A Content-based Probabilistic Correction Model for OCR Document Retrieval

Rong Jin, Alex G. Hauptmann , ChengXiang Zhai
School of Computer Science, Carnegie Mellon University

ABSTRACT

The difficulty with information retrieval for OCR documents lies in the fact that OCR documents comprise of a significant amount of erroneous words and unfortunately most information retrieval techniques rely heavily on word matching between documents and queries. In this paper, we propose a general content-based correction model that can work on top of an existing OCR correction tool to “boost” retrieval performance. The basic idea of this correction model is to exploit the whole content of a document to supplement any other useful information provided by an existing OCR correction tool for word correction. Instead of making an explicit correction decision for each erroneous word as typically done in a traditional approach, we consider the uncertainties in such correction decisions and compute an estimate of the original “uncorrupted” document language model accordingly. The document language model can then be used for retrieval with a language modeling retrieval approach. Evaluation using a TREC standard testing collection indicates that our method significantly improves the performance when compared with simple word correction approaches such as using only the top ranked correction.

Keywords: information retrieval for OCR texts, statistical model, content based correction model

1. INTRODUCTION

Information retrieval for OCR generated texts has attracted a lot of interests in recent years due to its practical importance and theoretical value. Since many documents are actually acquired by applying OCR techniques to recognize text information from images, information retrieval for OCR generated texts is essential for searching through such documents. Meanwhile, since OCR generated texts are usually erroneous, it poses a great challenge for information retrieval in terms of finding relevant documents under a noisy environment.

In order to deal with the word errors in OCR generated texts, previous research has followed two groups of approaches[1], namely correction based approaches [2][3] and partial match based approaches [4]. The former approaches try to correct the erroneous words by using spelling checking tools, which can be either dictionary based, language model based, or specific to OCR generated errors. Then, the information retrieval task is performed on the corrected OCR documents instead of the original ones. The second group of approaches are based on partial matching, i.e. even though an erroneous word in a document may not match exactly with the corresponding correct query word, some part of the word may still match with the query word. Therefore, instead of only considering the cases of complete matches with query words, we also need to give credits to the cases of partial matches. The usual practice of this idea is to decompose every OCR created word into a set of n-grams (i.e., a sequence of *n* characters), and compute the similarity between documents and queries based on the matched n-grams instead of the complete words.

Compared with the partial match based approaches, the correction based approaches have several advantages. First, by simply replacing the erroneous words with the correct ones suggested by spelling checking tools, we can use any standard information retrieval system with little modification to find the documents relevant to the user’s queries. Second, since spelling checking tools are able to take advantage of the characteristics of natural language and OCR procedures, they are often able to suggest the right words for the OCR mistakes somewhere in their correction list. Therefore, the correction based approaches usually are quite robust if the spelling checking tools are of high quality. Finally, correcting the OCR mistakes in a document would make the document more readable to a user. In principle, one could apply any spelling checking tool (e.g., [5]) to correct the documents and then use any standard retrieval algorithm for retrieving documents. However, in most cases, the spelling checking tool gives a *list* of possible corrections rather one single correction for an erroneous word. Thus, one difficulty with correction based approaches is that a process of disambiguation is required to decide which word in the correction list is the right correction. The retrieval performance can be affected significantly by the accuracy of such disambiguation. The intuitively appealing approach of using the top ranked word in the correction list as the right word is risky, because there is a good chance that the right word may be actually down at the bottom of the list. Approaches that treat every word in the correction list as equally likely being the right word is also problematic, since the top ranked words on the list usually have a much better chance to be the right correction than those at the bottom. Note that when a correction tool suggests only one correction, the problem is not really solved, but hidden in the correction tool, unless the correction tool makes no mistakes. Thus, the general problem here is how to deal with the uncertainties in the word correction decisions. The difficulty mentioned above actually reveals a major deficiency in the traditional approaches – resolving the uncertainties explicitly is neither necessary nor desirable. Indeed, for the purpose of retrieval, it is better to keep such uncertainties so that each candidate word in the correction list can potentially match a query word. Of course, we ought to weight these candidates appropriately so that matching a top-ranked term would count more than matching one at the bottom. In this paper, we propose a general Content-based Probabilistic Correction (CPC) model that not only would keep such uncertainties, but also could work on top of any existing OCR correction tool to boost retrieval performance. The correction model is based on a source-channel framework in which the original (uncorrupted) document language model is the source and any OCR correction tool provides "weak" information about the probabilistic "corruption" channel. Our goal is to estimate the original document language model given the observed words in the corrupted OCR document. Thus, while the word correction preferences are modeled through probability distributions, we would never make any explicit correction. Instead, such preferences are combined somehow to estimate a most likely (original) document language model, which can then be used to perform retrieval using a language modeling approach. The CPC model assumes a preference model for word correction based on the whole content of a document, but otherwise makes minimum assumption about the corruption channel model. In its most general form, it can incorporate any useful information that an OCR correction tool can provide as features in an exponential model, which allows for combining any preference information from the correction tool with the content-based preferences. In this paper, however, we only explore an extremely simple case where the only feature from the correction tool used is the rank of a correction word. We test the CPC model on top of the Microsoft word spelling checker by using a standard TREC-5 confusion track collection. The results show that the CPC model significantly outperforms the simple approach of using the top ranked words in the correction list.

