Searching Corrupted Document Collections

Jason Soo
Information Retrieval Laboratory
Georgetown University
Washington, DC, USA
soo@ir.cs.georgetown.edu

Ophir Frieder
Information Retrieval Laboratory
Georgetown University
Washington, DC, USA
ophir@ir.cs.georgetown.edu

***Abstract*—Historical documents are typically digitized using optical Character Recognition. While effective, the results may not always be accurate and are highly dependent on the input. Consequently, degraded documents are often corrupted. Our focus is finding flexible, reliable methods to correct for such degradation, in the face of limited resources. We extend upon our substring and context fusion based retrieval system known as Segments, to consider metadata. By extracting topics from documents, and supplementing and weighting our lexicon with co-occurring terms found in documents with those topics, we achieve a statistically significant improvement over the state-of-the-art in all but one test configuration. Our mean reciprocal rank measured on two free, publicly available, independently judged datasets is 0.7657 and 0.5382.**

***Keywords*-OCR; known item retrieval;**

I. Introduction

In the information era, the demand to retrieve documents related to one's query is commonplace. This task is simplified if the document originated electronically. Naturally, historical documents do not offer such luxury, but their value remains unchanged. Such documents often are processed by an optical character recognition (OCR) process to digitize them, which facilitates indexing and searching. However, the accuracy of this OCR process on the resulting document can vary greatly depending on the quality of the original document. We focus on increasing the quality of the search of the resulting document through post-processing.

The OCR process is comprised of two phases. In the first phase, algorithms infer pixel boundaries from an image and make probabilistic judgments about what characters the pixels represent. In the second phase, the resulting electronic document is analyzed using an array of methods including context considerations, text analysis, and machine learning, with the goal of correcting errors introduced in phase one.

Seminal second phase efforts appeared around the time of TREC-5 due to the creation of the confusion track dataset [1]. This TREC track focused on known-item document retrieval, where, given a query, a single target document was sought. The evaluation corpora contained an estimated error rate of 5% and 20%. Submitted approaches retrieved documents in the face of these error rates. The submissions adopted one of two approaches: correcting the corrupted document, or expanding the query to match the corrupted documents. Experimental results show the former approach achieved higher mean reciprocal ranks (MRR).

In the years following the TREC-5 confusion track competition, more solutions to this problem were developed. These solutions targeted specialized cases, such as handwritten documents [2], reading signs [3], or were trained for a domain [4]. Other proposals depended on the availability of a resource which may not be readily present: OCR engine confidence level [5]; heavy utilization of online resources [6], [7]; or extensive web-crawling [8].

Perhaps the most exhaustive research on post-processing was performed by Taghva et al. They studied this problem for approximately 15 years [9], designed specialized engines for searching degraded documents [10], and concluded various methods of error corrections had little impact on precision/recall versus unmodified search engines [11]. This result suggests that Solr, a modern, open source, widely-used enterprise search engine based on Lucene, should provide a suitable solution to searching OCR corrupted collections. More so, as Solr is modern and widely-used, it is kept current with the available search enhancements. Hence, Solr does not suffer from aging. Taghva et al.'s most comparable work to our method uses statistical methods to make corrections, but requires user training and assistance [12]. More recent work from that lab focused on similar supervised approaches [13].

Recent external efforts focused on customized solutions, but unfortunately were only evaluated on proprietary datasets (e.g. [14], [6], [15], [16]). As the exact implementation details of these approaches are not available nor are the datasets used in their evaluation, direct comparison against these approaches is not feasible.

This body of prior art helped inspire and direct our solution. Over the past years, we evaluated methods for reliably correcting phase one errors via post-processing using our method called Segments [17], [18], [19]. Segments differs from previous research in that it is an unsupervised approach, which makes minimal assumptions about resource availability, and has no dependence on language within the algorithm. It uses substring rules to correct phase one errors. More recent research has focused on the role of balance between context, such as word-level bi-grams and tri-grams,

and term focused substrings, finding that a fusion between the two improves performance [19].

