Retrieval from Noisy E-Discovery Corpus in the Absence of Training Data

Anirban Chakraborty, Kripabandhu Ghosh and Swapan Kumar Parui
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute, Kolkata, West Bengal, India
{chakraborty.abhi89, kripa.ghosh, swapan.parui}@gmail.com

ABSTRACT

OCR errors hurt retrieval performance to a great extent. Research has been done on modelling and correction of OCR errors. However, most of the existing systems use language dependent resources or training texts for studying the nature of errors. Not much research has been reported on improving retrieval performance from erroneous text when no training data is available. We propose a novel algorithm for detecting OCR errors and improving retrieval performance on an E-Discovery corpus. Our contribution is two-fold : (1) identifying erroneous variants of query terms for improvement in retrieval performance, and (2) presenting a scope for a possible error-modelling in the erroneous corpus where clean ground truth text is not available for comparison. Our algorithm does not use any training data or any language specific resources like thesaurus. It also does not use any knowledge about the language except that the word delimiter is blank space. The proposed approach obtained statistically significant improvements in recall over state-of-the-art baselines.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering; Query formulation

General Terms

Algorithms, Legal Aspects

Keywords

Noisy Data, Co-occurrence, E-Discovery

1. INTRODUCTION

Erroneous text collections have posed challenges to the researchers. Many such collections have been created and researchers have tried several error modelling and correcting techniques on them. The techniques involve training models on sample pairs of correct and erroneous variants. But such exercise is possible only if the training samples are available. There are several text collections which are created directly by scanning hard copies and then OCRing them. The Illinois Institute of Technology Complex Document Information Processing Test Collection, version 1.0, referred to here as "IIT CDIP" [4] and informally in the TREC community as the "tobacco collection" is one such collection. It was created for the TREC Legal track. The text version was created by Optical Character Recognition (OCR) from the original document images. However, it contains a widespread of OCR errors which made it very difficult for the participants to achieve meaningful improvements 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 unavailability of training data presents a different problem premise. Parapar et al. [7] combined the results of 5, 4, 3 and 2-grams with the actual terms. 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 improvement over a weak baseline in MAP. Moreover, they did not consider any recall specific evaluation measure which is vital for a legal IR retrieval. Also, a major drawback with the n-gram based methods is the increase in the size of the inverted index, which poses serious resource crisis and also slows down the retrieval process considerably. Ghosh et al. [2] proposed an algorithm based on word similarity and context information and an improved version was proposed by Chakraborty et al. [1]. We consider the later version as a baseline in our paper. A string matching technique (e.g., edit distance, n-gram overlaps, etc.) alone is not reliable in finding 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 unrelated. Such mistakes are even so 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, context information can be used to get more reliable groups of erroneous variants. Context information can be harnessed effectively by word co-occurrence. We say that two words co-occur if they occur in a window of certain words between each other. Word co-occurrence has been successfully used in identifying better stems ([5], [6]) than methods that use string similarity alone [3].

SIGIR'15, August 09 - 13, 2015, Santiago, Chile.
© 2015 ACM. ISBN 978-1-4503-3621-5/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2766462.2767828.

The rest of the paper is organized as follows: we describe the proposed approach in Section 2, provide the results in Section 3, and conclude in Section 4.

2. PROPOSED APPROACH

We first describe two key terms used in our algorithm. Then we describe our algorithm.

2.1 Key Terms

2.1.1 Word Co-occurrence

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. In our experiment we take the whole document as the window. Word co-occurrence gives a reliable measure of association between words as it reflects the degree of context match between the words. This association measure gets more strength when it is used in conjunction with a string matching measure. For example, two words with high string similarity are 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* is highly similar to *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.