The rest of the paper is arranged as follows: The full description of our content-based probabilistic correction model is presented in Section 2. Section 3 describes the setup of the experiments and the results. Conclusions and the future work are presented in Section 4.

2. A CONTENT-BASED PROBABILISTIC CORRECTION MODEL

The CPC model is expected to work on top of an existing spelling/OCR correction tool, but with only minimum assumptions about such a tool. Specifically, we assume that the correction tool is able to (1) detect whether an OCR generated word is correct or not; (2) suggest a ranked list of candidate correction words if the OCR word is detected as incorrect. We further assume that the spelling checking tool is sufficiently accurate so that the correct word is almost always in the correction list; with a sufficiently large list of candidate correction words, this is a reasonable assumption.

2.1 Intuition

The CPC model can be described using the source-channel paradigm [6] as shown in Figure 1. In this model, the OCR document is generated from the original 'perfect' English document through a noisy channel, which corrupted a English word w into the OCR word o according to the distribution $P(o|w)$, i.e. the probability of generating OCR word o given the English word w. To recover the word distribution $P(w)$ in the original 'perfect' document, we can "reverse the engineering" and infer the source word distribution based on the observed word distribution in the OCR document $P(o)$ and the noisy channel $P(o|w)$.

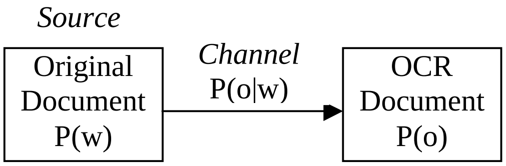

Figure 1: source-channel paradigm for correcting OCR mistakes. The source, namely the original document, has the word distribution P(w) and is corrupted into the distribution P(o) through the noisy channel P(o|w).

The key element is the channel model $P(o|w)$, which tells us how the OCR process introduces errors, and thus also gives us information about which English word w is likely to be the original word for a possibly erroneous OCR word o. The main idea of the CPC model is to compute an approximated noisy channel $P(o|w)$ using the content information of the document. More specifically, given a (ranked) list of candidate words for a given OCR word, we want to estimate which word in the correction list is more likely to be the right one, and we want to base such estimation on the whole content of the document so that a candidate word would be preferred if it is consistent with the content of the document. The simplest representation of the content of a document is its term frequency distribution. With this representation, whether a candidate word is consistent with the content of the document can be simply measured by the term frequency of the candidate word in the document. A candidate word with high frequency can be assumed to be strongly correlated with the content of the document and therefore should be treated as being highly likely a correct one, whereas if a candidate word rarely appears anywhere in the document, it can be assumed to have only a small chance to be correct.

