Open-source OCR engine integration with Greek dictionary

Alkiviadis, Tsimpiris*

Department of Computer, Informatics and Telecommunications Engineering, International Hellenic University, Serres, Greece, atsimpiris@ihu.gr

Dimitrios Varsamis

Department of Computer, Informatics and Telecommunications Engineering, International Hellenic University, Serres, Greece, dvarsam@ihu.gr

Charalampos Strouthopoulos

Department of Computer, Informatics and Telecommunications Engineering, International Hellenic University, Serres, Greece, strch@ ihu.gr

George Pavlidis

ATHENA - Research and Innovation Centre in Information, Communication and Knowledge Technologies, University Campus at Kimmeria, Xanthi, Greece, gpavlid@ gmail.com

Kiourt Chairi

ATHENA - Research and Innovation Centre in Information, Communication and Knowledge Technologies, University Campus at Kimmeria, Xanthi, Greece, chairiQ@athenarc.gr

The aim of this study is the evaluation of an open-source OCR engine (tesseract OCR ver.4.0) by integration of a Greek dictionary with more than 500,000 words. To achieve this goal, an open access dictionary was initially used which was enriched with words that exist in the Greek restaurant menus. The training applied in the embedded LSTM deep learning model of Tesseract, before the integration of the new Greek dictionary. The evaluation of OCR performance applied with combinations of dictionaries in a total of 98 images from Greek catering menus. A slight but stable improvement of OCR performance after training and integration of the new Greek dictionary is observed at the results.

CCS CONCEPTS • Applied computing • Computers in other domains • Document management and text processing

Additional Keywords and Phrases: OCR, Tesseract, Greek, dictionary

ACM Reference Format:

Alkiviadis Tsimpiris, Dimitrios Varsamis, Charalampos Strouthopoulos, and George Pavlidis. 2021 Open-source OCR engine integration with Greek dictionary:

* Corresponding author.

1 INTRODUCTION

Document digitization is a common practice implemented by many companies and organizations and involves a small or huge volume of documents. The OCR engines usually perform very well as they are configured to recognize text written in the most common fonts in the most popular languages. OCR systems have never achieved a 100% read rate and therefore efforts are being made to improve this performance. The success of any OCR machine to read accurately is not the sole responsibility of the programmer. Much depends on the quality of the data to be processed. There are many open source applications available for optical character recognition, running either as API or under a graphical user interface.

In a recent work of [Lin,2019] classification of conventional OCR methods is illustrated in detail and point out their advantages as well as disadvantages. At the same work the corresponding key issues and techniques, including loss function, multi-orientation, language model and sequence labeling are also presented with description of commonly used benchmark datasets and evaluation protocols. Extracting training text data of sufficient quality and quantity is cumbersome. A ground-truthing tool named Aletheia was developed by [Clausner, 2014] to efficiently create training data for open-source text recognition engines. In [Heliński, 2012] work it is reported how recognition rates for non-mainstream documents can be significantly improved by training the OCR engine used. The report states improvements from 45 to 80% character accuracy rate and 15–55% word accuracy rate after training ABBYY FineReader on Gothic documents. A thorough work by [Holley, 2009] identifies and tests solutions to improve OCR accuracy in large-scale newspapers in the Australian National Library Newspaper Digitization Program.

There are many approaches for OCR improvement of text detection rate and optical character recognition accuracy [Harraj, 2015] but the majority of them are using datasets of English documents. To the best of our knowledge there are also few works of Greek text recognition. GRPOLY-DB [Gatos, 2015] is the first publicly available old Greek polytonic database, for the evaluation of several document image processing tasks. It contains both machine-printed and handwritten documents as well as annotation with ground-truth information. A complete database system for storing images of Greek unconstrained handwritten characters is GCDB [Margaronis, 2009]. An ontology of Greek national and local foods and wines supported by a standards compatible multilingual thesaurus was develop by [Markantonatou, 2018] where problems that encountered in the particular terminological domain drive to outline a methodology of populating the thesaurus. The Greek restaurant menus usually do not provide additional information about the food, the ingredients, the way it is prepared or a picture of the food itself and it is difficult to read the name of a Greek food. Due to the above reasons, there is a need for text recognition from restaurant menus and development of corresponding applications. In a relevant work of [Pavlides, 2020], new methods and AI tools for dish recognition and menu translation have been designed with promising results, covering the basic needs of tourists during a visit that involves culinary experiences. The dataset of Greek catering menu images with the corresponding ground text that used at the present study is a part of the dataset that used at the above work. The present study deals with text detection in scanned Greek catering menus using an OCR engine improved with a Greek dictionary additionally to the embedded Greek dictionary. The dataset of Greek catering menus that used, involves combinations of graphics, photographs, printed and handwritten texts, varying fonts and complex backgrounds. Text recognition from Greek restaurant's menus is an area that needs further research and application development.

