Translation-Inspired OCR

Dmitriy Genzel, Ashok C. Popat, Nemanja Spasojevic, Michael Jahr, Andrew Senior, Eugene Ie, Frank Yung-Fong Tang
Google, Inc.
Mountain View, CA, USA
Email: {dmitriy,popat,sofra,mjahr,andrewsenior,eugeneie,ftang}@google.com

***Abstract*—Optical character recognition is carried out using techniques borrowed from statistical machine translation. In particular, the use of multiple simple feature functions in linear combination, along with minimum-error-rate training, integrated decoding, and N-gram language modeling is found to be remarkably effective, across several scripts and languages. Results are presented using both synthetic and real data in five languages.**

***Keywords*-Optical character recognition; statistical machine translation; minimum-error-rate training.**

I. Introduction

Optical Character Recognition (OCR) is an old field; approaches have evolved over time as needs and technologies have changed [1], [2]. This evolution continues to this present day — advances in related fields such as Speech Recognition influenced approaches developed in the 90s [3], [4]; advances in Machine Learning are often tested on various "toy versions" of the OCR problem, such as isolated glyph recognition, to evaluate and compare learning algorithms.

Here, we consider an approach to OCR that exploits recent advances in the related field of Statistical Machine Translation (SMT). We wish to assess the potential of our approach in a fairly realistic setting, so we work on entire text lines and consider a variety of languages, including ones having connected scripts and large character sets. We do not consider layout analysis however, and rely instead on the text lines having been previously extracted from the page images.

In SMT systems the decoder covers the source sentence (the line image in our case) with phrases which have corresponding translations (in our case characters, or more precisely *grapheme clusters*). By considering sequentially computed image features as source words and characters as target words, OCR becomes a special case of machine translation. Of course, if treated naively, this approach would lead to poor quality because of inherent variability in the underlying data and processes, making it unlikely that image segments will match previous observations pixel-for-pixel. We are able to address this issue within a typical SMT framework, implementing an OCR system using an existing SMT system as a back-end with only minor changes.

II. System description

Let I be a line image of width w. We treat I as a sequence of single-pixel columns $I = (I_k)_1^w$. Let $S = (S_i)^n$ be a horizontal partition of this image, where each cell S_i is a group of consecutive pixel columns. Each S_i is intended to be an area of an image where a single character is printed. Due to overlap of character bounding boxes, this ideal of horizontal partitioning into single-character cells can only be approximated. We refer to the character in S_i as t_i. The output of the system is the *transcription* $T = (t_i)_1^n$. Let the set of possible t_i be denoted as $\mathcal{T}$.

We use a maximum-entropy inspired linear model, similar to that of Och and Ney [5].

$$\begin{aligned}\hat{T} &= \arg\max_T P(T|I)\\ &= \arg\max_T \frac{\exp\left[\sum_{i=1}^m \lambda_i h_i(T,I)\right]}{\sum_T \exp\left[\sum_{i=1}^m \lambda_i h_i(T,I)\right]}\\ &= \arg\max_T \sum_{i=1}^m \lambda_i h_i(T,I)\end{aligned}$$

where $h_i(T,I)$ are feature functions (typically log-probabilities of various kinds) and λ_i are fixed weights. One critical kind of feature function is a language model:

$$h_{\text{LM}}(T,I) = \log P(T) = \log \prod_{i=1}^n P(t_i|t_0 \dots t_{i-1})$$

This feature is by definition context dependent: it cannot be evaluated for t_i independently of the preceding hypothesized transcription elements. Many other features can be so decomposed, if the segmentation is provided. Therefore, in practice, instead of maximizing over all transcriptions T we maximize over pairs of (T,S):

$$\hat{T} = \arg\max_{T,S} \sum_{i=1}^m \lambda_i h_i(T,S,I)$$

And for decomposable (local) features, we have:

$$h_k(T,S,I) = \sum_{i=1}^n h'_k(t_i,S_i)$$

An example of a local feature is what is known in SMT as CEF:

$$h'_{\text{CEF}}(t_i,S_i) = \log P(t_i|S_i) = \log \frac{\text{Count}(t_i,S_i)}{\text{Count}(S_i)}$$

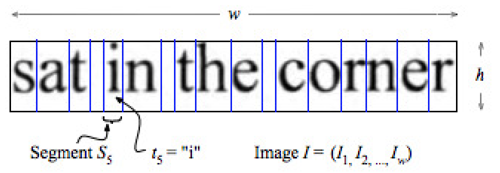

Figure 1. Example line image to explain notation. Here, the transcription $T = (t_i)_1^{17}$ is *sat in the corner*.