Unfortunately, when a large percentage of the OCR-generated words are incorrect, a direct counting of term frequency distribution for a document can be problematic since the choice of correction words for erroneous OCR words can also have significant influence on the term frequency distribution. There is a cycle between deriving the term frequency distribution for a document and choosing the correct candidate words. That is, the term frequency of a document is determined based on the choices of correction words and meanwhile the choices of correction words are also influenced by the term frequency distribution of the document. To handle this issue, we adopt the Expectation-Maximization (EM) algorithm [7]. The underlying idea is the following: Initially, since we don't know which candidate word within the correction list is more likely to be the correct one, we assign equal likelihood for every word in the list. With this presumed likelihood distribution, we can estimate the term frequency distribution of the document. Then, with the help of this rough term frequency distribution, the likelihood for each candidate word within the list can be recomputed. Based on the recomputed likelihood for the candidate words, the term frequency distribution is further refined. This iteration will be carried on until the convergence of both term frequency distribution and likelihood for the candidate words.

2.2 Formal description

In this subsection, we describe our approach more formally. For the purpose of information retrieval, our goal is to find the word distribution $P(w)$ in the original document based on the observed word distribution $P(o)$ in the corrupted OCR document. One reasonable assumption is that the correction of OCR mistakes should be consistent with the content of the document. Therefore, the optimal true word distribution $P(w)$ should have the highest probability to be corrupted into the OCR word distribution $P(o)$. The probability of corrupting the original document D_{orig} into the OCR document D_{OCR} can be expressed as

$$P(D_{OCR} \mid D_{Orig}) = \prod_{o} \left(\sum_{w} P(w \mid M_{orig}) P(o \mid w) \right)^{tf(o, D_{OCR})} \qquad (1)$$

where D_{OCR} stands for the OCR document and D_{orig} stands for the 'perfect' version of the same document. Probability $P(w|M_{orig})$ is the word distribution for the original document D_{orig} and corruption probability $P(o|w)$ stands for the likelihood that the OCR word o is generated by corrupting the English word w. $tf(o, D_{OCR})$ is the term frequency of the OCR word o in the OCR document D_{OCR}. Intuitively, Equation (1) means that we generate the corrupted document

D_{OCR} by generating every OCR word instance in the OCR document D_{OCR}, which results in the product in Equation (1). Since we are not sure which English word *w* in the original document D_{orig} is responsible for the corrupted OCR word *o*, we sum over all the words in the original document in order to generate the OCR word *o*.

To simplify Equation (1), we can rely on a spelling checking tool to tell which OCR word is incorrect and to provide a correction list for the incorrect OCR word. Let function *f* stand for the function of spelling checking, which takes an OCR word *o* as input, and outputs a ranked list of candidate words $f(o)=\{w_1, w_2, \ldots, w_n\}$. When the OCR word is correct, the spelling checking function *f* simply outputs the OCR word itself. With the help of the spelling checking function, we do not have to count every word *w* in the original document D_{orig} as a correction candidate for the OCR word *o*. Instead, we only need to consider the words in the correction list *f*(*o*). Therefore, Equation (1) can be rewritten as

$$P(D_{OCR} \mid D_{Orig}) = \prod_{o} \left(\sum_{w \in f(o)} P(w \mid M_{orig}) P(o \mid w) \right)^{tf(o, D_{OCR})} \qquad (2)$$

where, the sum only goes over the words in the correction list *f*(*o*).