2 METHODOLOGY AND MATERIALS

The methodology adopted in this work, shown in Figure 1, is independent of the selected dictionary language to be added to the Tesseract OCR machine and is described in the following steps:

1. Select a dictionary of the preferred language to be added in the OCR, with all the words as a sorted list in Unicode format.
2. Compare the words contained in the new dictionary with the existing words in the OCR's built-in dictionary of the selected language.
3. Created indexes for quick word search, according to the Tesseract OCR format.
4. Train the built-in LSTM model, used by Tesseract for recognition of the new dictionary words.
5. Insert the trained files of the new dictionary into the OCR engine.
6. Evaluate the performance of the OCR engine on a set of image dataset with the corresponding ground texts.

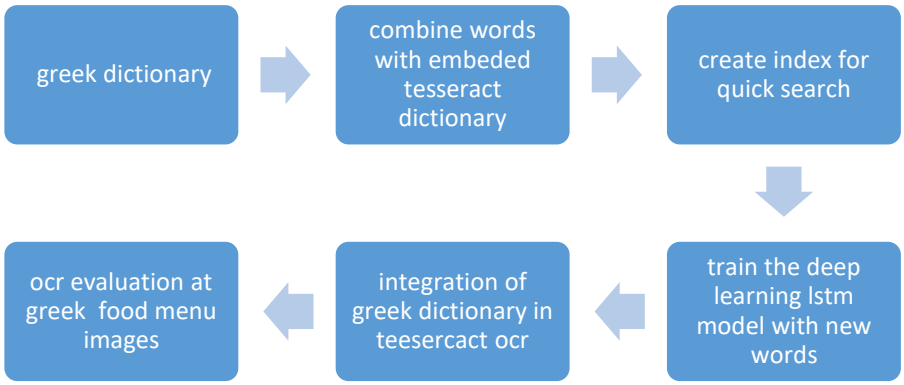

Figure 1. Methodology of OCR dictionary integration

2.1 Tesseract OCR

Tesseract is an open-source Optical Character Recognition Engine or API, available under the Apache 2.0 license appropriate for extraction text from images of different formats [Smith, 2007]. It is available for Linux or Windows operating systems and is supported by many programming languages. In this study Windows 10 and Python were used for Tesseract, on an i7 Intel processor with 16GB Ram. Tesseract OCR by default use Otsu's binarization as an initial step for image preprocessing, supports optical character recognition for Greek and other languages and also use a line finding algorithm so that a skewed page can be recognized without having to de-skew, thus saving loss of image quality. Moreover, Tesseract handle pages with curved baselines, measuring gaps in a limited vertical range between the baseline and mean line. Tesseract 4.0 added a new OCR engine based on LSTM neural network. Although considered the best open-source OCR engine, a training with new words, is able to increase the accuracy of this engine.

2.2 Image Preprocessing

The OCR systems are using by default, binarization processes to convert colored images to black and white images. Noisy background, blurred images or images captured by mobile devices need threshold methods like binarization to give assistance at OCR systems [Goupta, 2007].

2.2.1 *Binarization Otsu*

The Otsu binarization method is by default used at Tesseract to perform automatic image thresholding. In the simplest form, the algorithm returns a single intensity threshold that separate pixels into two classes, foreground and background calculating the threshold value from images histogram [Otsu,1979].

2.3 Dictionaries

Tesseract OCR uses a special effective format for dictionaries, called Directed Acyclic Word Graph (DAWG) [Daciuk, 2000]. A DAWG file can be created of a plain text word list using the *wordlist2dawg* command line tool. The reverse is also possible (a list of words can be extracted from a DAWG file), which can be useful for update of existing dictionaries.

There are five different types of dictionaries that can be created and added separately for each language to help Tesseract decide on different possible character combinations. For example:

1. Word list: All possible words that are expected to be met (lowercase - uppercase).
2. Common words: Words that appear frequently in the given language (e.g "the" for English).
3. Number Patterns: Special patterns that represent numbers with mathematical symbols, units, etc.
4. Punctuation patterns: Special patterns that represent punctuation marks for sentences and words.
5. Bigrams of words: Common combinations of word pairs.

