Scene Text Recognition and Retrieval for Large Lexicons

Udit Roy[1], Anand Mishra[1], Karteek Alahari[2] and C.V. Jawahar[1]

[1]CVIT, IIIT Hyderabad, India   [2]Inria[*]

Abstract. In this paper we propose a framework for recognition and retrieval tasks in the context of scene text images. In contrast to many of the recent works, we focus on the case where an image-specific list of words, known as the small lexicon setting, is unavailable. We present a conditional random field model defined on potential character locations and the interactions between them. Observing that the interaction potentials computed in the large lexicon setting are less effective than in the case of a small lexicon, we propose an iterative method, which alternates between finding the most likely solution and refining the interaction potentials. We evaluate our method on public datasets and show that it improves over baseline and state-of-the-art approaches. For example, we obtain nearly 15% improvement in recognition accuracy and precision for our retrieval task over baseline methods on the IIIT-5K word dataset, with a large lexicon containing 0.5 million words.

1 Introduction

Text can play an important role in understanding street view images. In light of this, many attempts have been made to recognize scene text [1–6]. Scene text recognition is a challenging problem and its recent success is mostly limited to the *small lexicon setting*, where an image-specific lexicon containing the ground truth word is provided. Typically, these lexicons contain only 50 words [3]. This setting has many practical applications, but it does not scale well. As an example consider the scenario of assisting visually-impaired people in finding books by their titles in a library. Here the lexicon is populated with all the book titles. In this case, the small lexicon setting becomes less accurate as the lexicon sizes can range from a few thousands to a million. For instance, when lexicon size increases from 50 to 1000, the recognition accuracy drops by more than 10% [6, 7]. In other words, the general problem of scene text recognition, i.e., recognition with the help of a large lexicon (say a million dictionary words) is far from being solved. In this paper, we investigate this problem.

One way to address the task of recognizing scene text is to pose the problem in conditional random field (CRF) framework and obtain the maximum a posteriori (MAP) solution as proposed in [3, 4, 7–10]. In these frameworks, an

---

Word Image	Top-5 diverse solutions (ranked)
PITT	PITA, PASP, ENEP, **PITT**, AWAP
COM	AUM, NIM, **COM**, MUA, PLL
MONSTER	MINSTER, MINSHER, GRINNER, MINISTR, **MONSTER**
BIKE	BRKE, BNKE, **BIKE**, BAKE, BOKE
This	TOLS, TARS, **THIS**, TOHE, TALP

Fig. 1. Examples where the MAP solution is incorrect, as the pairwise priors become too generic when computed from large lexicons. The set of top-5 diverse solutions contains the correct result.

energy function consisting of unary and pairwise potentials is defined, and the minimum of this function corresponds to the text contained in the word image. These methods demonstrated successful results in a small lexicon setting primarily due to the fact that the pairwise terms are computed with this lexicon have a positive bias towards the ground truth word. However, when the pairwise terms are computed from large lexicons, they become too generic, and often in such cases the MAP solution does not correspond to the ground truth. Besides this, MAP solutions suffer from drawbacks, such as (i) approximation errors in inference, (ii) poor precision/recall for character detection, (iii) weak unary and pairwise potentials. Consider the word "PITT" shown in Fig. 1 as an example. The MAP solution for the word is "PITA", which is incorrect. Our approach addresses this problem by using the top-M solutions to ultimately find text that is most likely contained in the image.

We begin by generating a set of candidate words with M-best diverse solutions [11]. With these potential solutions, we refine the large lexicon by removing words from it with a large edit distance to any of the candidates, and then recompute the M-best diverse solutions. These two steps are repeated a few times, which ultimately results in set of words most likely to represent the word contained in the image. Then a desired solution can be picked using various means (e.g., using minimum edit distance based correction using a lexicon). We show significant performance gain for recognition tasks in the large lexicon setting using this framework. We also present an application of computing the top-M solutions, i.e., text to image retrieval, where the goal is to retrieve all the occurrences of the query text from a database of word images. We will show that our strategy of re-ranking the words with the refined lexicon improves the performance over baseline methods.