2.1.2 Longest Common Subsequence (LCS) similarity

Given a sequence $X = \langle x_1, x_2, ..., x_m \rangle$, another sequence $Z = \langle z_1, z_2, ..., z_k \rangle$ is 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. 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*, an LCS is *industr*. Now, we define a similarity measure as follows :

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

So, $LCS\_similarity(industry, industrial)$

$$= \frac{StringLength(LCS(industry, industrial))}{Maximum(StringLength(industry), StringLength(industrial))}$$
$$= \frac{StringLength(industr)}{Maximum(8,10)}$$
$$= \frac{7}{10}$$
$$= 0.7$$

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

2.2 The Proposed Algorithm

Our algorithm has two major parts :

1. Grouping, and
2. Binding

2.2.1 Grouping

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

- **Filtering** :

  Let L be the lexicon or the set of all unique words in the documents in $\mathbb{C}$. 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 word. We construct a set $L^{\alpha}_{w_i} = \{w \in L$: $LCS\_similarity(w, w_i) > \alpha\}$. In other words, $L^{\alpha}_{w_i}$ contains all the words in L that has $LCS\_similarity$ of more than a selection threshold α with the query word w_i. Since, α is a value of $LCS\_similarity$, α belongs to the interval $(0, 1)$.

- **Graph formation** :

  We now define a graph $G = (V, E)$ where V is the set $L^{\alpha}_{w_i}$ of words. For two words w_1, w_2 in $L^{\alpha}_{w_i}$, let $cf(w_1, w_2)$ denote the co-occurrence frequency over all the documents. If $cf(w_1, w_2) > 0$ then (w_1, w_2) defines an edge in E and the weight of the edge is defined as $cf(w_1, w_2)$.

- **Trimming** : Let the maximum edge weight of graph G be $max_e w$. Then, we eliminate those edges in G whose weight is less than β ($\beta \in (0, 100)$) per cent of $max_e w$. Let this new graph be called G_r. Then G_r = (V, E_r) where $E_r = \{e \in E$: weight of $e \geq \beta$ % of $max_e w\}$. This step is done 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-occurrence are very low. 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.

- **Clustering** : We now cluster the vertices of graph G_r based on the edge weights. The 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

  v_1 is the *strongest neighbour* of v_2 if out of all the neighbouring (adjacent) vertices of v_2, the edge weight of edge joining v_1 and v_2 is the maximum. This clustering algorithm was used by Paik et al. [6]. It is more convenient to use this algorithm over popular clustering algorithms like single-linkage, complete-linkage, k means and k nearest neighbour since it is parameter-free.

- **Weight assignment** : Given a cluster, the weight of each node is the degree of the node (i.e., the number of edges connected to the node). In Figure 1, we see a cluster containing the words *Tobacc*, *obacc* and *Tobac0*. The degrees of these three vertices are 2, 1 and 1 respectively. So, the weights of *Tobacc*, *obacc* and *Tobac0* are 2, 1 and 1 normalized into 0.5, 0.25 and 0.25 respectively.

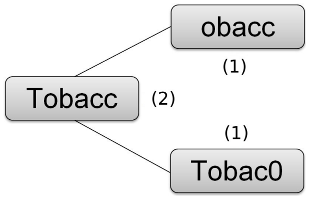

Figure 1: Weight assignment

2.2.2 *Binding*

Now, given a query word w_i, we need to find the 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 $\{(w_{j_1}, wt_{j_1}), (w_{j_2}, wt_{j_2}), .., (w_{j_m}, wt_{j_1})\}$, where (w_{j_t}, wt_{j_t}) is a word-weight pair in cluster Cl_j. In the clusters of $\mathbb{C}$, we look for the word that has the maximum $LCS\_similarity$ with w_i. Let $w_{closest} \in L^{\alpha}_{w_i}$ be the word such that $LCS\_similarity(w_{closest}, w_i) > LCS\_similarity(w_t, w_i)$, 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, we do not choose any cluster.