At this study, the 'Word List' type was used for the new Greek dictionary. The built-in Tesseract English dictionary is denoted as *eng*, the Greek built-in dictionary is denoted as *ell* and the new dictionary that added into the Tesseract engine is denoted as *greek* and consists of 500,000 different Greek words. This dictionary is open source and is mainly used as a dictionary for Latex word processing to check and correct spelling mistakes in Greek texts. Figure 2 shows how the DAWG files are constructed for high-speed search at the dictionaries that used by Tesseract OCR engine. The DAWG example refers to the following words: cat, cats, fact, fact, facet, facets

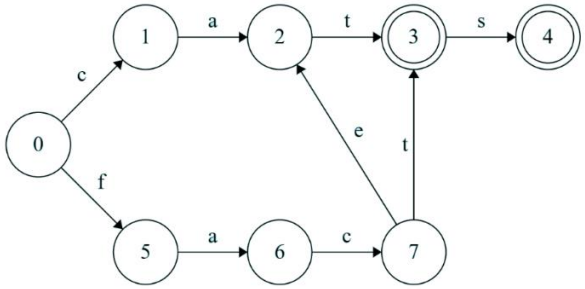

Figure 2. DAWG: Nodes are considered identical if they have the same terminal state, same outgoing edges, and for each of those edges, the children of both nodes are identical. Example for the following words :cat, cats, fact, facts, facet, facets. (https://jbp.dev/blog/dawg-basics.html, 5-8-2021)

2.4 Combine and indexing words

The command *combine_tessdata* is the main program to combine/extract/overwrite *tessdata (e.g greek.dic)* components in *[lang].traineddata* files that refer to the new dictionary files. The syntax of the command to combine all

the individual *tessdata* components (unicharset, DAWGs, classifier templates, ambiguities, language configs) located at a path (e.g. home/$USER/temp/eng.*) is: `combine_tessdata /home/$USER/temp/eng`.

The result will be a combined *tessdata* file at /home/$USER/temp/eng.traineddata. As a result, the corresponding tessdata/eng.traineddata file will contain the new language config and unichar ambigs, plus all the original DAWGs, classifier teamples, etc.

The follow are indicative components in a generated Tesseract lang.traineddata file:

- lang.config: provides control parameters which can affect layout analysis, and sub-languages.
- lang.unicharset: is the list of symbols that Tesseract recognizes, with properties.
- lang.punc-dawg: a dawg made from punctuation patterns found around words.
- lang.word-dawg: a dawg made from dictionary words from the language.
- lang.freq-dawg: a dawg made from the most frequent words which have gone into word-dawg.
- lang.lstm: Neural net trained recognition model generated by *lstmtraining* command.
- lang.lstm-punc-dawg: A dawg made from punctuation patterns found around words.
- lang.lstm-word-dawg: A dawg made from dictionary words from the language.
- lang.lstm-unicharset: The unicode character set that Tesseract recognizes, with properties. Same unicharset must be used to train the LSTM and build the lstm-*-dawgs files.

(a) (b)

Figure 3: Two scanned Greek catering menus with (a) simple fonts and background (b) complex fonts and background .

2.5 Scanned images dataset of catering menus

The dataset that used at this study to evaluate OCR's efficiency, consists of 98 images of different Greek catering menus with different background, fonts and brightness. Text recognition becomes a difficult and complicate procedure under these facts. All menu pages of this dataset were scanned in high resolution (600dpi) with fixed scan angle, giving high quality images for text recognition in order to achieve maximum OCR performance. This dataset of high resolution scanned Greek catering menus and the corresponded ground texts is a unique dataset for OCR evaluation in Greek language texts and is a part of [Pavlides, 2020] dataset. Two representative Greek catering menus samples are presented in Figure

3 with (a) plain font and soft background color and (b) complex font and background, in order to highlight the difficulty in recognizing the Greek characters and words from these images.