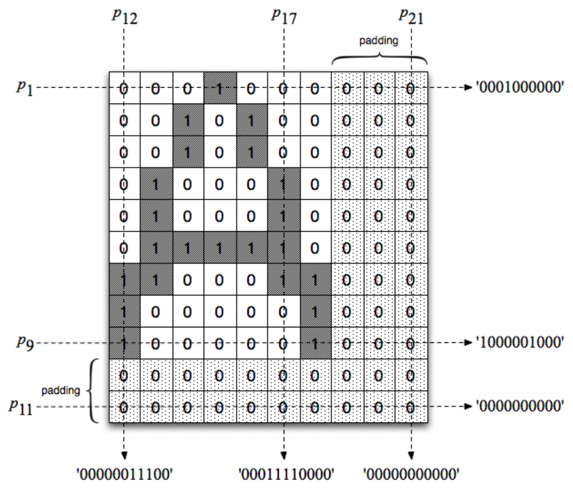

Figure 2. Simple row and column projections of the binarized segment form source-side pseudo-words.

where Counts are computed over a training corpus, and therefore

$$h_{\text{CEF}}(T, S, I) = \sum_{i=1}^{n} \log \frac{\text{Count}(t_i, S_i)}{\text{Count}(S_i)}$$

In SMT the CEF feature is very useful, but for OCR it is not, due to tremendous sparsity of the input domain. A single pixel having a different value would cause the count to become zero. We need feature functions that are more robust to input noise.

A simple way to deal with this problem is to project onto multiple smaller spaces. Let w_S and h_S be the respective maxima of the widths and heights over all S_i in all training line images.[1] We define a set of $s = w_S + h_S$ *projections* $\{p_1, \ldots, p_s\}$, each mapping S_i to a binary string. Specifically, each $p_k(S_i), k = 1, \ldots, h_S$ is a row projection obtained by reading off the binarized pixels in the k^{th} row of S_i as a string of zeros and ones, padding with trailing zeros (white background) as necessary. The column projections $p_k(S_i), k = h_S + 1, \ldots, s$ are defined analogously. See Figure 2.

[1]As a practical matter, very large cells deemed to be outliers are excluded from the computation of w_S and h_S.

Having defined these projections, we can compute the following features based on each:

$$h_{k,\text{CEF}}(T, S, I) = \sum_{i=1}^{n} \log \frac{\text{Count}(t_i, p_k(S_i))}{\text{Count}(p_k(S_i))}$$

and

$$h_{k,\text{CFE}}(T, S, I) = \sum_{i=1}^{n} \log \frac{\text{Count}(t_i, p_k(S_i))}{\text{Count}(t_i)}$$

These local features, the language model feature, and the length feature ($h_{\text{length}}(S, T, I) = |T| = n$) are all the features we need.

A. Training

The training data consists of text line images along with annotations indicating the line transcriptions and bounding boxes for each character. We assume that extraneous white space on the left and right of the line image has been removed. The provided bounding boxes are not necessarily a partition of the image: they may intersect, leave empty space between them, and they do not cover the entire pixel columns vertically. Our first task is to transform them to a partition. To do this we employ a heuristic. First, we extend all bounding boxes vertically to cover the entire image. Where the boxes intersect, we let the smaller of the boxes keep the area, and adjust the other box. Where there is empty space, we also expand the smaller of its neighbors to cover it. We also reduce image resolution to 100dpi to make the decoder search faster.

We iterate over each bounding box, padding the image contained in the box with white background to the fixed width w_S and height h_S as described above, and extracting all column and row projections from this image. We then compute co-occurrence counts of the form $C(k, p_k(S_i), t_i)$ over the entire training corpus. We then output a mapping where the key is $k, p_k(S_i)$ pair, and the value is a set of triples:

$$\left\{ t_i, \log \frac{C(k, p_k(S_i), t_i)}{C(k, p_k(S_i))}, \log \frac{C(k, p_k(S_i), t_i)}{C(t_i)} \middle| t_i \in \mathcal{T} \right\}$$

i.e. a set of possible characters it can correspond to, with a CEF and CFE score for each. We refer to this data as a projection table which serves as a functional parallel to a phrase table in SMT.

In addition to training the projection feature functions, we also train a 10-gram character language model, in the same way as similar model is trained for SMT, except that we need to turn spaces into pseudo-words.