Related Work. The problem of cropped word recognition has been looked at in two broad settings: with an image-specific lexicon [3–6, 10] and without the help of lexicon [1, 7, 8]. Approaches for scene text recognition typically follow a two-step process (i) A set of potential character locations are detected either by binarization [1, 2] or sliding windows [3, 4], (ii) Inference on CRF model [4, 7], semi Markov model [1, 8], finite automata [9] or beam search [2] in a graph (representing the character locations and their neighborhood relations) is performed.
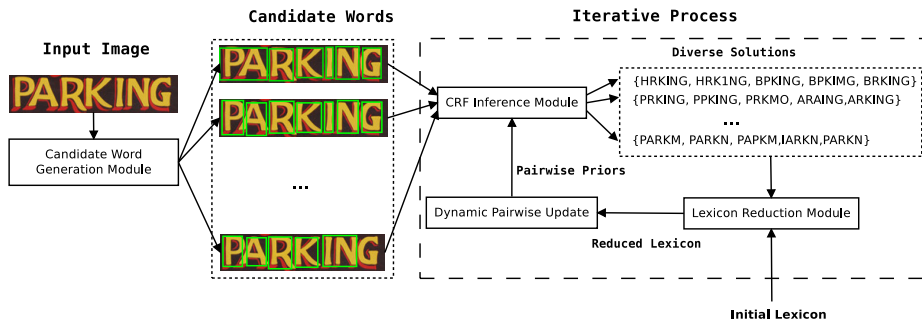
Fig. 2. Overview of the proposed framework. The input image is passed on to a multiple candidate word generation module which generates candidate words, each with a set of character regions and their corresponding unary potentials. With the help of an initial lexicon, pairwise priors are computed and diverse solutions are inferred from all the candidate words. These candidates are then used to reduce the lexicon. This process is repeated with the reduced lexicon until the lexicon is refined to a small size. The final solution is the word in the full lexicon closest to the diverse solutions computed in the last iteration.

These approaches work well especially in small lexicon settings, but suffer from two main drawbacks: (i) Obtaining a single set of true character windows in a word image in these methods is difficult, (ii) Pairwise information gets less influential as the lexicon size increases. We adopt a similar framework in this paper, but propose crucial changes to overcome the issues of previous approaches. First, we generate multiple word hypotheses and derive a set of candidate words likely to represent the word image. Second, we present a technique to prune the large lexicon based on edit distances between the candidate solutions and lexicon words. This proposed method allows us to significantly reduce the lexicon size and make the priors more specific to the image. Third, unlike prior works which yield a single solution, our method is also capable of yielding multiple solutions, and is applicable to the text-to-image retrieval task.

The remainder of the paper is organized as follows. In Section 2.1, we present CRF framework for word recognition. We utilize multiple segmentations of word images to obtain potential character locations in Section 2.2. We then present details of the inference method in Section 2.3. Our lexicon reduction and pairwise term update steps are described in Section 2.4. The two problem settings, i.e., recognition and retrieval, are then discussed Section 3. Section 4 describes the experiments and shows results on public datasets. Implementation details are also provided in this section. We then make concluding remarks in Section 5.

2   Proposed Method

We model the scene text recognition task as an inference problem on a CRF model, similar to [4], where unary potentials are computed from character clas-

sification scores and pairwise potentials from the lexicons. Small lexicon based pairwise potentials often help to recover from the errors made by character classification [12, 13]. However, when the pairwise potentials are computed from large lexicons, they become too generic, and the overall model cannot cope with erroneous unary potentials. To overcome this issue, starting from a large lexicon recognition problem, we automatically refine the problem statement and convert it to a small lexicon inference task.

The framework has the following components, as shown in Fig. 2: (i) Candidate word generation module, where we generate multiple words with each word as a set of characters spanning over the image, (ii) CRF inference module, where each word is represented as a CRF and inferred to obtain diverse solutions, and (iii) Lexicon reduction module, where we prune the lexicon by removing distant words after re-ranking the lexicon with a novel group edit distance computed using the diverse solutions. It is accompanied by re-computation of pairwise potentials which become image specific as the lexicon size decreases. We use different stopping criteria for recognition and retrieval tasks as we alternatively reduce our lexicon and infer solutions.

