HMM-based Script Identification for OCR

Dmitriy Genzel
Google, Inc.
Mountain View, CA, USA
dmitriy@google.com

Ashok C. Popat
Google, Inc.
Mountain View, CA, USA
popat@google.com

Remco Teunen
Google, Inc.
Mountain View, CA, USA
remco@google.com

Yasuhisa Fujii
Google, Inc.
Mountain View, CA, USA
yasuhisaf@google.com

ABSTRACT

While current OCR systems are able to recognize text in an increasing number of scripts and languages, typically they still need to be told in advance what those scripts and languages are. We propose an approach that repurposes the same HMM-based system used for OCR to the task of script/language ID, by replacing character labels with script class labels. We apply it in a multi-pass overall OCR process which achieves "universal" OCR over 54 tested languages in 18 distinct scripts, over a wide variety of typefaces in each. For comparison we also consider a brute-force approach, wherein a singe HMM-based OCR system is trained to recognize all considered scripts. Results are presented on a large and diverse evaluation set extracted from book images, both for script identification accuracy and for overall OCR accuracy. On this evaluation data, the script ID system provided a script ID error rate of 1.73% for 18 distinct scripts. The end-to-end OCR system with the script ID system achieved a character error rate of 4.05%, an increase of 0.77% over the case where the languages are known a priori.

1. INTRODUCTION

Over the past decade, several commercial and open-source optical character recognition (OCR) systems have expanded their language coverage dramatically — via optional modules [3], by providing a means for the user to train the system [1, 6, 9], or both [13]. Encouragingly, this expanded coverage extends in some cases to previously under-served languages, such as some written in the Arabic, Indic, and Southeast Asian scripts. While a welcome advance, support for these scripts and others is typically not automatic within a document: the recognition language, or the set of recognition languages, must be specified manually and in advance. In some cases an arbitrary mix of recognition languages can be specified [4], but to the best of our understanding doing so in current systems comes at the expense of both recognition accuracy and speed.

Ideally, OCR of a multiscript and multilingual document should "just work," without requiring advance specification of which scripts and languages are present, and without sacrificing much in the way of speed or quality. To build intuition that this should be possible, consider human performance in a multilingual setting.

A polyglot human reader is not unduly slowed down, or induced to make a greater number of errors in understanding, by the presence of passages in different languages that he or she individually understands. To emulate this capability by a computer, the added skill needed beyond the conventional OCR goal of monolingual text recognition, is simply that of identifying the language within each passage, and detecting the transitions between them. Thus, the key missing capability in order to achieve full-fledged multiscript, multilingual OCR is that of locating and identifying constituent monoscript/monolingual spans. Once these spans have been identified, the appropriately tailored recognition models may be applied to them.

Evidently, the task of locating and identifying constituent monoscript, monolingual spans is easier than that of recognizing the text within them. Again considering human performance, one need not understand any of the languages present in the document in order to be able to quickly and reliably segment the regions by script. For example, none of the present authors understands Korean, but all of us can reliably locate and identify Hangul material within a given multilingual page. While this may be an easy example due to the distinctiveness of Hangul relative to other scripts, anecdotally we've observed that we've become quite reliable in distinguishing and even identifying scripts for languages we do not understand; similar observations have motivated others to investigate vision-based / texture classification approaches to be used for script identification [7, 11, 12].

What makes the script identification (script ID) problem comparatively easy for a human? In part the answer must be the same as for what makes humans good at monolingual OCR: an innate faculty for vision and linguistics, supported by biology. But for script ID we believe there are two additional simplifying factors, both of which are available for exploit by an automated process.

First, neither script nor language tends to change rapidly from character to character, but rather each tends to persist over spans of at least several characters, on the order of words or phrases. This is a stable consequence of both linguistic and typographical convention. Second, the class

MOCR '13, Aug 24-24 2013, Washington, DC, USA
ACM 978-1-4503-2114-3/13/08.
http://dx.doi.org/10.1145/2505377.2505382

label set – i.e., the set of possible values for the script and language of a document fragment – is relatively small, consisting of no more than a few hundred at most, with typically much fewer values occurring within the same document.

