FILLING THE GAPS USING GOOGLE 5-GRAMS CORPUS

Costin-Gabriel Chiru, Andrei Hanganu, Traian Rebedea and Stefan Trausan-Matu
"Politehnica" University of Bucharest, Department of Computer Science and Engineering
313 Splaiul Independentei, Bucharest, Romania

Keywords: Text Recovery, OCR, Natural Language Processing, Probabilistic Parsing, N-grams.

Abstract: In this paper we present a text recovery method based on a probabilistic post-recognition processing of the output of an Optical Character Recognition system. The proposed method is trying to fill in the gaps of missing text resulted from the recognition process of degraded documents. For this task, a corpus of up to 5-grams provided by Google is used. Several heuristics for using this corpus for the fulfilment of this task are described after presenting the general problem and alternative solutions. These heuristics have been validated using a set of experiments that are also discussed together with the results that have been obtained.

1 INTRODUCTION

Lately, there have been a lot of attempts to digitize the content of some publications – the Gutenberg Project (http://www.gutenberg.org/), the Runeberg Project (http://runeberg.org/), or even Google Book Search (http://books.google.com/) – in order to increase their availability to the public and to give them the possibility of not being forgotten, as signalled in Baird (2003). The easiest and cheapest way to do that is to convert the printed papers to a digital format using OCR (Optical Character Recognition). The problem with this approach is that some publications are very old, written on cheap or partially damaged paper and therefore the quality of the digital documents produced by the OCR is not very good. In this paper, we propose a text recovery method based on a probabilistic post-recognition processing that tries to identify which are the words that are missing from the electronic form of the document. Our method uses the n-grams from the "Web 1T 5-gram Version 1" corpus (Brants and Franz, 2006) to predict the words that could fill in the spaces that have appeared because the words were not recognized from the original scanned documents. In the next section we shall present a short overview and related work in the domain of OCRs. The proposed approach is presented in the third section. Finally, Section 4 presents a set of experiments undertaken to validate our approach. The paper ends with conclusions and further improvements.

2 RELATED WORK IN IMPROVING OCR ACCURACY

The OCR scanning process is affected by two major factors: the document and the OCR device. The document which is subject of digitization has the biggest impact over the precision of the conversion. An analysis of how the characteristics of a document may affect OCR accuracy is discussed in (Nagy et al., 2000). Since the quality of the paper cannot be improved, some researchers tried to pre-process the documents in order to allow a better tuning of the set of the OCR attributes (Khoubyari and Hull, 1996): the resolution of the scanner measured in DPI, and the colour depth which can be either greyscale or colour, with different bit depths.

The text recognition algorithm has also been intensely improved. An improvement direction was based on more precise mapping of symbols to characters. One example for this tendency was presented by Breithaupt (2001) who used a voting system between several OCR devices in order to determine the best mapping. Another example was given by (Hong and Hull, 1995) that employed a method for identifying images depicting similar substrings, this way allowing the elimination of some of the mapping problems. The other direction refers to the post-processing of the converted text in order to search and correct the spelling errors. The automatic word correction focuses on three problems as shown in Kukich (1992), non-word error detection, isolated-word error correction and

Chiru C., Hanganu A., Rebedea T. and Trausan-Matu S. (2010).
FILLING THE GAPS USING GOOGLE 5-GRAMS CORPUS.
In *Proceedings of the 5th International Conference on Software and Data Technologies*, pages 438-443
DOI: 10.5220/0002932204380443

context dependent error correction. In order to correct such errors, powerful language processing tools are needed. Examples of such attempts are presented in (Meknavin et al., 1998 and Tong and Evans, 1996), where sequences of parts of speech are evaluated for likelihood of occurrence and unlikely sequences are marked as possible errors.

3 A STATISTICAL APPROACH FOR SOLVING THE OCR GAPS PROBLEM

Unlike most of the research that is focused on improving the detection rate of characters, in this paper we are focusing on a different aspect: the recovery of text that cannot be recognized, either because it is too damaged or simply missing. This paper tackles the issue of the reconstruction of damaged documents based on the prediction of the most plausible word sets that could fill in the missing areas that resulted from the impossibility of recognizing the original words used in the documents. From now on, these missing areas will be referred to as "gaps". Every gap has a very important property that is the most important factor which influences the accuracy of the recovery process: its dimension, usually expressed by the number of characters or words if we consider the text under analysis as a continuous stream of text.