2.6 OCR evaluation

The comparison between ground text A related to a food menu image and the OCR output text B is the procedure for OCR's efficiency evaluation. There are many text similarity indices at character level or word level. At this work the Word Error Rate (WER), the Character Error Rate (CER), the cosine, the Jaccard and the ratio indicators were used. The WER is based on the Levehnstein distance [Levenshtein, 1966] between B and A according the following equation:

$$WER = \frac{iw + sw + dw}{nw} \quad (1)$$

where *iw*, *sw* and *dw* refer to the number of words that need to be inserted, replaced, and deleted to transform the exported text B into the reference text A. The total count of these words is normalized by the number of words in the reference text A (*nw*). In this study the number of new words in the extracted text was ignored (*iw* = 0), transforming this index to the classical error rate index.

$$WERi = \frac{sw + dw}{nw} \quad (2)$$

The according accuracy index is formed as follows:

$$WAcci = 1 - WERi \quad (3)$$

In this work the *WAcci* index is used (ranged between 0 and 1), without taking into account the new words of the extracted text B (*iw* = 0) and is independent of words position in the text. At character level, the character error rate (*CER*) is formed as follows:

$$CER = \frac{ic + sc + dc}{nc} \quad (4)$$

where *ic*, *sc* and *dc* refer to the number of characters that need to be inserted, replaced, and deleted to transform the exported text into the ground text and nc refers to the total characters of ground text A. The same assumption of *WERi* index, in relation with the number of ignored inserted words, adopted at *CERi* index as well for the inserted characters that also ignored, setting the parameter *ic* = 0. The equivalent index of characters accuracy *CAcci* is also calculated.

$$CAcci = 1 - CERi \quad (5)$$

The word-level error rate is usually higher than the character-level error rate, as failures in recognizing individual characters significantly affect a word recognition. In many OCR application the correct words identification is much more important than correctly identifying single characters, numbers, or punctuation. It is defined that a word is any sequence of one or more letters. If *mw* of the *nw* words in the reference text A are correctly recognized, then the word accuracy is *WAcci* = *mw/nw*. It is considered that a word is not correctly recognized if one or more letters of the OCR extracted word are incorrect.

3 RESULTS

Figure5 shows the values of *1-CERi* and *1-WERi* indices for OCR performance over 98 catering menu images at the case where the *eng+ell* dictionaries used by the OCR. The *1-CERi* index range between 0 and 0.95 where the values close to zero indicating a bad OCR performance due to a very difficult background of catering menu images or due to unknown fonts. The *1-WERi* values range between 0 and 0.5 indicating that most of the words in the sentences of catering menus were wrongly transcribed. A menu image with the highest *1-CERi* value is presented at Figure 3 (a) and an image with the worst *1-CERi* value is also presented at Figure 3 (b).

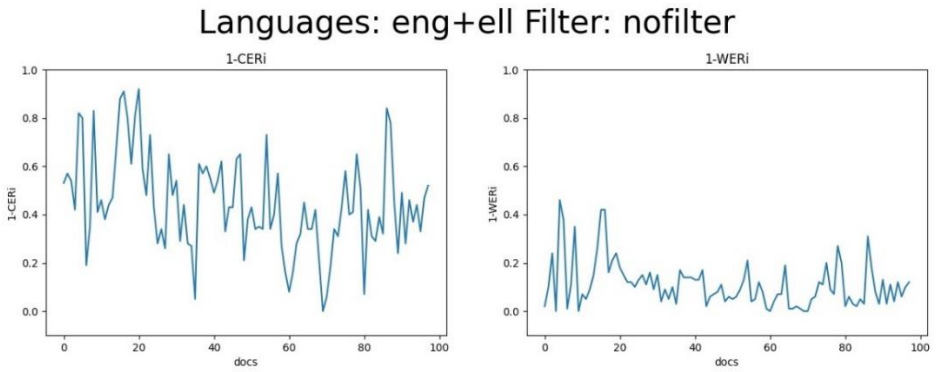

Figure 3. The evaluation of Tesseract OCR with 1-CERi and 1-WERi indices for the default dictionaries (eng+ell) of Tesseract without any preprocessing filter

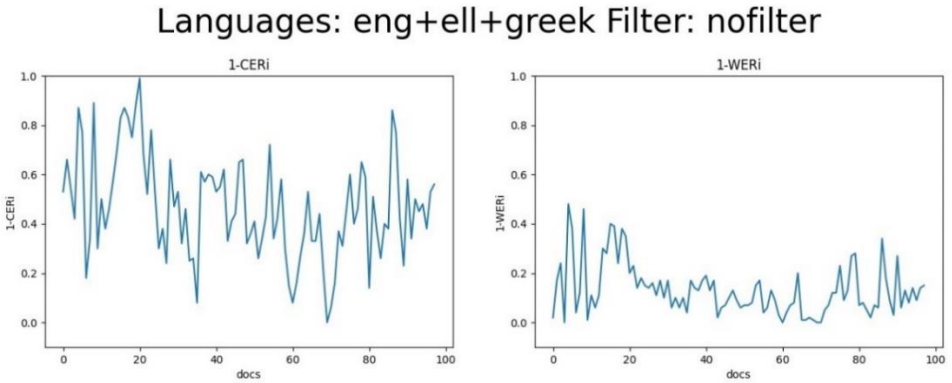

Figure 4. The evaluation of Tesseract OCR with 1-CERi and 1-WERi indices for the additional Greek dictionary (eng+ell+greek) without any preprocessing filter