Furthermore, while not required, Segments is designed for use within adverse environments. Adverse environments prevent commons approaches from being used, such as machine learning which requires labeled data. Some examples of conditions that create adverse environments include:

- No Internet connection, such as when you're working on an airgap network;
- Lack of rich query logs, occurring with low volume search engines;
- No accurate user model, which may result from not tracking users;
- Poorly refined algorithms, resulting from a lack of inconclusive statistical hypothesis testing.

Here, we introduce a method for topic extraction and incorporation into the Segments' Fusion process. We evaluate this method and show statistically significant improvement over other approaches. Specifically, our contributions are:

- We develop an extension to Segments that extracts topics from documents, and supplements the Segments lexicon by way of querying the extracted topics against a corpus such as Wikipedia, and adding resulting terms to our lexicon, or weighting them if already present.
- We show that this process can significantly improve our approach on both datasets, statistically significantly ($p<0.01$ and $p<0.05$) outperforming all other approaches.

II. Methods

A. Dataset

The evaluation dataset for this work is the TREC-5 Confusion Track dataset, which is free and publicly downloadable[1].

1) Document Set: The TREC-5 Confusion Track datasets used for evaluation each consist of approximately 55,600 documents published daily by the United States Government. TREC provides 49 queries, and their associated query relevance judgments. The task, as defined by TREC, is a known-item document retrieval search. That is, there is one target document for each query provided. The performance of all submissions to this TREC track are judged using the common Mean Reciprocal Rank (MRR) measure. Three datasets are provided: a ground truth; 5% corrupted dataset; and 20% corrupted dataset. All datasets contain the same documents, but vary in the magnitude of corruption the OCR process introduced.

2) Real Word Dictionary: To identify OCR errors, we first identify real words by checking for a term's existence within our *real word* dictionary. This dictionary comprises 99,044 words from the English dictionary[2] and 94,293 sir names in the United States[3].

B. Baselines

To measure the performance acheived by Segments' topic extraction approach, we measure baselines of two other primary options: Solr and the best performing prior art.

To use Solr, we first indexed all datasets using Solr 4.6.1. We use the standard English stemmer, tokenizer, and stop word remover. We then issued the 49 TREC queries and measured the MRR. As Solr is a currently used enterprise search engines, by using it, we are modernizing the search accuracy. That is, via Solr's use, we incorporate the search technique improvements that have transpired since the TREC competition and prior research into our comparisons.

Our confidence in using Solr as a baseline is based on the extensive research by Taghva et al. who found that OCR error correction had little impact on precision/recall versus the unmodified retrieval strategies[11].

Additionally, we compare against the best reported results to date on the TREC-5 Confusion Track, namely those reported by the Swiss Federal Institute of Technology (ETH for short). Their method – which replaces corrupted terms with candidate vectors chosen by a probabilistic technique based on feature frequencies – reported an MRR of 0.5737 on the 5% dataset, and 0.4978 on the 20% dataset[20].

C. Experiments

We evaluate the accuracy of each approach in finding the target document given a query and a corrupted document collection. This requires each approach to compensate for the corruption. Our experimental construct requires a setup, and then an evaluation.

The setup consists of taking the corrupted document set, and filtering out noise. Briefly, this is done by taking all terms not found within our *real word* dictionary, at least 4 characters long (approximately within the top quarterly of term-frequencies) and removing stop-words, tokenizing, and stemming, with each step being tailored towards OCR-style errors. For example, we don't include symbols as white space when tokenizing because of the increased probability of OCR systems to replace characters with punctuation. A detailed explanation of the filtering process is described in [19]. We then iterate over all the filtered terms, tasking each evaluated approach to generate substitution candidates for the unrecognized terms. We measure performance at different candidate vector sizes, as referenced in the graphs by TopK.

The evaluation process works as follows. For each query in the 49 queries provided by TREC-5's Confusion Track dataset, we issue it to each of the evaluated approaches.