A pictorial view of the algorithm is provided in Figure 2. For a *Query Word*, we get a *Subset of Lexicon* after filtering out from *Lexicon* based on an α threshold. *Co-occurrence Pairs' Block* represents the repository of all the co-occurrence information between all the word pairs in the document collection. For the *Subset of Lexicon*, the *Co-occurrence Pairs' Block* is used to read the co-occurrence values for this subset of words and form the *Graph*. Next, the *Graph* is trimmed using the β threshold and we get *Trimmed Graph*. Then, *Trimmed Graph* is clustered to get *Clusters*. After *weight assignment*, we get *Clusters (weighted nodes)*. Now, for the *Query Word*, we choose the appropriate cluster from *Clusters (weighted nodes)*. We call this process *Binding*. The chosen cluster (if any cluster is chosen), along with the *Query Word*, forms the *Expanded Query*. The *Expanded Query* is then used for retrieval.

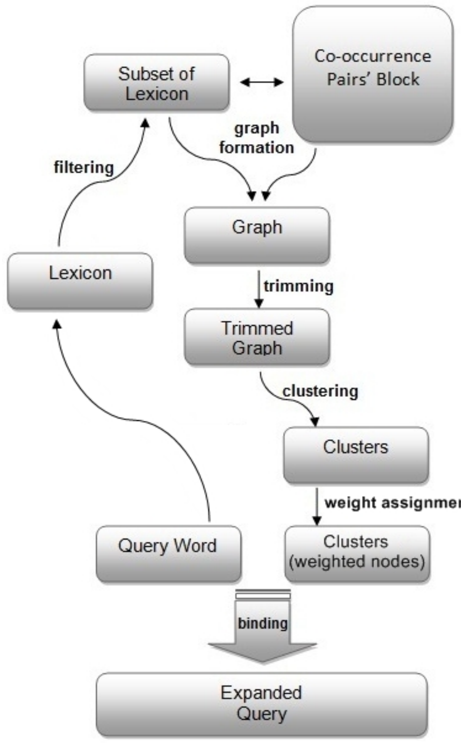

Figure 2: Algorithm : a pictorial view

3. RESULTS

We run our experiments on the IIT CDIP 1.0 dataset created for TREC Legal track. The collection statistics are shown in Table 1. We use the 43 topics of TREC Legal Ad Hoc 2007. We use only the **FinalQuery** field for our experiments. This query was prepared for Boolean retrieval and hence it contains Boolean operators like *OR* and *AND*. But, since we use it for ranked retrieval, we ignore the Boolean operators. We also convert the words in wildcard format like fertiliz!, phosphat!, agricultur! etc. to the shortest correct words like fertilize, phosphate, agriculture etc. respectively. The proximity operators like "w/15" are also dropped.

3.1 Performance

We have compared our approach with two baselines:

- The unexpanded query (*No Expansion*): Here retrieval is done solely on the basis of the original query terms. In other words, no query expansion is done here.
- Method proposed by Chakraborty et al. [1] (*KDIR*): Both co-occurrence and string similarity are used here. But the condition on co-occurrence of two words is quite relaxed resulting in more error variants.

Table 2 shows the results of the proposed approach. We see that the proposed approach produces superior performance in comparison to *No Expansion* and *KDIR*. The improvements in terms of MAP are only numerical. However, in Recall@100, the numerical improvements are also found to be *statistically significant* at 95% confidence level (p-value < 0.05) based on Wilcoxon signed-rank test [8]. The values of the parameters α and β are chosen by simple grid search in the intervals $(0, 1)$ and $(0, 100)$ respectively and the best result obtained is reported.

3.2 Error variants

Table 3 shows the error variants of some correct words in the corpus. We see that the proposed method is able to identify genuine error variants. For example, for the correct word *legal* the variants *egal*, *iegal*, *lega* and *legai* are identified. Several variants of *health* are also identified. All these variants are genuine error variants of *health*. Note here that LCS similarity between *health* and *iiealth* is 0.7143 and that between *health* and *wealth* (present in the corpus) is 0.8333. However, *iiealth* is identified as a variant while *wealth* is not. This is because the first pair has high co-occurrence frequency while the second pair has low co-occurrence frequency. So, both LCS similarity and co-occurrence have a role to play in identifying the error variants of a word. The proposed method is thus useful in the cleaning and error-modelling of the corpus where the clean text samples are not available for comparison and error-modelling.