Now, we still miss the most important component in the model, i.e. the corruption probability *P*(*o*|*w*). Since the parameter *P*(*o*|*w*) is required for every English word *w* and every OCR word *o*, there may be too many parameters in this model. Given that the corruption probability *P*(*o*|*w*) is unknown, it would be useful to first reduce the number of parameters. The question is how we should parameterize the probability *P*(*o*|*w*) so as to reduce the number of parameters to be estimated. Our idea is to exploit the “weak” preference information provided by the assumed OCR correction tool. Note that *P*(*o*|*w*) encodes our knowledge about how an OCR error is typically made, i.e., the correlation between *o*'s and *w*'s, and it is this probability that allows us to incorporate into our framework any existing OCR correction tool(s), whenever available. More specifically, we may assume that the OCR correction tool(s) can provide values for *k* features that are relevant to the estimation of *P*(*o*|*w*). At least, the rank information of a word in the suggested correction list can be such a feature. Formally, let $\{f_i(w,o) \mid i=1,\ldots,k\}$ be the k features that we are interested in , we can assume the following general exponential model for *P*(*o*|*w*):

$$P(o \mid w) = \frac{1}{Z_w} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(w,o) \right) \qquad (3)$$

where λ_i's are parameters and Z_w is a normalizer that ensures that *P*(*o*|*w*)'s sum to one. Under this assumption, our generative model for an OCR document (with explicit parameters) can be written as

$$P(D_{OCR} \mid D_{Orig}, M_{Orig}, \lambda_1, \ldots, \lambda_k) = \prod_{o} \left(\sum_{w \in f(o)} P(w \mid M_{orig}) \frac{\exp\left(\sum_{i=1}^{k} \lambda_i f_i(w,o) \right)}{\sum_{o'} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(w,o') \right)} \right)^{tf(o, D_{OCR})} \qquad (4)$$

The parameters for this model include λ_1, λ_2, …, λ_k, and the $P(w|M_{orig})$'s. So, instead of having a corruption probability *P*(*o*|*w*) for every English word *w* and every OCR word *o*, we now have only k parameters for all (*w*,*o*) pairs, corresponding to the “importance” of the *k* features respectively. The $P(w|M_{orig})$'s are the original document language model that we really want to estimate. These parameters can be estimated using the Maximum Likelihood (ML) estimator, that is, we obtain the optimal original document models M_{orig}'s and optimal λ_i's by maximizing the document corruption probability $P(D_{OCR}| D_{Orig})$ for all the OCR document D_{OCR} in the collection. Formally, let $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_k, M_{orig1}, \ldots, M_{origN})$, where *N* is the total number of OCR documents, our estimate of Λ^* is given by

$$\Lambda^* = \arg\max_{\Lambda} \prod_{i=1}^{N} P(D_{OCRi} \mid D_{Orig_i}, M_{Orig_i}, \lambda_1, \ldots, \lambda_k) \qquad (4)$$

Given the form of our likelihood function, in general, we can treat the actual original word as a hidden variable and apply the EM algorithm[7] with an embedded improved iterative scaling algorithm[8] to find the ML estimate.

In this paper, however, we explore a simple special case of this general correction model, in which we essentially use only one feature -- the rank of word w in the correction list for the OCR word o. That is, we assume that *P*(*o*|*w*) only depends on the rank position of the English word *w* in the correction list for OCR word *o*. Furthermore, to simplify the computation, we will parameterize *P*(*o*|*w*) in a slightly different form than the general exponential model. Let *r*(*o*,*w*) stand for the rank position of the English word w in the correction list for erroneous OCR word o. The corruption probability $P(o|w)$ is expressed as

$$P(o \mid w) = \frac{P(r(o,w))}{t(w,r(o,w))} \tag{5}$$

where $t(w,r(o,w))$ is the number of different OCR words that have English word w ranked at $r(o,w)$ in their correction list. Probability $P(r)$ stands for the probability when the ranked r correction is the right correction. Of course, the sum of $P(r)$ over all the possible ranks should be one, i.e. $\sum_r P(r) = 1$. Now, instead of having a different parameter for every word w and o, we only need probabilities for different ranks. Note that we have used $t(w,r(o,w))$ as an approximation for $\sum_{o'} P(r(o',w))$. This is not a very accurate approximation, but it simplifies the computation significantly, as now we can use a simple EM algorithm to estimate the parameters. Under this approximation, our new expression for the 'translation' probability $P(D_{OCR}| D_{Orig})$, is

$$P(D_{OCR} \mid D_{Orig}) = \prod_{o} \left(\sum_{w \in f(o)} P(w \mid M_{orig}) \frac{P(r(o,w))}{t(w,r(o,w))} \right)^{tf(o,D_{OCR})} \tag{6}$$