B. Decoding

We use a state of the art phrase-based statistical SMT system (similar to that of Och et al. [6]) as a back-end. For this, we turn an image into a collection of pseudo-words, each word encoding an image column. For the target side, we

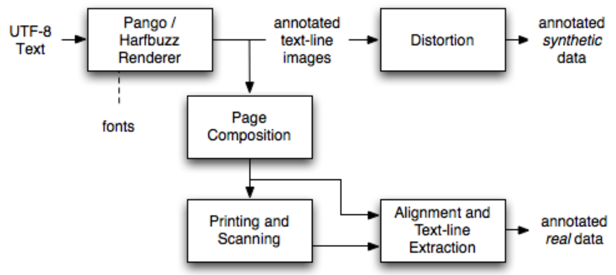
UTF-8 Text
Pango / Harfbuzz Renderer
annotated text-line images
Distortion
annotated *synthetic* data
fonts
Page Composition
Printing and Scanning
Alignment and Text-line Extraction
annotated *real* data

Figure 3. Data-generation pipeline.

Table I
SOURCE TEXT STATISTICS.

Language	Font	Codepoints	Characters	Lines
Arabic	Times New Roman	1,176,836	1,172,236	10,906
Chinese	AR PL New Sung	1,472,932	1,472,932	40,000
English	Times New Roman	3,306,423	3,306,423	40,000
Hindi	Lohit Hindi	893,466	627,665	11,876
Russian	Times New Roman	2,890,716	2,890,641	40,000

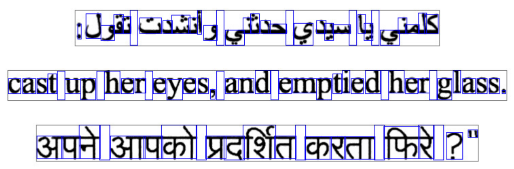
كلمني يا سيدي حدثني وانشدت تقول:
cast up her eyes, and emptied her glass.
अपने आपको प्रदर्शित करता फिरे ?"

Figure 4. Examples of ground-truth annotation of grapheme cluster (character) locations in Arabic, English, and Hindi PRAN-data examples.

treat each character as a word, except that we use a special token for spaces. We also modify the portion of the SMT system that is responsible for the retrieval of phrases. Instead of retrieving a set of target phrases for each source phrase, as is done in SMT, we consider each source span "phrase" of up to maximum width (as observed in training). This is then converted to an image, all the projections are computed and the corresponding sets of characters with their CEF and CFE scores are retrieved. For each character, we then associate all retrieved scores (twice the number of projections) as $2n$ different feature functions associated with that "phrase". We also add a feature function *NWORDS* with a value of 1 to each phrase. Some characters will have infinite costs for some feature functions, if they were not observed to co-occur with the corresponding projection. We smooth those, by assigning a very low fixed probability (high cost) instead. We also prune those phrases that did not co-occur with most projections.

The normal SMT search (with reordering disallowed) is then performed, using these local features as well as a language model, described above. This is essentially equivalent to a beam search in a lattice which we have effectively constructed in the previous step. Other lattice scoring algorithms (with language model support) could be used as well.

To optimize the feature weights $\{\lambda_i\}$, Minimum Error Rate Training [7] is run on a development set.

For Arabic, we have a choice of either treating it as a left-to-right language (requires inverting the transcription for the LM), or as right-to-left (simply flip the image, keeping transcription constant). We did both, and the results are substantially the same, and we report them only for left-to-right decoding. Processing Arabic left-to-right has the advantage of uniformity and means that in a multilingual system we do not have to detect the direction of the text before we start decoding.

III. DATA

As mentioned, our processing is currently oriented around isolated text lines, as we are not attempting to address layout analysis or page segmentation in this work. Our elementary unit of annotated or *ground-truth* data comprises a line image's raster data, its UTF-8 transcription, a collection of (possibly overlapping) bounding boxes indicating the locations of individual characters, and an indication of which codepoints in the transcription compose each character. The latter is potentially useful in training, debugging, and in evaluating the output segmentation returned by the OCR system. An example is shown in Figure 4.

Data sets were produced by obtaining UTF-8 text in the desired languages, then typesetting the results to produce sets of annotated ground-truth text-line data units. We converted these into two datasets for each language: *synthetic,* wherein the data is kept in digital form, corrupted only by pseudorandom noise, and *PRAN,* wherein the lines are composed on a page, printed, scanned, and aligned to produce annotated datasets (details below). In all cases, random splitting was used to partition the data into disjoint training, development, and test subsets.

