IJCLA VOL. 1, NO. 1-2, JAN-DEC 2010, PP. 305-317
RECEIVED 30/10/09 ACCEPTED 16/01/10 FINAL 08/03/10

A Case Study of Rule Based and Probabilistic Word Error Correction of Portuguese OCR Text in a "Real World" Environment for Inclusion in a Digital Library

BRETT DRURY AND J. J. ALMEIDA

LIAAD-INESC and University of Minho, Portugal

ABSTRACT

The transfer of textual information from large collections of paper documents to electronic storage has become an increasingly popular activity for private companies and public organizations. Optical Character Recognition (OCR) software is a popular method to effect the transfer of this information. The latest commercially available OCR software can be very accurate with reported accuracy of 97% to 99.95%[6]. These high accuracy rates lower dramatically when the documents are in less than pristine condition or if the typeface is non-standard or antiquated. In general, OCR recovered text requires some further processing before it can be used in a digital library. This paper documents an attempt by a private company to apply automatic word error correction techniques on a "real world" 12 million document collection which contained texts from the late 19th Century until the late 20th Century.

The paper also describes attempts to increase the effectiveness of word correction algorithms through the use of the following techniques: 1. Reducing the text correction problem to a restricted language domain, 2. Segmenting the collection by document quality and 3. Learning domain specific rules and text characteristics from the document collection and operator log files. This case study also considers the commercial pressures of the project and the effectiveness of both rule based and probabilistic word error

correction techniques on less than pristine documents. It also provides some conclusions for researchers / companies considering multi-million document transfers to electronic storage.

1 INTRODUCTION

Real world document collections are not always in pristine condition. The document may have surface contamination which can be due to the age of the document, the quality of the media, the type of media and other material affixed to the document such as official stamps. The typeface may be antiquated which may further degrade the accuracy of OCR software. The recovered text may contain too many errors to be used in a digital library. Frequently, some further correction of the text is required. This paper will describe an attempt by a commercial company to correct Portuguese text which had been recovered by OCR software for inclusion in a digital library.

1.1 *Document Collection Characteristics*

The document collection contained over 12 million documents which was created over a hundred year period. The quality of the documents ranged from the very good (clear type face and no surface contamination) to the very poor (illegible and heavy surface contamination). The collection contained some homogeneous text, for example correspondence. The correspondence was mainly letters, which on occasion had images as an attachment. This correspondence also included bill and product information which in some circumstances was in a language other than Portuguese. The document collection also contained some non-standard items such as reports.

1.2 *Processing Documents*

The paper documents were scanned using large commercial scanners which were capable of processing a large number of documents per hour. The scanners produced images of documents in Tagged Information File Format (TIFF) and were in monochrome. The images were then sorted by a simple algorithm and organized into folders which contained related

images. Each image was given a unique number within the folder. The images were then pre-processed (deskew and despeckle) in preparation for the OCR process. The images were processed by the OCR software which ran on two powerful computers which functioned 24 hours a day. The OCR software was set on the slowest and most accurate setting. The OCR software required nearly 2 years to process the 12 million documents. The text from the OCR process was inserted into a Database Management System (DBMS). The text was then subject to a post-processing correction process. The text was to be used in a full text index which would be used for searching, consequently stop words such as "por" could be excluded from the correction process.

1.3 *Initial Correction Attempts*

A popular approach is to use human operators to correct text. This can be slow. It was reported that an efficient company in Romania with 25 staff could process 600,000 documents a year.[10]. This mirrored our initial experience with a completely manual approach. A software application was built which used the Microsoft Word API to identify word errors and their possible replacements. The operator corrected the text one word error at a time. The mean time for each operator to correct one document was approximately 180 seconds. This was too slow as it would have taken a team of 5 operators approximately 72 years to complete the task. This was not only unacceptably slow, but would have represented a potential enormous cost to the company.

The operators mean time to process each document was reduced with a modified manual approach, which was to correct popular spelling and characters errors automatically. The performance of the application was increased by a multi-threaded approach. The errors and potential word candidates were cached by one thread, whilst another thread updated the user display whilst another thread was responsible for updating and fetching text from the DBMS. The error caching thread was significantly faster than the human operator, consequently there was no delay when moving from one error to the next.[1] Fetching the error and word candidates di-

[1] Although this improved the operators' mean time, the operators found it difficult to work with the application as the operators had to concentrate 100% of the time. If I were to write the application again I would add random delays to give the operators a small break in concentration.