The parameters now include all the $P(r)$'s and the $P(w|M_{orig})$'s. There is no analytic formula for the ML estimate of these parameters. Intuitively, we run into the following egg-chicken problem. To obtain the optimal rank probability $P(r)$, the information on the word distribution of the original document is required. On the other hand, the word distribution of the original document can be derived if the rank probabilities are known. To solve this problem, we can apply the Expectation-Maximization (EM) algorithm [7]. First, we can assume a uniform distribution for the rank probability $P(r)$. With the knowledge of rank probability $P(r)$, we can estimate the word distribution for the original document $P(w|M_{Orig})$ by probabilistically correcting every erroneous OCR word in the OCR document D_{OCR} using the rank probability $P(r)$. Then, we can have a new version of rank probability, and so on, so forth. More specifically, the EM updating equations for both rank probabilities $P(r)$ and the language model for the original document $P(w|M_{Orig})$ are

$$P(r) = \frac{1}{Z_r} \sum_{D_{OCR}} \sum_{o \in D_{OCR}} \left(tf(o,D_{OCR}) \sum_{w \in f(o)} \frac{\delta(r(o,w),r) P(w \mid M_{orig}) P'(r)}{t(w,r(o,w)) \sum_{w' \in f(o)} P(w' \mid M_{orig}) \frac{P(r(o,w'))}{t(w',r(o,w'))}} \right) \tag{7}$$

and

$$P(w \mid M_{orig}) = \frac{1}{Z(M_{orig})} \sum_{\{o \mid w \in f(o)\}} \frac{tf(o,D_{OCR}) P(r(o,w))}{|D_{OCR}|} \tag{8}$$

In Equation (6), $P'(r)$ stands for the rank probability obtained in the last iteration and $P(r)$ is the rank probability of current iteration. Symbol Z_r is the normalization constant that forces the sum of the rank probabilities $P(r)$ to be one. Symbol $Z(M_{Orig})$ is the normalization constant for the document model M_{Orig} so that the sum of the word distribution $P(w|M_{Orig})$ is one.

Equation (8) is a simple 'correction' procedure that replaces every OCR word o with English word w according to the rank probabilities $P(r)$ when the correction list of OCR word o includes the English word w. The underlying logic behind Equation (6) reflects our preference for a word consistent with the document content. As seen from the denominator of the inner term in Equation (6), rank probability $P(r)$ is proportional to the word distribution $P(w|M_{Orig})$, which indicates that rank r will be favored if for most cases the correction words at rank r are consistent with the 'expected' content of the document namely $P(w|M_{Orig})$. Thus, a correction is favored if it is consistent with the content of the document, which is represented by the word distribution $P(w|M_{Orig})$.

By using Equation (6) and Equation (7) iteratively, we are able to obtain the rank probabilities $P(r)$ and the expected language model for the original document $P(w|M_{Orig})$ at the same time. To accomplish the information retrieval task, we can simply adopt the language modeling approach to information retrieval [5], in which the 'expected' document language model would be used to compute the likelihood of the query.