A. Source texts

A number of works in each of the five languages were selected and downloaded from *wikisource.org*. One advantage of this particular source is that its data is in the public domain, and can therefore be distributed along with its derivative images for use by other researchers, as we plan to do. Another is that some of the data has been proofread and verified by volunteers literate in the relevant languages; all of the selected works met this standard.

The English and Russian are entirely from Dickens and Tolstoy, respectively. The Hindi is modern from novels; the Chinese and Arabic texts are classical (the former in Traditional Han script).

B. Print-Scan Pipeline

The PRint-scAN (*PRAN*) data pipeline is shown in Figure 3. The source text is fed to a renderer based capable of high-quality typesetting in a vast array of languages, correctly producing requisite complex ligatures and consonant clusters, and capable of reporting the extents of the grapheme clusters. Fonts were chosen such that primary glyph support was present for each character without resorting to fallback. In all cases the font size was 12 point, typeset at 300 dpi in grayscale, with pixel values ranging from zero to 255.

To prepare the *synthetic* data-sets, we simply added i.i.d. Gaussian noise ($\mu = 0$, $\sigma = 20$) to each line image. An example is shown in Figure 5.

Preparation of the *PRAN* datasets was more involved. We began with the same elementary text-line units used in *synthetic*, but rather than add noise, we composed them into a set of PNG letter-size page images, maintaining the 300 dpi resolution and 8-bit grayscale depth. For each page, we recorded the exact position (bounding box) of each line as side information. The page images were binarized to prevent stubble associated with halftone-screens, then printed on a Ricoh MP C5000 laser printer. The number of lines was limited to 40,000 in each language (see Table I), 40 lines per page. For Arabic and Hindi the available source text was insufficient to reach that limit.

The resulting physical pages were then scanned on a sheet-fed scanner Panasonic KV-S3105C in 8-bit grayscale, again at 300 dpi. A post-processing step was then applied to clean and de-skew the images and reduce the pixel-depth down to 16 gray levels. Despite the post-processing, the resulting scanned images contain various distortions, the most severe of which is uneven stretching and shrinking in the vertical dimension due to fluctuation in the rate at which the sheets move through the rollers.

In order to recover the line bounding-box locations (and therefore the precise character locations which are known relative to the line boxes), it is necessary to map the geometry of the original synthetic page image to that resulting from scanning. To accomplish this, we first scale the scanned image to be precisely the same size as the original. Next, we identify likely word-bounding boxes using simple image morphology operations. An initial matching is then obtained between the word boxes in each image, taking into account both centroid location and aspect ratio as matching features. [8]. Boxes that match with sufficient confidence are then used in a second-pass alignment. Specifically, the pixel locations in the matching boxes are placed in a correspondence $(x, y) \rightarrow (x', y')$, allowing least-square estimation of the affine transformation matrix in

$$\begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

This transformation matrix can model scaling, translation, rotation, and shear, making it amply suited to model the geometric page mapping in the present case of sheet-fed scanning.

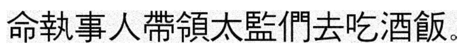
命執事人帶領太監們去吃酒飯。

Figure 5. Slightly noisy synthetic data example.

Table II
RESULTS ON SYNTHETIC DATA (CER)

Language	Commercial OCR	Our system
English	0.44%	0.10%
Russian	2.46%	0.50%
Arabic	N/A	0.34%
Chinese	3.69%	0.53%
Hindi	N/A	0.70%

After estimating the transformation matrix, the word-bounding boxes are discarded. The transformation is applied to the original line bounding boxes to obtain the corresponding boxes in the scanned image, and similarly for the character boxes. The character boxes shown in Figure 4 were recovered in this manner.

IV. EXPERIMENTS

A. Evaluation metric

We use a standard OCR evaluation metric, character error rate (CER) which is defined as the sum of edit distances between the proposed and the correct transcriptions over all the test instances, divided by the total length of the correct transcriptions.

B. Synthetic data

For our first set of experiments we use our *synthetic* datasets, randomly divided into training, development and test in proportion of 92%, 4%, 4%, for each language. We use bounding box information for training data, but for development and test sets we use only raw images as input.

The results are shown in Table II. We compare performance to a state-of-the-art, well-known commercial system as a baseline. (Our license may not allow us to state which commercial system we are using.) This system does not have support for Arabic and Hindi. As an additional comparison point, we ran English-only experiments using a speech-based system similar to that described in [4]; the resulting error rate was 0.49%.