The foregoing observations suggest a Markov-model-based approach to locating and identifying script-homogeneous span: first locate text lines, then treat each line as the output of a hidden Markov model, traversed in display order, where state represents the script label. Fortunately, HMM-based systems for OCR [5, 9] can be used exactly for this purpose. In previous work [10] we described a system that generalizes the HMM approach to work with multiple; heterogeneous features. Here we employ a purely HMM-based OCR system. We consider its use in script ID as a precursor to reapplying the same-structured system for OCR. In the first pass, decoding generates script ID labels rather than character identities; these labels are aggregated to hypothesize script-homogeneous segments. These segment-wise labels are then used in a second-pass to apply the correct OCR models in character recognition within each segment.

2. MODELS

We are adapting an HMM-based system designed for general purpose (line-level) OCR [10] to the task of script ID. The system is based on an HMM approach similar to that of [5, 9]. The character[1] HMMs are left-to-right, with a character-dependent number of states. Each state has a self-transition as well as transitions to the next and subsequent state (i.e., a Bakis model with a skip transition). The HMM emission models are Gaussian mixtures over a space of features extracted from sliding pixel windows. We are able to use two different feature sets. The first set is based on variable-sized Discrete Cosine Transform (DCT) of a narrow strip of pixels. A high-energy subset of the DCT coefficients are selected to yield a fixed, 45-dimensional feature vector. The second set is extracted by a deep neural network (DNN) [8], trained to discriminate between characters. Its input is a fixed-size window around a horizontal position being considered by the HMM, and this is fed through 4 hidden layers of 1000, 500, 250 and 45 nodes correspondingly into a softmax classification layer that attempts to distinguish character classes (or here, script classes in Script-HMM system, as described below). The network is trained by a gradient descent method, then the classification layer is discarded, and the preceding layer (of 45 nodes) is used to provide a set of features for the Gaussian mixture. Thus, the dimensionality of the feature vector is 45 for both the DCT and DNN variants.

2.1 Training

In all cases, the HMMs are trained on synthetic data: text-line images that have been digitally typeset in a variety of fonts and sizes, and distorted in various ways. The source text used for training was gleaned from various web sources on a per-language basis, in amounts sufficient to ensure that the accuracy is not significantly limited by lack of training data for any language. Evaluation and parameter tuning is always done on real (not synthetic) data, as discussed later. We have found that the generalization accuracy as measured on this real evaluation data for the OCR task (when script and language are known) is competitive with other state-of-the-art OCR systems, over a broad range of scripts and languages.

In addition to the use of synthetic data in training, we also make use of unlabeled scanned image data for the purpose of training the DNNs for some of the experiments reported here. Specifically, for the end-to-end results reported in Section 3.3, we use a preliminary, bootstrapping version of our OCR system to label (generate pseudo-ground-truth for) a relatively large amount of real scanned text images. We then identify a high-confidence subset of the results, and train the DNN feature extraction subsystem on that subset.

As will be discussed, the DNN-derived feature set is found to be generally more accurate than the DCT features on most scripts and languages.

We now describe two variant approaches to script ID that make use of the existing HMM-based OCR system.

2.2 Union-OCR model

Perhaps the simplest approach to the script ID task given an existing OCR system is to train it on the union of all the languages and scripts, perform OCR, then perform script (and language) identification on the resulting text. This can be done by combining the training data of the individual OCR systems (with sampling, to keep the size reasonable); hence we term this approach to script ID *Union-OCR*. This results in a mammoth system that is capable of recognizing text in any supported language and script, although it is less accurate than it would be if the script and language were known in advance, and is very slow. The output vocabulary of the system consists of about 24000 character classes. For comparison, the largest individual system (Simplified Chinese) has only about 8000. For script ID purposes we do not particularly care about the character-level accuracy as long as it's sufficient to identify the script, but its very low speed makes this system impractical for use for preliminary script ID within a multistage OCR system.