The solution that we propose in this paper is intended for the recovery of text chunks that represent pieces of phrases from the original document and it is based on two assumptions. The first one is related to the intra-document similarity: we assume that a model of the document can be built based on the existing text and that the missing text also respects this model. We considered that the document model has two components: the style model, representing the structure of the text and the language model, depicting the vocabulary used by the author, the n-grams that were built with these words and the frequency of the n-grams. These two models are combined in order to identify the word sets that could fit in the gaps. Two heuristics have been developed to allow us to benefit from the style model. Regarding the language model, there is a problem that sometimes new words that haven't been used before in the document could appear in the gaps, but these words cannot be discovered using only the language model of the document, since these words are simply missing from it. This problem leads us to the use of the Google corpus and to the second assumption: the corpus dimension is large enough to subsume most of the language models of the documents posted on the Internet and in the meantime, any word that does not appear in this corpus, should not be considered as a possible candidate to fill in the gaps.

Considering these two assumptions to be true, our solution starts with the identified gaps and follows a few steps in order to identify the missing words. First of all, the style model of the document is used in order to identify the dimension of the gap. Therefore, we consider two heuristics: estimated character count and estimated word count. The estimated character count is a numeric value which is determined based on the margins and indentation of the recovered document format, on the existing characters that were correctly identified and that are in the gap's vicinity and on some statistical information regarding the document under analysis (mean and deviation of the number of characters per phrase). This value is used to determine a maximum and a minimum number of characters that could fill in the gap. The estimated word count is also a numeric value, which uses the estimated character count and some statistical information regarding the mean and deviation of the number of characters per word and the mean and deviation of the number of words per phrase observed in the document. This value is used to determine a range for the number of words that we are looking for in order to fill in the gap.

Once having estimated the number of words we are looking for, we are able to start using the language model. At this point, there are a couple of heuristics that can be used. First of all, the gaps do not usually start or end with whitespace characters representing the limit between distinct words, so one could scan the document for partial words at the beginning or at the ending of the gaps. Using both the n-grams corpus and the words that have been correctly identified before and after the gap, it is easier to detect the whole words starting from the characters representing parts of them. Since the maximum dimension for n-grams in the corpus is 5-grams, the detection starts from the previous four words before the gap in order to identify the first word missing from the gap. We consider that these four words represent the starting words from a 5-gram, and we try to identify which is the most probable word to follow this combination. The same method is applied to the next four words after the gap in order to determine the last word missing from the gap, considering that these words represent the ending words from a 5-gram, and trying to detect the most probable word to precede them. If there is no 5-gram that is composed of the four words preceding or following the gaps, the same method can be used for the 4-grams, considering only three words from the text, and not four like before. This decrease in the number of considered words can go down to bigrams, where only the next word after the gap or the previous one before it is considered. The same decrease in the order of the considered grams can be generated by the lack of words between the beginning of the phrase and the starting of the gap or between the ending of the gap and the ending of the phrase. In such cases, only the amount of words that can be found near the gap is used and the order of the n-gram is reduced accordingly. All the possible candidates for the first and last position in the gap are stored and then the process is restarted for every one of these candidates using the same methodology. This way the identification of the missing words starts from both ends hoping to merge in the middle. The process will be repeated in the same manner for all possible branches until one of the following events occurs for a specific branch:

- The number of words or characters from the left-side and/or the right-side branch do not respect any more the heuristics built on the estimated word count or the estimated character count. This means that branches are too long to be valid candidates, and therefore these branches can be discarded.
- A left-side branch matches at some point a right-side branch. This means that at a moment in time, the last token added to the left-side branch will be the same as the mirrored last token added to a right-side branch, therefore identifying a valid candidate for the missing words.
- The left-side branch has reached an end sentence mark-up (</S>) and the right-side one has reached a beginning of sentence mark-up (<S>). At this point a "partial match" has been obtained, which contains a possible unrecoverable gap inside it. Such an inside gap can be disregarded if the added size of the branches fits in the estimated character and word count, and therefore it can be considered a valid candidate.