C. PRAN data

For the second set of experiments, we use *PRAN*, prepared as described above, and subdivided into training, development and test in the same proportion as for synthetic data. Results are presented in Table III; the error rate for the speech-based system for this data was 0.74% (English).

In addition, we performed an experiment where we combined all the data for different languages together, and split into training, development and test in the same proportion. The data was not annotated with respect to language. There

Table III
RESULTS ON REAL DATA (CER)

Language	Commercial OCR	Our system
English	0.81%	0.45%
Russian	1.16%	0.73%
Arabic	N/A	1.97%
Chinese	4.01%	3.00%
Hindi	N/A	2.20%

were no special features added. The system obtained a CER of **2.81%**.

V. DISCUSSION

The experimental results indicate that the OCR system based on machine translation can be effective on *PRAN* data across diverse scripts and languages, including connected (Arabic) and large-character-set (Chinese). Moreover, the system appears to adapt well to lower-resolution images (e.g., 100 dpi), and in principle can handle mixed languages by simply using merged phrase tables and language models (albeit at a computational cost).

The features used in these experiments are rather crude: they involve binarizing the image as a first step (a throw-back to traditional pipelined approaches to OCR), and they fundamentally give up the ability to generalize observations by appealing to "nearness" in projection space. With the framework in place and baselines established, it should be straightforward to incorporate more sophisticated features, including those known to perform well in other OCR approaches. We leave this for future work.

While the error rates appear to be competitive with the commercial system on both *PRAN* and synthetic data, some caution is required in interpreting the results. The language models were trained on the training part of the data-split; since the data comes from limited works it is highly in-domain, giving our method an advantage.

Key to the success of approaches to OCR like the one described here is the availability of suitable training data. The *PRAN* pipeline we described for producing precisely annotated data scales quite well in principle, and extends readily to diverse scripts and languages.

While real in the sense that actual physical pages were printed and scanned, the *PRAN* retains some artificial qualities: homogeneity in the script, language, typeface, and point size. An alternative approach would be to scan existing printed material, and fill in the ground truth annotations using a suitable manual data-entry process. The amount of skilled human effort required per page for that approach is substantial, and any errors made in the manual data entry will cause the transcription to fail to match the image. While *PRAN* also requires some human effort (e.g., to correct the source text), the impact of errors is less: the transcription will still be faithful to the image, even if it contains errors relative to the intended text. From this point of view the human role in the *PRAN* pipeline can be seen as somewhat optional, opening the door to using vast amounts of mined text for training, or even pseudorandom text generated by running Markov chains.

VI. CONCLUSION

We have presented an approach to OCR patterned after a modern statistical machine translation system, and noted reasonable performance on both *PRAN* and synthetic data across several scripts and languages. The resulting system can incorporate multiple and diverse feature functions, automatically weighing each appropriately for the given data and conditions. The system is easily trained from data for new scripts, languages, and input modalities.

REFERENCES

[1] H. F. Schantz, *History of OCR, Optical Character Recognition*. Recognition Technologies Users Association, 1982.

[2] G. Nagy, "At the frontiers of OCR," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1093–1100, Jul 1992.

[3] G. E. Kopec and P. A. Chou, "Document image decoding using Markov source models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 602–617, 1994.

[4] I. Bazzi, C. LaPre, J. Makhoul, C. Raphael, and R. M. Schwartz, "Omnifont and unlimited-vocabulary OCR for English and Arabic," in *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 842–846.

[5] F. J. Och and H. Ney, "The alignment template approach to statistical machine translation," *Computational Linguistics*, vol. 30, no. 4, pp. 417–449, 2004.

[6] F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev, "A smorgasbord of features for statistical machine translation," in *HLT-NAACL 2004: Main Proceedings*, Boston, Massachusetts, USA, May 2 - May 7 2004, pp. 161–168.

[7] W. Macherey, F. Och, I. Thayer, and J. Uszkoreit, "Lattice-based minimum error rate training for statistical machine translation," in *Proceedings of the EMNLP-2008*, Honolulu, Hawaii, October 2008, pp. 725–734. [Online]. Available: http://www.aclweb.org/anthology/D08-1076

[8] N. Spasojevic, G. Poncin, and D. Bloomberg, "Discrete point based signatures and applications to document matching," in *ICIAP 2011: Proceedings of the 16th international conference on image analysis and processing*, 2011.