	MAP	Recall@100
No Expansion	0.0899	0.1574
KDIR	0.0898	0.1658
Proposed	**0.0947** (+5.34%, +5.45%)	**0.1741**^{nk}(+10.6%, +5%)

Table 2: The table shows comparison of the proposed method with No Expansion and KDIR. Bold font shows the best value. Percentage improvements are shown in bracket. The super-scripts show significant improvements. The first letter of the less effective method appears in the super-script.

Term	Error variants
tobacco	lobacco, obacco, tabacco, tobacc, tobacca, tobaceo, tobacoo, tobaeco, tobaoco, tobecco, tohacco
smoking	smcking, smeking, smnking, smokiag, smokin, smokina, smokinc, smokine, smokinp, smokinq, smokins, smoktng, smolcing, smuking, snoking
mice	micee, miice, milce
oncology	0ncology, oncoiogy, oncolog, oncologv, oneology, onoology
polonium	olonium, poionium, polonlum
sewage	ewage, sewaqe
legal	egal, iegal, lega, legai
health	iiealth, heaith, hcalth, healt, healthy, bealth, healtb
securities	curities, secunties, securitles, seeurities
insurance	insuranee, lnsurance, nsurance
natural	atural, natura, naturai, naturall

Table 3: Error variants : IIT CDIP 1.0

Feature	Value
number of documents	6,910,192
number of tokens	6,524,494,574
number of unique terms	135,985,661
average document length	944.184

Table 1: Collection statistics for IIT CDIP 1.0

4. CONCLUSIONS

In this paper we have proposed 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 algorithm to solve the problem on a large and extremely noisy E-Discovery corpus. The results show that we have achieved statistically significant improvements over the baselines in recall. Also, we have shown that the use of context information is extremely beneficial and reliable in agglomerating semantically related erroneous variants. In addition, the automatic identification of error variants lays a sound platform for the modelling of error patterns for the text corpora where the clean text sample is not available for comparison. We have seen that it is feasible to obtain useful error variants of the terms in the noisy corpus. This may be extremely useful in automatic cleaning of the IIT CDIP 1.0 corpus which, in its cleaned version, will be very important to the E-Discovery community.

5. REFERENCES

[1] A. Chakraborty, K. Ghosh, and U. Roy. A word association based approach for improving retrieval performance from noisy ocred text. KDIR '14, pages 450–456, Rome, Italy, Oct. 2014. SCITEPRESS.

[2] K. Ghosh and A. Chakraborty. Improving ir performance from ocred text using cooccurrence. *FIRE RISOT track 2012 working notes*, Dec. 2012.

[3] P. Majumder, M. Mitra, S. K. Parui, G. Kole, P. Mitra, and K. Datta. Yass: Yet another suffix stripper. *ACM Trans. on Information Systems*, 25(4):18:1–18:20, Oct. 2007.

[4] D. W. Oard, J. R. Baron, B. Hedin, D. D. Lewis, and S. Tomlinson. Evaluation of information retrieval for e-discovery. *Artificial Intelligence and Law*, 18(4):347–386, 2010.

[5] J. H. Paik, M. Mitra, S. K. Parui, and K. Järvelin. Gras: An effective and efficient stemming algorithm for information retrieval. *ACM Trans. on Information Systems*, 29(4):19:1–19:24, Dec. 2011.

[6] J. H. Paik, D. Pal, and S. K. Parui. A novel corpus-based stemming algorithm using co-occurrence statistics. In *ACM SIGIR*, SIGIR '11, pages 863–872, 2011.

[7] J. Parapar, A. Freire, and A. Barreiro. 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, pages 680–684, Berlin, Heidelberg, 2009. Springer-Verlag.

[8] S. Siegel. *Nonparametric statistics for the behavioral sciences.* McGraw-Hill series in psychology. McGraw-Hill, 1956.