[1] http://trec.nist.gov/data/t5_confusion.html

[2] ftp://ftp.gnu.org/gnu/aspell/aspell-0.60.6.1.tar.gz

[3] http://www.census.gov/genealogy/www/data/1990surnames/dist.all.last

Then, we review the first 60 retrieved documents for each approach (60 was artificially chosen as a maximum number of search results a user may review; it represents 3 screens in a web retrieval, a number of screens rarely surpassed). We measure the MRR of the target document for each approach. We repeat 49 times, once per query, and report the average MRR of each approach.

D. Evaluated Correction Approaches

We compare our newly developed topic extraction Segments approach (Fusion-Topics) with three other retrieval strategies, plus the 2 baselines. Specifically, we compare against:

1) Word-level bigrams selected using Segments (Bi-Seg)
2) Fusion of Segments and word-level bigrams using Segments (Fusion)
3) Same as the above, but the candidates list is re-ranked using Edit Distance (Fusion-Ed)

The main goal of our evaluation is to measure the performance of our new approach, Fusion-Topics, over the previously best results in our work, Fusion. By also including Fusion-Ed and Bi-Seg, it gives us insight into whether there is any performance degradation in particular cases that yield advantages to another approach.

1) Segments (Seg): Segments is a system that takes an input string, and using 6 substring rules, returns a list of possible correction candidates derived from a lexicon, ranked by similarity.

The segments system consists of two parts: a candidate generation process (Algorithm 1), and a topic extraction/lexicon supplementing routine (Algorithm 2). Algorithm 2 is an optional pre-processing step to improve the performance of Algorithm 1. Notationally, an Algorithm 1 run on input string t is denoted as $\alpha(t)$.

Algorithm 1 takes input string q and checks for an exact match within our $Lexicon$. Should one exist, all processing of that term terminates. If the term is not found in our Lexicon, we execute our substring rules on the term. The rules, presented below, use the following descriptive conveniences: 1) λ represents the length of string t; 2) $t[0..4]$ returns the zeroth to fourth characters in t; 3) r_i will immediately return if $\lambda \leq 3$; 4) Cases where a floor or ceiling may be required are ignored; 5) $*$ represents a wildcard of zero or more characters.

1) $r_1(t) =$ return $t[0..\lambda/2]*$
2) $r_2(t) =$ return $*t[(\lambda/2)+1..\lambda]$
3) $r_3(t) =$ return $*t[2..\lambda-2]*$
4) $r_4(t) =$ return $*t[1..\lambda-1]*$
5) $r_5(t) = \beta := t[0..(\lambda/2)-1] * t[(\lambda/2)+1..\lambda]; f_5(\delta);$ return β
6) $r_6(t) = \beta := *t[1..\lambda-1]*; f_6(\delta);$ return β

The results of these substring rules comprise the set $Substrings$. Once generated, each string $s \in Substrings$ is searched against our Lexicon. If this is the first time seeing this candidate suggestion for this corrupted term, update the $VotesHash$ value associated with that string s to reflect the following value: $\Phi(r) = 1/(r+1)$, where r is the number of recursive function calls required to generate s from t. If there is already a non-zero value in $VotesHash$ for s then increment the value by $\Phi(r)$.

Next, $VotesHash$ is ordered by similarity scores and a confidence is computed:

$$Confidence(VotesHash) = \frac{max_{c \in VotesHash}\{c.value\}}{\sum_{VotesHash} value}$$

If $C_{Segments} \geq 0.3$ – suggesting a tight distribution of votes around a single term – return the ordered list.

```
for t ∈ Terms do
    if t ∈ Lexicon then
        return t;
    end
    Substrings ← ∅;
    for i ∈ {1..6} do
        Substrings ← append(Substrings, r_i(t));
    end
    VotesHash ← ∅;
    for s ∈ Substrings do
        Search Lexicon for s;
        if VotesHash[s] == ∅ then
            VotesHash[s] ← Φ(r);
        else
            VotesHash[s] ← VotesHash[s] + Φ(r);
        end
    end
    C_Segments ← Confidence(VotesHash);
    if C_Segments < 0.3 then
        n − grams ← n-grams result set;
        C_n−grams ← Confidence(n − grams);
        if C_Segments < C_n−grams then
            VotesHash ← n − grams;
        end
    end
    return VotesHash;
end
```