At some points, some branches will not return any possible completion values for the order of the n-gram used at that point. The first thing to be done is to use a lower-level n-gram until a reasonable number of candidates are obtained or until reaching the bigrams. Although this is a problem, much more often the opposite situation occurs: a very large number of candidates are generated for each possible word. Considering that *no* is the estimated word count and that *min* is the minimum number of candidates generated for each of the *no* positions of the gap, around min^{no+1} candidates are generated. Since the number of the generated candidates is exponential, this process is time and space consuming, and some improvements have to be made. One idea that could reduce the space of the candidates is to consider the words' part-of-speech (called POS in the rest of the document) and to build a heuristic that can predict the POS of the expected word. If the candidate word doesn't have the expected POS, then it can be discarded. The faster a word is discarded, the more reduction it causes. In a similar way, semantic relations with the context of the gap are exploited.

After the generation of the valid candidates, the most probable solution must be chosen. The filtering from the other possible candidates is done based on a set of scores computed for each branch according to some heuristics. One of the possible heuristics regards the frequency of the n-grams that are built in the process of words' identification. The branches containing n-grams with higher frequencies should have a higher score, since those combinations are more probable and are preferred to other less probable combinations. Another heuristic is related to the distance between the ends of the gap and the current word, counted as number of words. This heuristic should give higher scores to the words closer to the ends of the gaps, which means that the earlier a word has been found, the more score gain it produces, since the words that are used to discover this new word are more reliable than the words that are discovered later in this process and are used for the discovery of the other words. Finally, the length of the identified branches should be considered, by normalizing the scores given by the words from each branch. After all the scores have been computed, the branch with the best score is chosen.

4 EXPERIMENTS AND RESULTS

In order to test the accuracy and the success rate of the system we started from complete documents and simulated the results of an OCR given the paper quality is very bad. For this simulation, various sections of text have been removed from the original document. The next step was to fill in the resulting gaps and to compare the generated solution with the initial text.

In this section we will present some of the tests that we made starting from the transcript of the

Wikipedia webpage about Literature: http://en.wikipedia.org/wiki/Literature. We considered this document for two important reasons: the vocabulary that is used in this document is not general, but domain specific and because it is available on the Internet, there are better chances that the n-grams of the document are found in the corpus. From this document we have randomly chosen the next phrase, eliminated the 5th, 6th and 7th words – "interpretation is that", and replaced them by <gap>:

”An even more narrow **interpretation is that** (<gap>) text have a physical form, ...”

Then, the text has been tokenized in the same way the Google corpus also has, so that the compatibility between our text and the corpus to be maximized. The next step was to use the TreeTagger (Schmid, 1994) in order to annotate the phrase with POS. The results show the words, their most probable POS and their lemma.

“An DT an
even RB even
more RBR more
narrow JJ narrow
<gap> NN <unknown>
text NN text
have VBP have
a DT a
physical JJ physical
form NN form
, , ,”

At this point we detect the gaps from the text and store the basic information related to each of them: the starting position in the document, the expected number of characters and words, the words found before and after the gap.

Initially, the number of expected words and characters is not defined but it will be computed after the statistics of the document are determined and these values are evaluated.

Once these numbers have been determined and having the above information related to the gaps, the generation of the candidate n-grams starts. Initially, the 5-grams corpus is interrogated in order to detect the 5-grams that have “an even more narrow” as their first 4 words. Since no result has been found for 5-grams, the next step is to lower the n-grams order and to look in the 4-gram corpus with the text “even more narrow”. After finding no results in this corpus, the search continues in the trigram corpus with the words “more narrow”, and 168 hits are found. Out of these, the results containing symbols, punctuation marks or words with less than 256 appearances in the corpus have been filtered out, remaining only 22 results, the top 6 being presented below:

[3] and [4816] [CC : 0.527744] [-1]
[3] approach [399] [NN : 0.885605] [5]
[3] as [372] [IN : 0.829617]
[3] definition [1934] [NN : 1.221063] [1]
[3] focus [2276] [NN : 1.057171] [11]
[3] interpretation [583] [NN : 1.221063] [4]

The first number ([3]) represents the number of words that still have to be found in order to fill the gap completely. This number is the same for all the words generated in a step and is decreased with the advance in the depth (with each word that fits in the gap). Once it reaches 0, no requests for new words are done and the suggestion for filling the gap is chosen from the resulting paths.