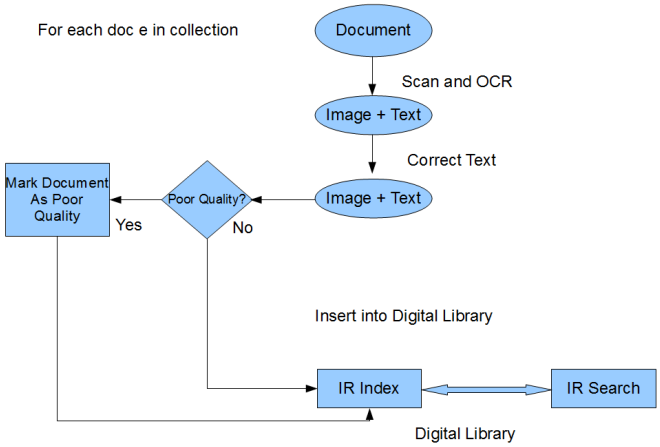

Fig. 1. Process For Transferring Documents

rectly from the Microsoft Word API without caching introduced a significant delay. The mean time was reduced to 30 seconds a document. A team of 5 operators working full-time could process a million documents a year, which was significantly faster than the Romanian case study[10]. This efficiency improvement was still not fast enough as it would have taken 12 years to complete the task and would have represented a cost of approximately 500,000 Euros in labour. An automated process was required to process a significant number of texts, not only to reduce the time required to complete the project, but to ensure the company realised a profit from the project.

It should be noted that the operators required significant supervision. There was pressure for each operator to reduce their times to process each text. The less able operators simply cheated by marking documents as complete when the document had not been processed or marking a good document as unprocessable. This would lower the mean time of the operator. It was necessary to review at regular intervals a statistically significant sample of each operators output to identify which operators were "honest" and which were "cheating".

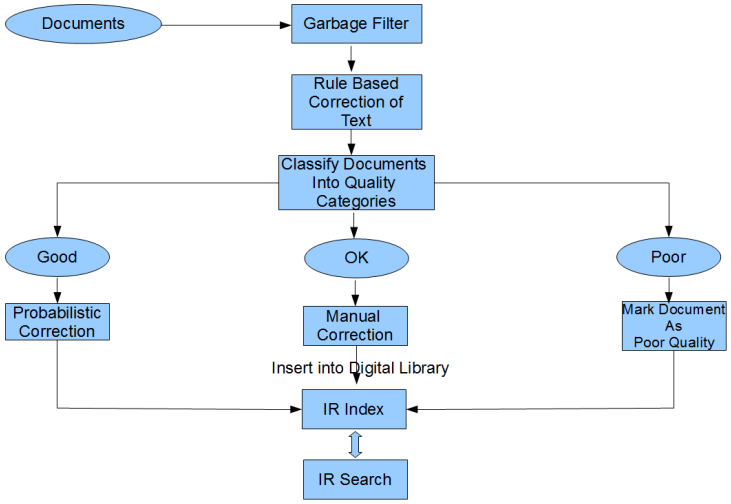

Fig. 2. Modified Process For Transferring Documents

1.4 *Summarization of Problem*

The initial process is described in figure: 1. This process was too slow and costly. There was a demand to move to a partial automatic system, as described in figure: 2. The rest of the paper will discuss the transformation to the automatic word correction process as described in figure: 2. This will include:

1. Reducing the text correct problem to a restricted language domain.
2. Segmenting the collection by document quality.
3. Learning domain specific rules and text characteristics from the document collection and operator log files.

2 Document Collection Preparation

2.1 *Assessing The Document Collection*

A "quality measure" was assigned to each document, so that it was possible to measure the performance of the error correction techniques. A

simple measure was used, which was the number of correct words divided by the total number of words. A statistically significant sample of the documents was manually verified against the quality score. This simple measure provided an accurate reflection of the document's quality. Low scoring documents had heavy surface contamination or were handwritten. High scoring documents were free from contamination with a clear typeface. The distribution of quality effectively followed a normal distribution, with the bulk of the documents having a quality score between 0.5 and 0.7.

$$Quality = NumberofCorrectWords \div TotalNumberofWords$$

2.2 *Rule Induction from Operator Generated Log Files*

The five employees had processed the document collection with the modified manual system for three years. Three million documents were processed. The spelling corrections were logged for each operator. It was possible to categorize the error types from the log files into the following categories: 1. Substitution of Characters, 2. Elimination of Characters, 3. Insertion of Characters, 4. Split Word Errors, 5. Joined Word Errors. A number of frequently occurring errors were unique to a Latin based language, for example, the inaccurate splitting or joining of reflexive words, for example "da-me" would be joined as "dame".

