fails to find any variant, whereas our proposed method finds variants বাজেটে (*budgete*) and বাজেঢ (*budgedh*).

Now we look at some Hindi query terms and their erroneous variants produced by our proposed method and the error modelling method. These are shown in Table 6. The first query term is पदार्थो (*padaartho*) which means "materials". Our proposed method finds the variants पदाथों (*padaathon*) and पदार्थ (*padaarth*). On the other hand, the error modelling finds पदर्थो (*padartho*). For the term अपहरण (*apaharan*), which means "kidnapping", the error modelling fails to find any variant. However, here our proposed method finds अपहरणर् (*apaharanar*) and अपहरणोा (*apaharanoa*). In the case of साइबर (*saaiber*), which means "cyber", the two methods find two different variants. Our proposed method finds साहबर (*saahber*) while the error modelling finds सइबर (*saiber*). For query term मंडल (*mandal*), a surname, we again got different variants from the two methods. While our proposed method identifies मंगल (*mangal*) and मंडेल (*mandel*), error modelling finds हडल (*hadal*). Here, मंगल (*mangal*), an output of our proposed method, is a wrong variant in the sense that it also is a valid word.

Table 2 shows that for Bangla, the performance achieved by the error modelling method that makes use of the original text, is somewhat close to the one obtained from the original error-free corpus. Our proposed method also is not too far away. However, for Hindi, the result is quite different. Here, we see that the performances achieved by both the methods are much worse than the one obtained from the original. Also, the error modelling method is numerically worse than our proposed method. This means that the Hindi corpus has far too many non-recoverable errors caused by the OCR. We looked for the query terms in the relevant documents and found that many important query terms were badly garbled up in the Hindi OCRed corpus. Table 7 shows some of the serious misrecognitions. Let us consider some of the adversely affected query terms. The first Hindi query term is खगोलीय (*khagoliya*) which means "astronomical". In most of the relevant documents, this word was curtailed to खग (*khag*), which is a serious error that hurts retrieval badly as *khagoliya* is a key term in the query. The next query term is मूल्य (*mulya*) which means "price". This was recognized as पूछ (*puuchh*), पूल्स (*puuls*) and पूजा (*puuja*) which are radical departures from the actual word. भ्रुण (*bhrun*) (which means "Foetus") was recognized as भन (*bhan*) and .पूण (*.pun*). Note that this is a small list of query terms, which demonstrates serious misrecognitions.

Table 8 shows some error variants identified in the IIT CDIP 1.0 corpus. For the query word *potassium* the variants *otassium*, *patassium*, *polassium* etc. have been identified by our proposed method. Similarly, for *pathology* the correct variants *aetiology*, *patholo*, *patholog* etc. have been identified by our method. Note that there are no variants given by error modelling since, for this corpus, the error-free clean text version is not available. So, this corpus provides an ideal use-case scenario for our proposed method.

Table 9 shows some error variants identified in the TREC-5 confusion track. For the query word *department* the variants *bepartmeat*, *departmeat*, *nepartmeat*, *oepartmeat* and *oepartment* have been identified by our proposed method. Similarly, for *universities* the correct variants *niversities*, *univerdties*, *univerfities* and *uuversities* have been identified. Table 4 shows that the improvement of our proposed method over *No Expansion* is noticable for the 20% version. While investigating the reason, we found that there are many query terms which are often misrecognised in the 20% collection. This has resulted in poor retrieval performance on the 20% collection denoted by *No Expansion* in the table. Our proposed method has identified many authentic erroneous variants of such query terms and this has led to a superior performance. For example, the term *department* has been correctly identified 474 times (collection term frequency); while it has been misrecognised as *bepartmeat* or *departmeat* 5420 times in the 20% corpus. Our proposed method has identified *bepartmeat* or *departmeat* as the variants of *department*. The term *universities* has never been correctly recognised. The error variants *univerfities* and *uuversities* have been identified by our proposed method. Our proposed method has outperformed *SIGIR2015* because the former has identified better error variants. For example, for the query term *congress SIGIR2015* has identified the variants *gongress* and *ongress*. Our proposed method has identified the variants *caongress* and *congresws* along with the variants *gongress* and *ongress*. The high MRR value of the *Original* run (on the clean text), also attests the presence of high recognition errors in the two noisy collections.