The OCR system outputs text; however, for this task we need to output the script the line is in. To accomplish this, each output character provides a vote for the script it belongs to, and we output the script with the most votes. Some special handling is done for Chinese and Japanese, as described in the next section.

2.3 Script-HMM system

Our Union-OCR system is slow because it is fundamentally trying to solve a much harder problem as a step toward solving an easier one: it is forced to predict the identities of various characters, whereas we only care about which script each character belongs to. A natural solution is to replace the character labels by script labels during training, then use the resulting system to directly output the desired script labels. We term this approach *Script-HMM*. Relative to Union-OCR, Script-HMM reduces the number of labels that must be distinguished from 24000 to around 25. Besides the script names, we use a separate label for non-script-specific digits, another for punctuation and symbols, and another for spaces. We also include some scripts in the hypothesis space for which we are not reporting evaluation results. We use the Unicode database for mapping from letters to scripts, and use the standard Unicode set of scripts (ISO 15924), with some exceptions for Japanese and Chinese, as we now discuss.

[1]Throughout, *character* refers to a fully composed grapheme cluster, possibly comprising multiple Unicode points.

Japanese is mostly written using three scripts: Kanji, Katakana and Hiragana. Each line may contain a mix of any of them. For our purposes we do not need to distinguish between the latter two, since the Japanese-specific OCR model handles all three. We do need to distinguish Japanese from Chinese however, and this is made complicated by the fact that in our approach we treat Kanji the same way as the Chinese characters, since they share the same Unicode space, and in many cases (though not all) they look the same as the corresponding Chinese character.

Chinese poses an additional challenge of having two different scripts: Simplified and Traditional (denoted *Chinese-S* and *Chinese-T*, respectively) share a large number of characters, not just visually (like Latin, Cyrillic and Greek), but also in digital representation. For our purpose, however, we need to distinguish them. To do so, we treat Chinese as consisting of three scripts: Common Han, Simplified, and Traditional. For each Unicode point in the Han blocks (this includes Japanese Kanji) we use the UniHan database [2] to map it to one of these three scripts.

The system is then trained, and lines are recognized, but now the system output simply contains the script estimated for each hypothesized character location. We then need to determine the primary script for each line, based on the estimated scripts for each position. One approach would be to train another classifier to perform this task, using the result of our script ID as input. However, in this paper we use a simpler, voting based approach. Specifically, we take the most frequently occuring script label within each line to the the primary script for the line. For Chinese, we count Common Han characters as both Simplified and Traditional. For Japanese, we treat all Chinese characters as also Japanese if there are at least 10% purely Japanese characters (Katakana or Hiragana) in that line.

3. EVALUATION DATA AND RESULTS

3.1 Evaluation Data

Our development and evaluation data consists of roughly 2000 grayscale text line images at variable resolution averaging about 200 ppi, in each of 55 languages, representing 18 distinct scripts, for a grand total of about 100,000 lines. The lines were extracted from a wide sampling of scanned books; for most languages a single book contributes no more than two lines. Thus a wide variety of typefaces, publication dates, and subject matter are represented in the data. All lines were verified as being in the target language by a native speaker of that language. We use half of this data for development (tuning), and half for evaluation. Note that training data is separate, as discussed in Section 2.1.

All languages except Serbian are considered to be in one script; Serbian has lines in both Latin and Cyrillic. We treat Japanese as a single script for evaluation purposes.

Often, text lines contain fragments in multiple scripts – for instance, an English phrase within in a Bengali sentence. Both the Union-OCR and Script-HMM methods can be easily adapted to classify script at the sub-line (e.g., phrase or word) level, as is discussed briefly in Section 4. However, in the present study for the purpose of computing script-classification error rates, we assign a single script label to the entire line. When an evaluation line does contain multiple scripts, to assign it a reference label, a judgment was made as to which is the primary script, considering the overall text from which the line was extracted.