Algorithm 1: Segments Process

```
for d ∈ DocumentSet do
    Topics ← Extract topics from d;
    for t ∈ Topics do
        Query datasource for t;
        Extract keywords from top 10 results;
        Add keywords to Lexicon;
    end
end
```

Algorithm 2: Segments Topic Extraction

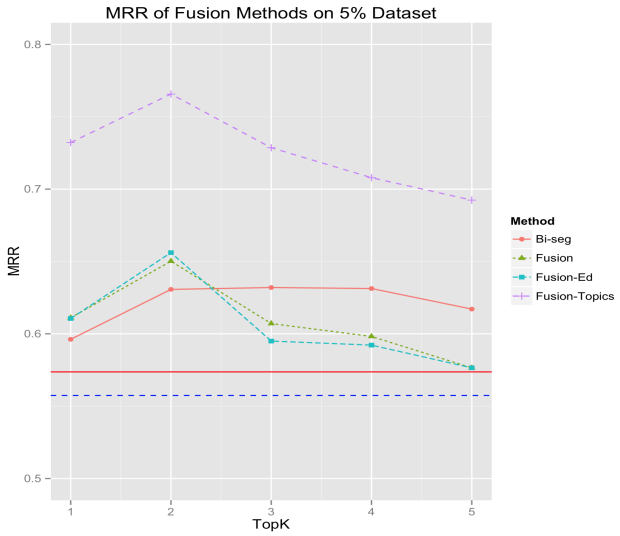

Figure 1. MRR of fusion methods on using the 5% dataset.

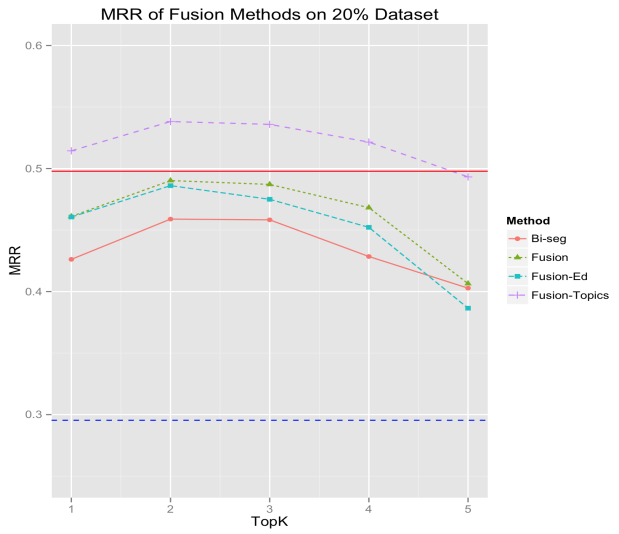

Figure 2. MRR of fusion methods on the 20% dataset.

Otherwise, the list has a high variance, so we do a final check with vanilla 3-grams. Using a standard 3-grams search process, we order the list of found candidates, and return the result set with higher confidence.

2) Segments Re-Ranked Selected Using Edit Distance (ED): This method runs the Segments method, but will re-rank the results using edit distance scores prior to returning the ranked candidate set. Note that this method is only used in the fusion approach, Fusion-Ed.

3) Word-level Bigrams Selected Using Segments (Bi-Seg): This method modified Segments to consider context, rather than individual strings in isolation. By using a sliding window approach derived from [21], we are able to correct corrupted terms using the terms occurring before and after (if any) the corrupted term. This approach is modified from other approaches, in that it will not pass line feed boundaries, but will pass punctuation boundaries, as OCR errors typically introduce punctuation.

By using the sliding window approach, each bigram pair contains a corrupted term t_c, and an additional set member t_p or t_f, where the former represented the previous term, and the latter represents the following term. Using this set, we check our bigram dataset W which is derived from Wikipedia [22], and return the following: $\forall w_i \in \alpha(w_c), \{w_p, w_i\} \in W \cup \{w_i, w_f\} \in W$, where w_i is a correction candidate for w_c as suggested by Segments. These bigrams are ranked by the vote count. Finally, we remove w_p and w_f

Note, it is possible that w_p or w_f are corrupted. We make no attempt to correct for this, and simply allow such occurrences to fail to find any matches.