The parameter γ has caused differences in performance of our proposed method over the baselines in Bangla and Tobacco corpus. For example, the Bangla compound character ঙ্ঘ (misrecognised as ঙঘ in the noisy corpus) is rare in the Bangla corpus. The word সিঙ্ঘল (Singhal) containing this compound character, has df as 4. Its variants have df as low as 1. Our proposed method agglomerates all these variants in the same cluster unlike SIGIR2015 where the variants get separated across clusters whereas in KDIR and RISOT, semantically unrelated words fall in the same cluster. Similarly, in the Tobacco corpus the term *fluorapatite* has maximum df as 10 and its variant *fluorapatit* has df 1; these are dispersed by SIGIR2015 unlike our proposed method. This has happened for many crucial query terms. Similar is the fate for many rare terms in TREC-5 Confusion track dataset. The term *euthanasia* has df as 3 and its variants *euthanafia* and *euthdnasia* have lower df. In these cases also, our proposed method has scored over all the baselines.

Retrieval is hurt in the tobacco collection due to numerous OCR misrecognitions. Many important query terms have several variations in the corpus. Once identified, most of the variants must be used in retrieval because it is non-trivial to deduce which particular variant would lead to the retrieval of relevant documents. KDIR and RISOT2012 identify many query term variants which are not semantically related like *yield* and *field*, while Table 8 shows that our proposed method has identified the correct variants *jield* and *kield* for the query term *yield*. SIGIR2015 produces more correct variants than KDIR and RISOT2012 due to its restrictive nature. However, inappropriate weight assignment on the variants has a diminishing effect on retrieval result, particularly in the cases where a large number of variants have been identified. For example, for SIGIR2015, in the case of the query term "oncology", the word oncology gets a weight of 1.0 whereas all the other variants have much lower weight like 0.009434:

1.000000 oncology 0.009434 0ncology 0.009434 cncology 0.009434 dncology 0.009434 nacology 0.009434 ncology 0.009434 necology 0.009434 oacology 0.009434 oncalogy 0.009434 oncnlogy 0.009434 oncofogy 0.009434 oncoiogy 0.009434 oncolagy 0.009434 oncolbgy 0.009434 oncoloay 0.009434 oncolocy 0.009434 oncoloey 0.009434 oncolog 0.009434 oncologv 0.009434 oncolooy 0.009434 oncolopy 0.009434 oncoloqy 0.009434 oncotogy 0.009434 onculogy 0.009434 oneology 0.009434 onoology 0.009434 onrology 0.009434 ontology

So, all the variants are substantially down-graded. This has adverse effect particularly on recall. Our proposed method assigns equal weight to all the variants as it is difficult to determine the relative superiority of a variant over another. Table 3 shows that our approach has higher recall values (which are significantly better at depths 500 and 1000) over SIGIR2015.

The difference between our proposed method and SIGIR2015 in Hindi, is not statistically significant. Serious mis-recognitions in this corpus may have made these two approaches indistinguishable.

5. Conclusion

In this paper we have presented a new paradigm which has not been well explored - improving IR performance from erroneous text without the availability of training data or language-specific resources. We have also proposed a novel approach to solving the problem and we have obtained statistically significant improvements over most of the baselines. The results show that we have achieved statistically significant improvements over the baselines. However, we are far from the performance as reported on the original text, which serves as an upper bound of performance. An error modelling approach is also kept for comparison. Encouraging results were obtained for two Indic scripts, namely, Bangla and Hindi. This is particularly useful as in these scripts the quality of the OCRs is still not very satisfactory. The application of our proposed method on a noisy legal collection devoid of clean-text version, has also established the efficacy of the approach.

References

Chakraborty, A., Ghosh, K., & Parui, S. K. (2015). Retrieval from noisy e-discovery corpus in the absence of training data (accepted). SIGIR '15. Santiago, Chile: ACM.

Chakraborty, A., Ghosh, K., & Roy, U. (2014). A word association based approach for improving retrieval performance from noisy ocred text. KDIR '14 (pp. 450–456). Rome, Italy: SCITEPRESS.

Chaudhuri, B., & Pal, U. (1996). Ocr error detection and correction of an inflectional indian language script. In *Proceedings of the 13th Int. Conf. on Pattern Recognition* (pp. 245–249). Vienna volume 3.

Choudhury, M., Thomas, M., Mukherjee, A., Basu, A., & Ganguly, N. (2007). How difficult is it to develop a perfect spell-checker? a cross-linguistic analysis through complex network approach. *Proceedings of the second workshop on TextGraphs: Graph-based algorithms for natural language processing*, (pp. 81–88).