OCR performance in the case of adding the Greek dictionary to the default OCR dictionaries is shown in Figure 6, where a slight improvement in OCR efficiency is observed in both character level recognition and word level recognition, as shown by the accuracy indicators *1-CERi* and *1-WERi*. The average values of *1-CERi* and *1-WERi* indices, over 98 images are presented at Table 1. It is observed that the mean of *1-CERi*=0.45 for eng+ell dictionaries where the respective value for eng+ell+greek dictionaries is equal to 0.47. The same behavior is observed also for *1-WERi* index. The improvement of OCR efficiency is 2% for both character and word level recognition, according the evaluation indices. The values of *1-WERi* index are lower than *1-CERi* index, but this is an expected result for the word level accuracy.

Table 1: Average results of OCR evaluation

dictionaries	*Average values*	
	1-CERi	*1-WERi*
eng+ell	0.45	0.11
eng+ell+greek	0.47	0.13

4 CONCLUSIONS

The investigation of the improvement of Tesseract OCR after the addition of a new Greek dictionary and OCR evaluation on Greek catering menu images is achieved in this work. The procedure steps for integration of a new Greek dictionary into Tesseract were described. The results showed that the addition of the new dictionary increased the accuracy of word and character recognition in the case of Greek catering menus by 2%. Future work of this study is the application of the same dataset and the comparison of the results to other popular OCR engines, like Azure, Google Vision, ABBYY Finder.

ACKNOWLEDGMENTS

This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE, project code:(T1EDK-02015).

REFERENCES

Smith, R., Google, I. (2007) An overview of the Tesseract OCR Engine. Proc. 9th IEEE Intl. Conf. on Document Analysis and Recognition (ICDAR), pp. 629–633.

Gupta, M.P., Jacobson, N.P. and Garcia, E.K. (2007) OCR binarization and image pre-processing for searching historical documents. Pattern Recognition, 40, 389–397.

Otsu, N. (1979) A threshold selection method from gray level histograms. IEEE transactions on systems, man, and cybernetics, 19, 62–66.

Clausner, C., Pletschacher, S., Antonacopoulos, A. (2014) Efficient OCR Training Data Generation with Aletheia, Short Paper Booklet of the 11th International Association for Pattern Recognition (IAPR) Workshop on Document Analysis Systems (DAS2014), Tours, France, pp. 19-20

Heliński, M., Kmieciak, M., Parkoła, T. (2012) Report on the comparison of Tesseract and ABBYY FineReader OCR engines. PCSS, oai:lib.psnc.pl:358

Gatos, B.; Stamatopoulos, N.; Louloudis, G.; Sfikas, G.; Retsinas, G.; Papavassiliou, V.; Sunistira, F.; Katsouros, V. (2015) GRPOLY-DB: An old Greek polytonic document image database. 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 646–650.

Margaronis, J., Christou, M., Kavallieratou, E. and Tzouramanis, T. (2009) GCDB: A Character Database System. Proceedings of the International Workshop on Multilingual OCR; ACM: New York, NY, USA, 2009; MOCR '09, 17:1–17:7

Holley, R. (2009). How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs. D-Lib Magazine: The Magazine of the Digital Library Forum. 15.

Harraj A. and Raissouni, N. (2015) OCR Accuracy Improvement On Document Images Through A Novel Pre-Processing Approach. Signal & Image Processing : An International Journal (SIPIJ) Vol.6, No.4

Levenshtein, V.I. (1966) Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Cybern Control Theory, 10, 707–710.

Markantonatou, S. and Pavlidis, G. (2018), Greek culinary tourism is lost in translation. 7th International Conference on Strategic Innovative Marketing and Tourism 2018

Pavlidis, G., Markantonatou, S., Toraki, K., Vacalopoulou, A., Strouthopoulos, C., Varsamis, D., Tsimpiris, A. Mouroutsos, S., Kiourt, C., Sevetlidis, V. and Minos, P. (2020) AI in gastronomic tourism. In Proceedings of the 2nd International Conference on Advances in Signal Processing and Artificial Intelligence International Frequency Sensor Association (IFSA) Publishing, S. L., Berlin, Germany