Table 1: Evaluation data

Script	# of Langs	Example languages
Latin	29.5	English, Vietnamese
Cyrillic	6.5	Russian, Serbian(0.5)
Arabic	2	Arabic, Farsi
Hebrew	2	Hebrew, Yiddish
Greek	1	Greek
Armenian	1	Armenian
Japanese	1	Japanese
Korean	1	Korean
Chinese-S	1	Chinese (simplified)
Chinese-T	1	Chinese (traditional)
Devanagari	2	Hindi, Marathi
Bengali	1	Bengali
Gujarati	1	Gujarati
Kannada	1	Kannada
Malayalam	1	Malayalam
Tamil	1	Tamil
Telugu	1	Telugu
Thai	1	Thai
Total	55	

The detailed data breakdown by script is given in Table 1. Note that Latin-script data accounts for the majority of our data, by virtue of most of the considered languages being in Latin script, and each language having equal representation. Thus, an aggregated measure of accuracy estimated on this data will weigh languages written in Latin script more heavily than others. In terms of assessing end-to-end accuracy of an omni-script OCR system that uses this method of script ID, such an aggregated measure can be appropriate if the collection being digitized is similar in language breakdown and proportion to the evaluation set. Note that equal representation by language actually results in less Latin bias than would have been the case had we sampled truly at random from the underlying scanned book collection, without equalizing language representation. Thus, for the purpose of single-number accuracy assessment, the degree of Latin bias in our data obtained by uniform language sampling represents a middle ground between the two extremes of uniform script representation and uniform collection sampling. In addition to single-number overall accuracy, it is also important to consider un-aggregated accuracy for specific languages, to answer questions such as, "how does the end-to-end OCR system do on the average Armenian document, without being told in advance that it's Armenian?" While not reported in detail here, such considerations inform the present work.

3.2 Results

We ran both the Union-OCR and the Script-HMM models on the dataset described above, using both the DCT and DNN features. DNN features were discriminatively trained appropriately for each system; i.e., for character accuracy in Union-OCR and for script accuracy in Script-HMM. We report line-level error rate (number of misclassified lines in each script, divided by the total number of lines in that script), as well as per-line speed (in seconds). The results are summarized in Table 2.

For the Script-HMM model, the most common errors are

Table 2: Summary Results

Model	Features	Script-ID error rate (%)	Runtime (sec/line)
Union	DCT	4.50%	20.8
Union	DNN	1.71%	15.0
Script-id	DCT	12.95%	1.58
Script-id	DNN	1.73%	0.16

Table 3: Top Script Confusions

Script	Union DNN top confusion	Script-HMM DNN top confusion
Latin	Cyrillic	Kannada
Cyrillic	Latin	Latin
Arabic	Latin	Latin
Hebrew	Cyrillic	Latin
Greek	Latin	Latin
Armenian	Latin	Arabic
Japanese	Chinese-S	Chinese-T
Korean	Latin	Latin
Chinese-S	Chinese-T	Chinese-T
Chinese-T	Chinese-S	Chinese-S
Devanagari	Latin	Bengali
Bengali	Latin	Latin
Gujarati	Arabic	Latin
Kannada	Telugu	Telugu
Malayalam	Latin	Arabic
Tamil	Latin	Latin
Telugu	Kannada	Kannada
Thai	Latin	Lao

between the two Chinese scripts. These errors constitute about a third of the total (see Table 4). This is perhaps unsurprising, as the characters in these lines are very similar when viewed as a group; in fact most are the same.

For the Union model, however, this is a minor source of error, constituting less than 5% of the errors. This is because the characters are recognized individually, and the distinction between a specific simplified and traditional character can be made quite easily. The largest total source of errors for the Union model is the confusion between Cyrillic and Latin which are the two most common scripts and have many similar or identical letters, but the actual error rate for both of these is low. If we examine the per-script error rates of the Union model, the script that has the most errors is Gujarati (15%) which is unsurprising as it is currently also one of the most challenging languages for our per-language OCR systems, largely because of limitations in available data.