The second element of each entry is the word that fits in the n-gram, along with its frequency from the Google corpus.

The next information is related to the POS of the candidate word and the probability of finding an n-gram composed by the POS of the previous n-1 words and the current one. The POS n-grams probabilities are computed based on the words found in the document, considering the POS instead of the words.

Finally, the last number is a score given to the candidate word representing how well it fits in the context from the semantic point of view. This score is determined using the lexical chains that are computed based on the WordNet lexical database and the words from the text. The higher this score is, the better the word is suited to the meaning of the words in the document. Nevertheless, the lexical chains emphasize on the meaning of the words and thus they eliminate most of the functional words. In order to give this particular type of words a fair chance, they have been introduced in a special list, and their relevance according to WordNet has been set to -1 (as it can be seen in the above examples). This value signals that these words should not be filtered out by the filter based on semantic relevance.

The obtained results have to be filtered out in order to determine the best options for filling the gap. The threshold values of the three filters (frequency, POS score and semantic relevance) are computed as normalized sums of the scores obtained by each word. Their values are: 308 for frequency, 0.883849 for POS score and 4 for semantic relevance. From the previous 22 candidate words, only 6 words satisfied all the imposed restrictions: “approach”, “focus”, “interpretation”, “range”, “sense”, and “view”.

The process continues with each of these candidates until either no n-grams are found to continue on the current path or the maximum depth degree has been reached (the number of generated words is equal to the number of expected words to fill in the gap).

While the gap is filled in with candidates, every time a new candidate is added to the path, we check if the last word to be added is identical with the first one after the gap. In case of identical words, the path is saved as a possible fill for the gap.

4.1 Results

In our case, the first possible candidate would be: “An even more narrow *approach is a* text”. Another 194 possible candidates are found. These candidates are ordered based on their scores and then the candidates with the best scores are presented as the application results.

The best 10 results for our example, along with their scores, are presented below:

interpretation to make - Weight: 5.247865
interpretation of history - Weight: 4.659081
interpretation of information - Weight: 4.659081
interpretation of output - Weight: 4.659081
interpretation of source - Weight: 4.659081
interpretation of science - Weight: 4.659081
interpretation of article - Weight: 4.659081
interpretation of news - Weight: 4.659081
interpretation of body - Weight: 4.659080
interpretation of course - Weight: 4.659026

Since the correct solution for filling the gap (“interpretation is that”) has not been found, we will analyse what happened to it. The partial solution has been considered until the discovery process reached the third word (“interpretation is ?”). In order to replace the ? by a word, the word “that” had the following parameters:

[1] that [63850] [13 | IN/that : 0.450276] [11]

The thresholds imposed for this level were: 394 for frequency, 0.574628 for POS score and 2 for semantic relevance. As it can be seen, the test that caused this solution to fail is the POS score. The absence of the word “is” from the best 10 results shows that this word doesn’t have very good scores among the candidates. A readjust of the computed thresholds could allow the partial solution to pass the tests and to get into the final set of possible solutions, but that would not necessarily guarantee that it would have a score that allows it to get in the top 10 best results.

Although the exact solution has not been found, one can see that all of the top 10 candidates contained the content word from the gap – interpretation.

4.2 Other Results

In the following subsection, we shall present the results that have been achieved for three additional tests:

1) “An even more narrow <gap> is that text have a physical form, such as on paper or some other portable form, to the exclusion of inscriptions or digital media.”

Missing word(s): interpretation.
Results: approach [399][NN], view [754][NN], focus [2276][NN], interpretation [583][NN] and sense [1346][NN].

2) “for scientific instruction, yet <gap> remain too technical to sit well in most programmes”

Missing word(s): they.
Results: still [210782][RB] and they [418129][PP].

3) “and often have a primarily utilitarian purpose: <gap> data or convey immediate information.”