2.3 *Pre-processing of Text (rule based correction)*

GARBAGE REMOVAL A large number of documents were printed on ruled paper which was interpreted by the OCR software as miscellaneous symbols. A filter was constructed which attempted to remove text which was generated by physical markings other than text.

JOIN/SPLIT WORDS A number of rules to detect and correct split and join errors were inferred from the log files. Join errors were detected by identifying "word boundaries" in continuous text, for example capital letters or punctuation. The text was split on the word boundary and the resulting words checked against a dictionary. If they were both correct than the words were accepted. Split errors were detected by joining two continuous errors and evaluating the resultant text with a dictionary. If the text was a correct word then it was accepted.

CORRECTING COMMON WORD AND CHARACTERS ERRORS The join and split word rules were incorporated into a pre-processor application with the hard coded rules from the modified manual system for popular word and character substitutions. Two runs were made, the first was "strict" where the resultant words had to be correct. The second was "permissive", where there was a tolerance of one edit distance. The preprocessor was relatively successful and moved the "bulge" of the normal distribution for the quality to the right with a mean average improvement of quality of 0.2, i.e on average a document which scored 0.5 would score 0.7 after the pre-processor runs.

3 PROBABILISTIC ERROR CORRECTION

In recent years there has been a number of advances in probabilistic error correction for text produced by OCR systems. These techniques assume that text recovered by OCR is semi-determinate[9]. The assumption is that OCR systems will consistently identify identical/similar markings on a document as the same character. This semi-determinate nature allows a certain degree of predictability of the errors produced by the software and that some types of errors are more frequent than others.

The following three techniques were utilized in this case study.

CHARACTER CONFUSION MATRIX A character confusion table provides a list of transformation probabilities from one character to another, for example c $\rightarrow$ ç would be high where as c $\rightarrow$ w would be low. A probability of a word candidate substituting an error was achieved by a simple summing of the individual character probabilities and calculating the mean value [3].

The character confusion matrix in this project was built from the operator log files which documented all word error changes over a three year period. The substitution errors were calculated by comparing error and correction words of the same length. Insertion and deletion errors were calculated by comparing error and correction words which had a difference in length of 1 character.

DICTIONARY THINNING Dictionary thinning allows the reduction of possible word candidates. A custom dictionary was developed which con-

tained only the correct words which were in the document collection and their frequency. The frequency was important because word frequency in a document collection obeys Zipf distribution [7] and may provide an indication of likelihood of the word candidate being correct [8].

The dictionary was constructed by parsing the whole document collection and comparing the words to the J-Spell dictionary. The words which were not in the J-Spell[2] dictionary were initially rejected and written to a file with their frequency. The remaining words were written to another text file which was our initial dictionary. The top 1,000 most frequent errors were analysed by a human operator. The operator identified words which were incorrectly rejected, for example surnames and names of companies. These words were reintroduced into the dictionary.

WORD N GRAMS Word n-grams provided an indication of conditional probability of certain word combinations[5]. Words frequently co-occur, consequently the presence of one word may imply the presence of another word. In the case study another measure was developed, the gapped bi-gram where the middle word from a tri-gram was removed. The gapped bi-gram assisted in the identification of conditional probability of words separated by a stop-word, for example "agua da pedras", where there is a semantic relation between "agua" and "pedras". To generate the n-gram dictionaries the whole corpus was parsed. The n-grams were listed by frequency and the top 2,000 n-grams were selected for their relevant dictionary.