The detailed breakdown of errors is provided in Table 4. In comparing DCT and DNN models we can see that while having a DNN model is always helpful (row 1 vs. 2 of Table 2), it is absolutely critical when script-ID is the direct output of recognition (row 3 vs. 4). Apparently, while the dimensionality-reduced DCT feature vector retains information needed to represent character shapes, and hence distinguishes them somewhat well, script identity is effectively a latent variable, and the DNN's ability to discriminatively learn the feature vector specifically for the task gives it a decisive advantage.

Table 3 shows the top script ID confusions made by the

Table 4: Detailed results (Script-ID error rate)

Script	Union DCT	Union DNN	Script-id DCT	Script-id DNN
Latin	1.71%	0.16%	11.38%	0.35%
Cyrillic	3.01%	1.11%	13.01%	1.70%
Arabic	0.39%	0.27%	0.27%	0.27%
Hebrew	0.63%	0.15%	1.16%	0.09%
Greek	3.35%	0.83%	23.95%	0.83%
Armenian	16.83%	13.60%	41.33%	1.81%
Japanese	18.62%	14.49%	31.43%	9.96%
Korean	10.65%	5.56%	14.32%	2.24%
Chinese-S	7.83%	3.53%	46.14%	33.12%
Chinese-T	23.76%	10.39%	30.00%	10.65%
Devanagari	1.47%	0.22%	3.17%	0.82%
Bengali	7.39%	6.24%	6.59%	5.89%
Gujarati	49.15%	15.40%	35.06%	1.16%
Kannada	5.98%	1.56%	30.29%	2.95%
Malayalam	2.14%	0.55%	8.66%	0.74%
Tamil	2.75%	2.12%	10.18%	2.01%
Telugu	14.87%	6.43%	10.74%	1.37%
Thai	7.73%	1.98%	4.17%	7.44%
Combined	4.49%	1.71%	12.95%	1.73%

two DNN-based systems. Note that often Latin is the top incorrect script-ID output for both systems, even in cases where the true script is markedly dissimilar – but in most of those cases, the overall number of confusions (script ID error rate in Table 4) is quite low for the respective system, and the effect may be due to a tuning bias from the prevalence of Latin-script data overall.

Also note in the table that Thai is confused with Lao by the Script-HMM system, although Lao isn't listed as one of the scripts in the left column. This is because the systems are actually aware of — and have as their hypothesis space — a larger set of scripts than those represented by the evaluation data. Thus, if the hypothesis space were restricted to include only the evaluation scripts, the script-ID error rate would likely be slightly lower than what is reported.

3.3 Multipass OCR: end-to-end accuracy

We were able to perform OCR experiments on this dataset, using a 3-stage process: first, script-id (using DNN) system; second, a script-specific OCR system followed by a language identification step; third, a language-specific OCR system.

We consider Character Error Rate (CER), defined as the sum of edit distances between the produced and the correct transcriptions over all the test instances, divided by the total length of the correct transcriptions.

If the script and language are correctly known in advance, CER averaged over all languages represented in our evaluation data is 3.28%. While not the focus of this paper, we note as an aside that the accuracy of our system is competitive with that of other state-of-the-art OCR systems, when the languages and script are given and supported by the systems.

If we assume the script is known in advance but the language is not (idealizing stage 1), we obtain an average CER of 3.55%. The 0.27% increase in CER is attributable to errors in language identification performed on the result of using the correct stage-2 script-specific OCR system.

Finally, using the full 3-step system in the above para-

graph, assuming no advance knowledge of either language or script, we obtain the CER of 4.05%. Thus the contribution to overall CER that is attributable to errors in script and language identification together, is about 0.77%.