Missing word(s): to record.
Results: over 50 results, the closest results being: to [62786][TO] - present [6934][JJ],
to [62786][TO] - share [5828][NN],
to [62786][TO] - gain [7704][NN],
to [62786][TO] - study [5423][NN],
to [62786][TO] - test [3854][NN],
to [62786][TO] - order [4641][NN],
to [62786][TO] - move [8527][NN],
to [62786][TO] - process [3899][NN],
to [62786][TO] - control [4081][NN] and
to [62786][TO] - access [3631][NN].

5 CONCLUSIONS

In this paper we presented a generative method for reconstruction of partially damaged documents based on the text that remained intact. The method also uses the 5-grams Google corpus and the WordNet lexical database.

At the beginning of this project, we were very confident in the 5-gram Google corpus, thinking that the extent of the n-grams from this corpus will be adequate to cover all the n-grams from the analyzed documents and that we would never lower the n-grams order below 4. The experiments that we have made relative to the degree of n-grams from the documents that were also found in the corpus proved the contrary. The results showed that not all the n-grams from the documents are covered by the corpus n-grams and that the covering decrease varies from 90% in the case of bigrams to 15% in the case of 5-grams. The problem is that considering only bigrams could lead to a very large number of candidates that are not related to the document. This is why a trade-off has to be made between the covering percent of the n-grams and their order. Therefore, we considered that the best order of the n-grams is 3 (where the coverage is around 60%), with the option to decrease the order to bigrams whenever needed.

A different approach to overcome this problem is to use the Google search engine or the Google Search API instead of the Google n-grams corpus, and to analyze the results returned by the searches on the Web. The main problem with this approach is that the application issues many queries to the search engine, therefore the engine might restrict or even block the access to its data at least for a period. Another problem that has been identified is the situation where the gap contains proper names or numbers. It is very improbable that the same numbers or proper nouns could be identified in other documents. In the case of proper nouns the application could still be adapted, by replacing the nouns with pronouns that could be linked to the proper nouns found in the documents.

We consider that this method is worth further investigation, and if the results are good, the same method could be adapted to any field that supposes communications that could be faulty – starting from intermittent radio transmissions, continuing with damaged dialogue transcripts, and ending with archaeology. The only condition is to be able to model the field in a way similar to the modelling of the English language using n-grams.

ACKNOWLEDGEMENTS

We would like to thank the Linguistic Data Consortium for providing us the Web 1T 5-gram Version 1 Corpus for this research free of charge. The research presented in this paper was partially performed under the FP7 EU STREP project LTfLL.

REFERENCES

Baird, H. S., 2003. Digital libraries and document image analysis. In *International Conference on Document Analysis and Recognition,* pages 2-14.

Brants, T., Franz, A., 2006. Web 1T 5-gram Version 1, *Linguistic Data Consortium*, Philadelphia.

Breithaupt, M., 2001. Improving OCR and ICR accuracy through expert voting. Technical report, *Oce Document Technologies*. (www.csisoft.com/applications/OCE%20Intellidact%20Whitepaper.pdf)

Hong, T., Hull, J. J., 1995. Algorithms for Postprocessing OCR Results with Visual Inter-Word Constraints. In *Procs. International Conference on Image Processing*, Volume 3, Issue, pages 312 - 315.

Khoubyari, S., Hull, J. J., 1995. Font and Function Word Identification in Document Recognition. In *Computer Vision, Graphics, and Image Processing: Image Understanding*.

Kukich, K., 1992. Techniques for Automatically Correcting Words in Text. In *ACM Computing Surveys*, Vol. 24, No. 4, pages 377-439.

Meknavin, S., Kijsirikul, B., Chotimonkol, A. Nuttee, C., 1998. Combining Trigram and Winnow in Thai OCR Error Correction. In *Proceedings of COLING*, pages 836-842.

Nagy, G., Nartker, T. A., Rice, S. V., 1999. Optical character recognition: An illustrated guide to the frontier. In *Procs. Document Recognition and Retrieval VII, SPIE*, Volume 3967, pages 58–69, Kluwer Academic Publishers.

Tong, X., Evans, D., 1996. A Statistical Approach to Automatic OCR Error Correction in Context. In *WVLC-96*, pages 88-100.