Cormen, T., Leiserson, C., Rivest, R., & C.Stein (2009). *Introduction to Algorithms.* (3rd ed.). The MIT Press.

Fataicha, Y., Cheriet, M., Nie, J. Y., & Suen, C. Y. (2002). Content analysis in document images: A scale space approach. In *ICPR (3)* (pp. 335–338). IEEE Computer Society.

Fataicha, Y., Cheriet, M., Nie, Y., & Suen, Y. (2006). Retrieving poorly degraded ocr documents. *Int. J. Doc. Anal. Recognit.*, *8*, 15–26.

Garain, U., Paik, J., Pal, T., Majumder, P., Doermann, D., & Oard, D. (2013). Overview of the fire 2011 risot task. (pp. 197–204). Springer volume 7536.

Ghosh, K., & Chakraborty, A. (2012). Improving ir performance from ocred text using cooccurrence. *FIRE RISOT track 2012 working notes*, .

Ghosh, K., & Parui, S. K. (2013). Retrieval from ocr text : Risot track. (pp. 214–226). Lecture Notes in Computer Science: Springer volume 7536.

Grossman, M. R., & Cormack, G. V. (2011). Technology-assisted review in e-discovery can be more effective and more efficient than exhaustive manual review. *Richmond Journal of Law and Technology*, *XVII*.

Harman, D. (1995). Overview of the fourth text retrieval conference. (pp. 1–24). The Fourth Text Retrieval Conference.

Kang, H.-K., & Choi, K.-S. (1997). Two-level document ranking using mutual information in natural language information retrieval. *Inf. Process. Manage.*, *33*, 289–306.

Kantor, P., & Voorhees, E. (1996a). Report on the trec-5 confusion track. (pp. 65–74). The Fifth Text Retrieval Conference.

Kantor, P. B., & Voorhees, E. M. (1996b). The trec-5 confusion track: Comparing retrieval methods for scanned text. Available at : http://trec.nist.gov/data/t5_confusion.html.

Kolak, O., & Resnik, P. (2002). Ocr error correction using a noisy channel model. (pp. 257–262). Proceedings of the Second International Conference on Human Language Technology Research.

Magdy, W., & Darwish, K. (2006). Arabic ocr error correction using character segment correction, language modeling, and shallow morphology. (pp. 408–414). In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006).

Majumder, P., Mitra, M., Parui, S. K., Kole, G., Mitra, P., & Datta, K. (2007). Yass: Yet another suffix stripper. *ACM Trans. Inf. Syst.*, *25*, 18:1–18:20.

Marinai, S. (2009). Text retrieval from early printed books. In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data* AND '09 (pp. 33–40). New York, NY, USA: ACM.

Oard, D. W., Baron, J. R., Hedin, B., Lewis, D. D., & Tomlinson, S. (2010). Evaluation of information retrieval for e-discovery. *Artif. Intell. Law*, *18*, 347–386.

Paik, J. H., Mitra, M., Parui, S. K., & Järvelin, K. (2011a). Gras: An effective and efficient stemming algorithm for information retrieval. *ACM Trans. Inf. Syst.*, *29*, 19:1–19:24.

Paik, J. H., Pal, D., & Parui, S. K. (2011b). A novel corpus-based stemming algorithm using co-occurrence statistics. In *ACM SIGIR* SIGIR '11 (pp. 863–872).

Paik, J. H., & Parui, S. K. (2011). A fast corpus-based stemmer. *ACM Trans. Asian Lang. Inf. Process.*, *10*, 8.

Parapar, J., Freire, A., & Barreiro, A. (2009). Revisiting n-gram based models for retrieval in degraded large collections. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval* ECIR '09 (pp. 680–684). Berlin, Heidelberg: Springer-Verlag.

Reynaert, M. (2014). Ticclops: Text-induced corpus clean-up as online processing system. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations* (pp. 52–56). Dublin City University and Association for Computational Linguistics.

Siegel, S. (1956). *Nonparametric statistics for the behavioral sciences.* McGraw-Hill series in psychology. McGraw-Hill.

Singhal, A., Salton, G., & Buckley, C. (1996). Length normalization in degraded text collections. (pp. 149–162). Las Vegas, NV, USA: In Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval.

Taghva, K., Borsack, J., & Condit, A. (1994). Results of applying probabilistic ir to ocr text. (pp. 202–211). Dublin, Ireland: In The Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.