4) Fusion Approach (Fusion and Fusion-Ed): An additional experiment is the evaluation of fusion approaches. Research in other areas has demonstrated that fusion-based approaches can improve performance [23]. As such, we fused together Seg and Bi-Seg, to form Fusion, and Ed and Bi-Seg, to form Fusion-Ed. The fusion approaches merge the result sets from the previously independent methods.

5) Fusion Using a Weighted Lexicon (Fusion-Topics): This approach is the same as the Fusion approach, except we create a weighted lexicon by extracting 5 topics from each document in each dataset (see Algorithm 2). For each unique topic, query an offline copy of Wikipedia, and parse the content of the top-10 results. After tokenizing, stemming and removing stop words, add these terms to our lexicon (if absent) and mark them as weighted terms (even if already present). In Segments, if a candidate term is marked as weighted, it uses the following modified similarity score function: $\Phi_w(r) = 1/(r+1) + 1.3^{-r}$. This essentially casts an additional vote for the candidate, making selection of that candidate term more likely.

III. Results

The Solr and ETH baseline are shown in the graphs using horizontal dashed and solid lines, respectively. For Solr, the reported MRRs on the 5% and 20% dataset are 0.5574 and 0.2954, respectively, and 0.5737 and 0.4978, respectively, for the ETH method.

The highest reported MRR from all measured approaches is 0.7657 and 0.5382, reported by Fusion-Topics with a candidate vector size of 2, on the 5% and 20% datasets, respectively. The statistical t-test was used to verify the increase in MRR is statistically significant versus Solr ($p<0.01$), the prior art ($p<0.01$ on the 5% dataset; $p<0.05$ on the 20% dataset), and all other approaches ($p<0.01$).

A. Methods

The results of our evaluations are shown in Figures 1 and 2, with the former graphing performance on the 5% dataset, and the latter graphing performance on the 20% dataset.

The immediate observation from the figures is the marked improvement on both the 5% and 20% datasets of the Fusion-Topics method. The approach, using its weighted lexicon, exceeds the performance of any other approach in both experiments, except for when k = 5 on the 20% dataset, where it nearly matches the performance of the prior art. In both datasets, we attribute the improvement to the modified similarity score function ($\Phi_w(r)$), which provides additional votes to the weighted candidate terms, making them more likely to be selected. This upholds our belief that the weighted terms are more likely to be relevant to corrupted terms because of their high probability to be found within a document containing such topics. To a lesser extent, new terms found during our lexicon supplemental process also account for the improvement. These terms primarily consisted of proper nouns not immediately found in our original lexicon, such as business names.

From looking closely at the results, we see that on both the 5% dataset, and the 20% dataset, maximum performance occurs with a candidate vector size of 2. This suggests that 2 may be the best candidate vector size for most applications. This is supported by previous research ([24], [19]) which show that a balance of context-free and context-dependent candidates perform best. By using a fusion method with a candidate vector size of 2, we select 2 candidates based on context (selected using bigrams) and 2 candidates not based on context (selected using Segments' substrings rules). The selection of only 2 candidates from each helps prevent potential term drift, wherein less relevant candidates are suggested.

Finally, we see that performance on the 20% dataset, while still a statistically significantly ($p<0.05$) improving over all other approaches (except at k = 5), is not as dramatic as on the 5% dataset. Error analysis shows that this is attributed to the difficulty in identifying good topics in the face of increased corruption. Poor topic extraction results in a weighted lexicon not as relevant to the document's true topics, and thus, the results are hampered, as shown when comparing the 5% results to the 20% results.

These results suggest that even when searching in adverse environments, MRR can still be improved with tailored solutions. Specifically, without an Internet connection, while ignoring language, an unsupervised approach can deliver good results.