2.1 CRF framework

The CRF is defined over a set of random variables $x = \{x_i | i \in \mathcal{V}\}$, where $\mathcal{V} = \{1, 2, ..., n\}$, denotes the set of n characters in a candidate word. Each random variable x_i denotes a potential character in the word, and can take a label from the label set $\mathcal{L}$ containing English characters and digits. The energy function, $E : \mathcal{L}^n \to \mathbb{R}$, corresponding to a candidate word can be typically written as the sum of unary and pairwise potentials:

$$E(x) = \sum_{i \in \mathcal{V}} E_i(x_i) + \sum_{(i,j) \in \mathcal{N}} E_{ij}(x_i, x_j), \tag{1}$$

where $\mathcal{N}$ represents the neigbourhood system defined over the candidate word. The set of potential characters is obtained by a segmentation procedure, discussed in Section 2.2.

Unary Potentials. The unary potential of a node is determined by the SVM confidence score. The unary term $E_i(x_i = c_j)$ represents the cost of a node x_i taking a character label c_j, and is defined as:

$$E_i(x_i = c_j) = 1 - p(c_j | x_i), \tag{2}$$

where $p(c_j | x_i)$ denotes the likelihood of character class c_j for node x_i.

Pairwise Potentials. The pairwise cost of two neighbouring nodes x_i and x_j taking a pair of character labels c_i and c_j is defined as,

$$E_{ij}(x_i, x_j) = \lambda_l(1 - p(c_i, c_j)), \tag{3}$$

where $p(c_i, c_j)$ is the bigram probability of the character pair c_i and c_j occurring together in the lexicon. The parameter λ_l determines the penalty for a character pair occurring in the lexicon. Similar to [4], we use node-specific prior, where the priors are computed independently for each edge from the bigrams in the lexicon that have the same relative position to that of the edge in the CRF. This enforces spatial constraints on prior computation, and are found to be more effective than the standard node prior [4].

2.2   Generating Candidate Words

Obtaining potential character locations with a high recall is desired for our approach. There are two popular methods for character extraction based on: (i) sliding window [4, 7], (ii) binarization [1, 2]. We follow the binarization based approach as it results in fewer potential character locations, in the form of connected components (CCs), than those generated by the sliding window based method. This avoids redundant character windows with similar size at a specific image location. Binarization based methods reduce the number of candidate windows with threshold parameters and by leveraging fast pruning techniques on the CCs. To ensure that all the characters are present in the candidate windows as CCs, we combine results with different thresholds. This significantly improves the character recall at the cost of generating some false windows that can be overcome in the latter steps.

To remove obvious false windows we use heuristics based on information such as character sizes, aspect ratio and spatial consistency, followed by a character specific non-maximal suppression. This step removes false positive windows occurring in the background or unwanted foreground text elements like text bounding boxes. We also detect other anomalous windows, like holes in characters and invalid windows present within the characters, by finding configurations where a smaller window is contained completely within a larger window, and then remove the smaller one.

After pruning, we get a set of potential character windows which are used to generate candidate words. We first build a graph by joining the potential character windows which are spatially consistent and likely to be adjacent characters. In other words, the windows are connected with an edge if (i) overlapping windows have an overlap less than a threshold, and (ii) non-overlapping windows are less than a threshold away. We remove a few edges connecting windows whose width or height ratio is not in a desired range, to ensure that only character-to-character links are preserved. Then we estimate the most probable words for further analysis as described in the following.

Selection of Candidate Words. Our objective is to find a set of probable candidate words from the directed graph described above. We define a candidate word as a set of character windows representing the text present in the image. We first find the most probable start and end character windows by selecting windows close to the left and right image boundaries. Representing these start and end windows as candidate start and end nodes, we find possible connected

paths (i.e., candidate words) between all pairs of start and end nodes using a depth first all paths algorithm [14]. We reject candidate words which do not cover sufficient area over the word image. The shortlisted candidate words are represented as a CRF, inferred and re-ranked according to their minimum energy value which is normalized by the number of nodes in the CRF. The least energy candidate words are retained for the subsequent stage as the correct candidate words assuming they have nodes with better unary potentials.