4. DISCUSSION AND FUTURE WORK

We have demonstrated that an HMM-based OCR system can be easily adapted for the purpose of script ID and reach very high accuracy at high speed. This makes it possible to create an efficient "universal" OCR system as a cascade of two architecturally identical recognition components, where the first one is a script ID system (the main focus of this paper), and the second a script-specific system, which in turn can produce either the final output when the script and language are one-to-one (e.g., in case of Armenian), or which can be further reprocessed by language identification and language-specific systems for scripts that are used by several languages (e.g., Latin script).

We have found that training the feature set specifically for the script ID task is of great importance – using the same DNN features as for OCR without re-training them in the script ID context has not worked well in our experiments. This is an additional motivation for using the flexible, data-driven approach to learning features based on recent work in distributed deep networks.

For comparison purposes, we also demonstrate a Union system that is capable of distinguishing between over 24000 characters with high enough accuracy to allow for reliable script (and, in principle, language) identification.

To mitigate error propagation from script ID, we could attempt to make a soft script-ID decision and perform OCR using a combined model of the hypothesized scripts, each weighted by the confidence. However, the models thus combined could have a large effective hypothesis space, especially for East Asian languages, and may therefore prove prohibitively expensive. For this reason, improving our ability to distinguish Chinese and Japanese decisively at the script-ID level remains an important challenge.

For lines containing multiple scripts, we would like to be able to identify the parts of the lines that are in each script, in order to allow the second-stage OCR system to use the appropriate models for the scripts and languages present in the various sub-line segments. The beginnings of this capability are already present in the systems we described, since they provide as an intermediate result a script label for each hypothesized character position: sub-line script identification can be performed by voting within a sliding window, say, or alternatively, by dynamic programming. Specifically, each script can correspond to a state, which emits its script's label with high probability, and those of the other scripts with lower probability (uniform or, as an optional refinement, non-uniform). Transitions are penalized to strike a balance between the ability to accommodate multiple scripts, and the tendency to switch erratically and erroneously from one script to another. In addition, if per-label confidences are available, these could be incorporated as additional information within the dynamic programming search to yield an improved segmentation-by-script within each line. We leave the detailed development within-line segmentation search as future work.

5. REFERENCES

[1] Speech language & multimedia: Optical character recognition. http://www.bbn.com/technology/speech/optical_character_recognition.

[2] Unicode standard annex 38: Unicode han database (UNIHAN). Technical report, The Unicode Consortium, 2013. http://www.unicode.org/reports/tr38/tr38-13.html.

[3] Abbyy. ABBYY FineReader professional edition recognition languages. http://finereader.abbyy.com/recognition_languages.

[4] Abbyy. ABBYY FineReader version 11 user's guide. http://www.abbyy.com/fr11guide_en.pdf.

[5] 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*, pages 842–846, Washington, DC, USA, 1997. IEEE Computer Society.

[6] T. M. Breuel. The OCRopus open source document analysis and OCR system. https://code.google.com/p/ocropus.

[7] A. Busch, W. Boles, and S. Sridharan. Texture for script identification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11):1720–1732, 2005.

[8] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems 25*, 2012.

[9] P. Dreuw, G. Heigold, and H. Ney. Confidence and margin-based MMI/MPE discriminative training for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 2011.

[10] D. Genzel, A. C. Popat, N. Spasojevic, M. Jahr, A. W. Senior, E. Ie, and F. Y.-F. Tang. Translation-inspired OCR. In *ICDAR*, pages 1339–1343, 2011.

[11] P. Hiremath and S. Shivashankar. Wavelet based co-occurrence histogram features for texture classification with an application to script identification in a document image. *Pattern Recognition Letters*, 29(9):1182 – 1189, 2008.

[12] P. B. Pati and A. G. Ramakrishnan. Word level multi-script identification. *Pattern Recognition Letters*, 29:1218–1229, 2008.

[13] R. Smith. Tesseract manual page. http://tesseract-ocr.googlecode.com/svn-history/r725/trunk/doc/tesseract.1.html.