B. Limitations

We recognize that we evaluated our work on only 2 datasets. We further acknowledge the existence of other datasets (e.g. UNLV [25], IMPACT [26]), but these datasets are not applicable to our work, as these datasets do not provide means to accurately evaluate our system; namely, they are lacking query relevance (qrel) judgments. Without those, we would only be measuring the correction accuracy of Segments, which has already been exhaustively studied in prior papers using heterogeneous datasets [27], [17], [18]. Therefore, despite the age of the TREC collection, it remains the only collection that provides ground truth, corrupted text, and 3rd party qrel judgments, in a publicly available package. We believe the described approach will generalize to other datasets as it has no reliance on language, writing style, document type, supervised training, or confusion matrices. So long as a lexicon can be obtained for the language, Segments can be used.

Furthermore, we acknowledge that a solution such as Google's "did you mean" feature is likely the state-of-the-art. However, it is not feasible in all scenarios. For example, if you are attempting to correct OCR errors on any moderately sized corpus, you will quickly be rate-limited by Google. That is, you will not be able to retrieve candidates for all of your corrupted terms. We observed this during our experimentation. Furthermore, our work is developed with the assumption that you will be working within an adverse environment (as described earlier). In such an environment, we presume there is no Internet connection available to process any potential errors within a query or document, and the approach must instead rely on available offline resources.

IV. Conclusion

Our goal herein was to improve on the performance of Segments through the use of metadata. By extracting topics from the document collection, and using them to find supplemental keywords for our lexicon, we were able to improve the performance of Segments. Using two free and publicly available datasets, we showed that our unsupervised, language independent algorithm, achieved improved results over the previous method.

This new method obtained an MRR of 0.7657 and 0.5382 on the 5% and 20% datasets, which statistically significantly outperforms the prior art's MRR of 0.5737 ($p<0.01$) and 0.4978 ($p<0.05$) and Solr's MRR of 0.5574 ($p<0.01$) and 0.2954 ($p<0.01$), as well as all other methods tested ($p<0.01$).

These findings support our previous research, and our hypothesis that both context and substring rules are important to correcting OCR errors. We therefore believe that fusion methods, supplemented with a with a weighted lexicon, are the best approach.

References

[1] P. B. Kantor and E. M. Voorhees, "The TREC-5 Confusion Track: Comparing Retrieval Methods for Scanned Text," *Information Retrieval*, vol. 2, pp. 165–176, 2000.

[2] N. Naji and J. Savoy, "Information Retrieval Strategies for Digitized Handwritten Medieval Documents," in *Information Retrieval Technology*, M. Salem, K. Shaalan, F. Oroumchian, A. Shakery, and H. Khelalfa, Eds. Springer Berlin Heidelberg, 2011, vol. 7097, pp. 103–114. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25631-8_10

[3] W.-s. W. Lian, "Heuristic-Based OCR Post-Correction for Smart Phone Applications," Ph.D. dissertation, University of North Carolina at Chapel Hill, 2009.

[4] J. Parapar, A. Freire, and A. Barreiro, "Revisiting N-Gram Based Models for Retrieval in Degraded Large Collections," in *31th European Conference on Information Retrieval Research*, M. Boughanem, C. Berrut, J. Mothe, and C. Soule-Dupuy, Eds. Springer Berlin Heidelberg, 2009, vol. 5478, pp. 680–684. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-00958-7_66

[5] S. Chen, D. Misra, and G. R. Thoma, "Efficient automatic OCR word validation using word partial format derivation and language model," in *Document Recognition and Retrieval XVII*, 2010, pp. 75 340O–1 – 75 340O–8.

[6] Y. Bassil and M. Alwani, "OCR Post-Processing Error Correction Algorithm Using Google's Online Spelling Suggestion," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, pp. 90–99, 2012.

[7] Y. Li, H. Duan, and C. Zhai, "CloudSpeller: Spelling Correction for Search Queries by Using a Unified Hidden Markov Model with Web-scale Resources," in *Special Interest Group on Information Retrieval*, 2011, pp. 0–4.

[8] C. Whitelaw, B. Hutchinson, G. Y. Chung, and G. Ellis, "Using the Web for Language Independent Spellchecking and Autocorrection," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2009, pp. 890–899.