3.1 *Selection of Word Candidates*

Word candidates were selected from the customized dictionary as described in the above section. Although the dictionary had been "thinned", it still contained thousands of possible word candidates. It was not possible to assess each word in the dictionary for each error because the application would have been too slow. Consequently, a reduction of the number possible word candidates would improve the efficiency of the application. A common method is to use n-grams [4] to retrieve word candidates for a given error. Popular letter n-grams however, can lead to large numbers of word candidates being retrieved for a single error. It is possible to reduce the number of word candidates without removing any highly probable replacement through the use of skip grams.[1] Skip

Table 1. character error & replacement character & probability

Character Error	Character Replacement	Probability
c	ç	6.5%
a	ã	5.8%

grams are formed from letters which occupy either odd or even numbered positions in a word, for example the word "teste" would have the following 2 letter skip gram "ts et se". The popular letter n-grams were broken up into less popular skip grams and consequently when the word candidates were returned through the application of a skip gram distance a smaller and more relevant set was returned.

The use of skip grams highlighted a "quirk" of the OCR system. The OCR software frequently failed to recognize the Portuguese characters ’ç’ and ’ã’. It frequently replaced them with the characters ’c’ and ’a’. This was a significant error as ’ç’ and ’ã’ frequently appear together in Portuguese. This mistake would result in two incorrect skip grams, which may have excluded a valid word candidate from being selected.

The frequency of this mistake is shown in Table 1.

Note: These figures understate how often the OCR software made these mistakes as these figure were taken after the pre-processors had corrected the common character errors.

3.2 *Alignment of Word Candidate and Error*

The calculation of the transformation probability of error to word candidate required alignment of the word candidate and the error. This was a trivial task, if the word candidate and error were the same length. When the word candidate and error were different lengths it was necessary to return the most probable alignment with a ’#’ representing the missing character(s). There were two considerations for the algorithm design, which were accuracy and efficiency. Two algorithms were developed, one algorithm was for when the difference in length between the error and word candidate was 2 or less and the other was when the difference in length was 3 or more. The first algorithm calculated every alignment permutation and returned the most probable. The second algorithm was a compromise between accuracy and efficiency this was because the larger

the difference the more the total permutations and consequently there would be a drop in performance of the algorithm. A "sliding alignment" algorithm[8] was used where the shorter word would be moved across the longer word one character at a time. At each stage the alignment would be verified for successfully aligned characters. The word form with the most correctly aligned characters was returned.

3.3 *Calculating the Word Candidate Scores*

The scoring process initially applied a transformation probability for each of the word candidates. Word candidates were eliminated if they had less than a 0.5 transformation probability. This was because through experimentation with a statistically significant sample it was determined that word candidates with a score of less than 0.5 were unlikely to be correct. Elimination was necessary to improve the efficiency of the application. The remaining word candidates were scored for their co-occurrence probabilities with existing word n grams and gapped n-grams and the log frequency of the word candidate in the corpus was calculated as follows:

$$S = P(E \rightarrow W_c) \times (log(W_cF) + 50) \times (1 + P(X, W_c)) \times (1 + P(Y, W_c))$$

S =Score W_c =Word Candidate E =Error
W_cF =Word Candidate Frequency
X =Word which has position $\pm$ 1 of E
Y =Word which has position $\pm$ 2 of E

3.4 *Excluding Documents*

Automatic processing of the whole collection was not possible because the document collection was not of a uniformly high quality. It was possible to automatically process a large number of documents, which reduced the number of documents which needed to be processed manually. This reduced the time that was required to process the documents, but also reduced the costs involved.

The documents were classified into three categories:poor quality (no manual processing possible), low-medium (manual processing only) and medium-high quality (automatic processing possible). The quality borders where set by operators who analysed a statistically significant sample of documents at varying quality levels. The quality measures are shown in Table 2.

Table 2. Document Classification

Category	Quality measure	Action
Poor quality	$0 < q < 0.5$	no processing possible
Low - medium quality	$0.5 \leq q < 0.7$	manual processing possible
Medium - high quality	$0.7 \leq q < 1$	automatic processing possible

Poor / low quality documents had surface contamination, degraded document media and obscure or unclear typefaces which provoked an erratic response from the OCR software. In some circumstances the document was too degraded to perform any form of manual correction or re-keying. There were other documents where Tong's assumption[9] that OCR software is semi-determinate system no longer held, but were of sufficient quality to be re-keyed or manually corrected. The exclusion of documents on which probabilistic methods would function poorly allowed the algorithm to process "good quality" documents where there was sufficient certainty that the results would be acceptable. The operators worked on the remaining documents.