3. EXPERIMENTS

The goal of our experiments is to examine the effectiveness of our content-based correction model for OCR documents. We use the OCR document collection (with 20% degradation) from the TREC5 confusion track[1], where 20% of the texts in the OCR collection are corrupted. There are a total of 50 queries, and for each query there is one and only one relevant document within the whole collection. As pointed out in [1], this is a known item retrieval problem for which the average reciprocal rank can be used as the evaluation metric. To retrieve documents relevant to a query, we use the language modeling approach [8], in which we compute the query likelihood according to the language model estimated for each document. We use a popular linear interpolation smoothing method, and end up with the following generic form for computing document-query similarity (see e.g., [9])

$$P(Q \mid D) = \prod_{qw \in Q} (\alpha P(qw \mid D) + (1-\alpha) P(qw \mid GE)) \tag{8}$$

where α is a smoothing constant which is set to 0.5 in all experiments. $P(qw|D)$ is the unigram language model for the document D. $P(qw|GE)$ is the unigram language model for the general English, which can be computed by averaging the document unigram language model over all the documents within the collection. To obtain the correction lists for erroneous OCR words, we use MS WORD for spelling checking. With the help of the API of MS WORD, we are able to automatically obtain the suggested corrections for erroneous words and save them to a file. For the sake of efficiency, we only keep up to top 10 suggested corrections.

To evaluate the retrieval effectiveness of the CPC model, we choose three simple baseline models: **Model 1** uses the top 2 correction words as the potential right corrections; **Model 2** considers the top 5 corrections to be equally likely candidates for the right correction; **Model 3** treats all the 10 suggested corrections as equally good corrections. Table 1 lists the results for the three baseline models as compared with the CPC model:

	Baseline Models			Content-based Correction Model
	Model 1	Model 2	Model 3	
Ave. Reciprocal Rank	0.20	0.33	0.37	0.41

Table 1: Average reciprocal rank for the three baseline models vs. the content-based correction model.

The first thing to be noticed from Table 1 is that there is an order among the three baseline models: Baseline model 3 is better than model 2 which is better than model 1. Since the sole difference among model 1, model 2 and model 3 is the number of candidate words from the correction list that are actually used, this performance order indicates that it is better to include more candidate words in consideration for the purpose of retrieving documents. This is expected, as most information retrieval techniques are based on word matching, and so, to find the document relevant to the query, it is critical for the relevant document to match the query words. With more correction words under consideration, the chance to have the right correction will be higher, which results in the improvement on the performance of information retrieval. Given this observation, it would be very interesting to further experiment with other more inclusive cutoff values.

Secondly, the CPC model gives much better performance than all the three baseline models with an average reciprocal rank of 0.41. To better understand the success of our model, we can look at the top 10 rank probabilities shown in Table 2.

Rank	1	2	3	4	5	6	7	8	9	10
$P(r)$	0.475273	0.205318	0.099045	0.082305	0.046861	0.033991	0.026327	0.013597	0.010900	0.006383

Table 2: Rank probabilities

As seen from Table 2, a majority of the probability mass is distributed over rank 1 and 2, which indicates that the correction at ranks 1 and 2 has a 2/3 chance to be correct , if we assume that the right correction always falls into one of the top 10 ranks. Meanwhile, there are still 1/3 of the times when the correct word is ranked from 3 to 10. This simple computation gives a quantitative explanation why baseline model 1 has performed significantly worse than all the other models: It is because model 1 only considers the top 2 candidates, and therefore throws away 1/3 of the correct candidates. Due to the reliance on word-matching of information retrieval, this can be expected to degrade the performance of retrieval significantly.

Both the baseline model 3 and the CPC model consider all the top 10 candidates, but the CPC model has the advantage of being able to give them a different priority based on the rank probability $P(r)$. According to Table 2, the top 2 candidate words should be considered as being much more important than those ranked from 3 to 10. With the help of the optimal rank probability $P(r)$, the CPC model is able to emphasize the right candidates and penalize the wrong candidates probabilistically, and therefore results in an even better performance than model 3 -- improving the average reciprocal rank from 0.37 to 0.41. The performance of the CPC model is quite competitive when compared with the performance of the official TREC5 systems[1].

4. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel correction model for OCR documents, namely a content-based probabilistic correction model. This correction model intends to prefer correcting erroneous OCR words in a way that is consistent with the content of the document. Specifically, for the unigram representation of a document, this model will look at the word distribution of a document and give high probabilities to those candidate words that are popular within the document.

The correction model is a very general model that can work on top of an existing OCR correction tool to "boost" retrieval performance. The whole content of a document, as represented by a unigram language model, is integrated with any other useful feature information provided by one or more existing OCR correction tools in a unified probabilistic generative model. Furthermore, instead of making an explicit correction decision for each erroneous word as typically done in a traditional approach, we consider the uncertainties in such correction decisions and compute an estimate of the original "uncorrupted" document language model accordingly. The document language model can then be used for retrieval with a language modeling retrieval approach.

We implemented a special case of the general correction model that uses the rank information provided by an external OCR correction tool, and evaluated this model using the standard testing collections from the TREC5 confusion track. The experiment results indicate that the correction model significantly improves the performance when compared with three baseline simple word correction approaches of using top-k ranked word candidates for correction with equal probabilities. Our performance is also very competitive when compared with the performance of the official TREC5 systems.

A main line of future work is to extend this correction model to its full spectrum. For example, we have only explored the use of rank information as a feature, it would be very interesting to consider using more features from an existing OCR correction tool, which can be expected to improve the model for the 'corruption' probability $P(o|w)$, Indeed, we could consider combining features from *different* OCR correction tools in our framework. Finally, we believe that the proposed correction model can also be applied to other retrieval tasks involving "corrupted" documents. One possible application is cross-language retrieval where documents in one language can be regarded as being generated by "corrupting" documents in another language.

ACKNOWLEDGEMENTS
This material is based in part on work supported by National Science Foundation under Cooperative Agreement No. IRI-9817496. Partial support for this work was provided by the National Science Foundation' s National Science, Mathematics, Engineering, and Technology Education Digital Library Program under grant DUE-0085834. This work was also supported in part by the Advanced Research and Development Activity (ARDA) under contract number MDA908-00-C-0037. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or ARDA.

REFERENCES

1. P. Kantor and E. Voorhees, Report on the TREC-5 Confusion Track, In *Proceeding of the fifth Text Retrieval Conference TREC-5*, NIST Special Publication 500-238, 1996
2. X. Tong and C. Zhai and N. Milic-Frayling and D. A. Evans, OCR Correction and Query Expansion for Retrieval on OCR Data -- CLARIT TREC-5 Confusion Track Report, In *Proceeding of the fifth Text Retrieval Conference TREC-5*, NIST Special Publication 500-238, 1996
3. X. Tong and D. A. Evans, "A Statistical Approach to Automatic OCR Error Correction in Context". Proceedings of the Fourth Workshop on Very Large Corpora (WVLC-4), Copenhagen, Denmark, August 4, 1996, 88--100.
4. S. M. Harding, W. B. Croft and C. Weir, Probabilistic Retrieval of OCR Degraded Text Using N-Grams. In *Proceedings of First European Conference on Digital Libraries*, 1997

5. A. R. Golding and D. Roth. A Winnow based approach to context-sensitive spelling correction. Machine Learning, 34(1-3):107-130, 1999. Special Issue on Machine Learning and Natural Language.
6. C. E. Shannon, ` ` A mathematical theory of communication," Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.
7. A. P. Dempster, N. M. Laird, and D. B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society*, 1977.
8. S. Della Pietra, V. Della Pietra, J. Lafferty, "*Inducing Features of Random Fields*", in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, p.380 (1997).
9. J. Ponte and B. Croft. A language modeling approach to information retrieval. In Proceedings, 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, page 275—281, Melbourne, Australia, August 1998
10. D R. H. Miller, T. Leek, and R. M. Schwartz. A Hidden Markov Model Information Retrieval System. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.