[9] K. Taghva, T. Nartker, and J. Borsack, "Information access in the presence of ocr errors," in *Hard Document Processing*, 2004.

[10] K. Taghva, J. Coombs, and I. Science, "Hairetes: A search engine for ocr documents," in *In Proc. of 5th Intl. Workshop on Document Analysis Systems, Lecture Notes in Computer Science*. Springer-Verlag, 2002, pp. 412–422.

[11] K. Taghva, J. Borsack, and A. Condit, "Evaluation of model-based retrieval effectiveness with ocr text," *ACM Transactions on Information Systems*, vol. 14, pp. 64–93, 1996.

[12] K. Taghva and E. Stofsky, "Ocrspell: an interactive spelling correction system for ocr errors in text," *International Journal of Document Analysis and Recognition*, vol. 3, p. 2001, 2001.

[13] S. Poudel, "Post processing of optically recognized text via second order hidden markov model," *Masters Thesis, University of Nevada, Las Vegas*, 08 2012.

[14] E. Borovikov, I. Zavorin, and M. Turner, "A filter based post-ocr accuracy boost system," in *ACM HDP*, 2004.

[15] Y. Bassil and M. Alwani, "Ocr context-sensitive error correction based on google web 1t 5-gram data set," *American Journal of Scientific Research*, vol. abs/1204.0188, 2012. [Online]. Available: http://arxiv.org/abs/1204.0188

[16] Y. Mohapatra, A. K. Mishra, and A. K. Mishra, "Spell checker for ocr," *International Journal of Computer Science and Information Technologies*, vol. 4, pp. 91–97, 2013.

[17] J. Soo and O. Frieder, "On foreign name search," in *European Conference on Information Retrieval*. Springer Berlin Heidelberg, 2010, pp. 483–494.

[18] J. Soo and O. . Frieder, "On searching misspelled collections," *Journal of the Association for Information Science and Technology*, 2014.

[19] J. Soo and O. Frieder, "Revisiting Known-Item Retrieval in Degraded Document Collections," in *Document Recognition and Retrieval XXIII*, 2016.

[20] J. P. Ballerini, M. Büchel, R. Domenig, D. Knaus, B. Mateev, E. Mittendorf, P. Schäuble, P. Sheridan, and M. Wechsler, "SPIDER Retrieval System at TREC-5." in *TREC*, 1996.

[21] S. Mengle and N. Goharian, "Passage Detection Using Text Classification," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 4, pp. 814–825, Apr. 2009. [Online]. Available: http://dx.doi.org/10.1002/asi.v60:4

[22] Wikipedia, "Wikipedia Bigram Open Datasets," https://github.com/rmaestre/Wikipedia-Bigram-Open-Datasets/blob/master/datasets/bigram_EN.dat.gz. [Online]. Available: https://github.com/rmaestre/Wikipedia-Bigram-Open-Datasets/blob/master/datasets/bigram_EN.dat.gz

[23] B. Larsen, P. Ingwersen, and B. Lund, "Data fusion according to the principle of polyrepresentation," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 4, pp. 646–654, 2009.

[24] J. Soo, "A non-learning approach to spelling correction in web queries," in *Proceedings of the 22nd international conference on World Wide Web companion*. International World Wide Web Conferences Steering Committee, 2013, pp. 101–102.

[25] T. Nartker, R. Bradford, and B. Cerny, "A preliminary report on unlv/gt1: A database for ground-truth testing in document analysis and character recognition," in *Proceedings of the First Symposium on Document Analysis and Information Retrieval, Las Vegas, NV*, 1992.

[26] C. Papadopoulos, S. Pletschacher, C. Clausner, and A. Antonacopoulos, "The impact dataset of historical document images," in *Proceedings of the 2Nd International Workshop on Historical Document Imaging and Processing*, ser. HIP '13. New York, NY, USA: ACM, 2013, pp. 123–130. [Online]. Available: http://doi.acm.org/10.1145/2501115.2501130

[27] J. Soo, R. Cathey, O. Frieder, M. Amir, and G. Frieder, "Yizkor books: a voice for the silent past," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 1337–1338.