4 RESULTS

The probabilistic approach worked well on word errors which were not the result of errant splitting and joining and had a small number of character errors. The probabilistic approach functioned adequately on words with a larger number of character errors, however there were a significant number of incorrect choices which declined with the increasing length of the error. The same results were gained with split and joined word errors. The probabilistic approach functioned well on good quality documents because they contained more errors with a small number of character errors. Accuracy declined rapidly with decreasing document quality because of the increased number of split and joined words errors as well errors with increased number of incorrect characters. The probabilistic approach was tested on a documents which were earmarked for manual processing only and for a large number of errors there were no suggested replacements.

The rule based pre-processors corrected a larger proportion of errors than the probabilistic technique because error frequency followed a Zipf

distribution, consequently common errors constituted a very large proportion of the total error count.

This approach reduced the number of documents that needed to be processed from 9 million to 2.2 million. Approximately 5 million documents were processed by probabilistic methods and 1.8 million were rejected as too poor to process. This allowed the reduction of time required to transfer the remaining documents to electronic storage from 9 years to 2 years. It had taken the previous three years to process three million documents manually. If the project had been approached in this manner from the beginning it is estimated that the total project length would have been less than 3 years, which was the original project estimate. The project was two years late.

5 CONCLUSION

Transferring large numbers of less than pristine documents to a digital library / storage with a high degree of accuracy is a time consuming process. Manual correction / re-keying is only feasible if there are sufficiently large numbers of staff or the document count is reasonably small. Probabilistic methods work well on pristine documents with errors which have a low number of character errors, but their performance declines dramatically as media quality drops. Rule based methods are more robust as quality declines. Error frequency follows a Zipf distribution, consequently correcting common errors will have disproportionate effect on document quality. Portuguese has it's own unique challenges with accents and the "ce de cedilha (ç)" which OCR software frequently misinterprets.

Companies which attempt to transfer large numbers of documents to electronic storage via OCR software with the text requiring certain degree of accuracy should consider automatic methods of correcting text. The reduction of the time required for manual processing equates to a saving in costs which will pay a programmers time in constructing the text correct algorithms. Automatic text correction should be considered from the beginning of the project, not when the project is in obvious trouble. The economic case of automatic text correction methods increases with the size of the document collection.

REFERENCES

1. Pirkola A, Keskustalo H, Leppanen E, Kansala A., and Jarvelin K. Targeted s-gram matching: a novel n-gram matching technique for cross- and monolingual word form variants. Information Research, 2002.
2. J.J. Almeida and Ulisses Pinto. Jspell – um módulo para análise léxica genérica de linguagem natural. In Actas do X Encontro da Associão Portuguesa de Linguística, pages 1-15, 1994, 1995.
3. Eric Brill and Robert More. An improved error model for noisy channel spelling correction. In Annual Meeting of the ACL.
4. Ethan Miller Dan, Dan Shen, Junli Liu, Charles Nicholas, and Ting Chen. Techniques for gigabyte-scale n-gram based information retrieval on personal computers. In Personal Computers, Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 99), Las Vegas, NV, 1999.
5. S. M. Harding, W. B. Croft, and C. Weir. Probabilistic retrieval of ocr degraded text using n-grams. In Research and Advanced Technology for Digital Libraries.
6. JSTOR. JSTOR OCR rates. fsearch-sandbox.jstor.org/about/images.html, consulted in 2008.
7. W. Li. Random texts exhibit zipf's-law-like word frequency distribution. In Information Theory, IEEE Transactions on.
8. Lasko TA and Hauser SE. Approximate string matching algorithms for limited-vocabulary ocr output correction. In Proceedings of SPIE.
9. Xiang Tong and David A. Evans. A statistical approach to automatic ocr error correction in context. In Proceedings of the Fourth Workshop on Very Large Corpora (WVLC-4, pages 88–100, 1996.
10. I. Written and D. Bainbridge. How to Build a Digital Library. Morgan Kaufmann, 2003.

BRETT DRURY
LIAAD-INESC, PORTUGAL
E-MAIL: <BRETT.DRURY@GMAIL.COM>

JOSE JOÃO ALMEIDA
UNIVERSITY OF MINHO, PORTUGAL
E-MAIL: <JJ@DI.UMINHO.PT>