2.3   Diversity Preserving Inference

Once the optimal candidate words are selected, we infer the text each of them contains by minimizing the energy (1). However, the minimum energy solution of the word may be at times incorrect due to poor unary or pairwise potentials. Hence, diverse solutions are preferred a over single solution. Inspired by [11], we obtain M-best solutions instead of one MAP solution. This is done for all the selected candidate words from the previous stage individually. We approach the problem of diversity preserving inference with a greedy algorithm. First, we obtain the MAP solution with TRW-S [15] and then, the next solution is defined as the lowest energy state with minimum similarity from the previously obtained solutions.

Rewriting the problem of optimizing the energy function (1) we obtain,

$$\min_{\boldsymbol{\mu}} \quad \sum_{i\in\mathcal{V}}\sum_{s\in\mathcal{L}}\alpha_i(s)\mu_i(s) + \sum_{i,j\in\mathcal{N}}\sum_{s,t\in\mathcal{L}}\alpha_{ij}(s,t)\mu_{ij}(s,t), \tag{4}$$

where $\alpha_i(s)$ is the unary potential and $\alpha_{ij}(s,t)$ is the pairwise potential. The terms $\mu_i(s)$ and $\mu_{ij}(s,t)$ are their corresponding binary indicator variables. This function (4) can be re-written with standard constraints on unary and pairwise potentials as well as the diversity constraint (to get the second best solution) in the form of function $\Delta(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu})$, where $\hat{\boldsymbol{\mu}}$ is the best solution found after inferring with the diversity constraint as follows,

$$\min_{\boldsymbol{\mu}} \quad \sum_{i\in\mathcal{V}}\sum_{s\in\mathcal{L}}\alpha_i(s)\mu_i(s) + \sum_{i,j\in\mathcal{N}}\sum_{s,tin\mathcal{L}}\alpha_{ij}(s,t)\mu_{ij}(s,t), \tag{5}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{L}}\mu_i(s) = 1, \tag{6}$$

$$\sum_{s\in\mathcal{L}}\mu_{ij}(s,t) = \mu_j(t), \quad \sum_{t\in\mathcal{L}}\mu_{ij}(s,t) = \mu_i(s), \tag{7}$$

$$\Delta(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}) \geq k, \tag{8}$$

$$\mu_i(s), \mu_{ij}(s,t) \in \{0,1\}. \tag{9}$$

Here, (6) and (7) denote the constraints on unary and pairwise potentials. The constraint (8) is the diversity measure that has to be greater than a scalar k. The Langrangian relaxation of this optimization problem is formed by the dualizing

the constraint (8), which yields,

$$\min_{\boldsymbol{\mu}} \quad \sum_{i \in \mathcal{V}} \sum_{s \in \mathcal{L}} \alpha_i(s)\mu_i(s) + \sum_{i,j \in \mathcal{N}} \sum_{s,t \in \mathcal{L}} \alpha_{ij}(s,t)\mu_{ij}(s,t) - \lambda(\Delta(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}) - k). \quad (10)$$

Using a dot product dissimilarity (Hamming distance) as our Δ function we obtain,

$$\min_{\boldsymbol{\mu}} \quad \sum_{i \in \mathcal{V}} \sum_{s \in \mathcal{L}} \alpha_i(s)\mu_i(s) + \sum_{i,j \in \mathcal{N}} \sum_{s,t \in \mathcal{L}} \alpha_{ij}(s,t)\mu_{ij}(s,t) - \lambda(-\hat{\boldsymbol{\mu}} \cdot \boldsymbol{\mu} - k), \quad (11)$$

which can be re-written as,

$$\min_{\boldsymbol{\mu}} \quad \sum_{i \in \mathcal{V}} \sum_{s \in \mathcal{L}} (\alpha_i(s) + \lambda\hat{\mu}_i(s))\mu_i(s) + \sum_{i,j \in \mathcal{N}} \sum_{s,t \in \mathcal{L}} \alpha_{ij}(s,t)\mu_{ij}(s,t) + \lambda \cdot k. \quad (12)$$

In summary, only the unary potentials need to be modified by adding the original solution scaled by the diversity parameter λ. The TRW-S [15] algorithm can be utilized again to infer the second best solution.

2.4   Lexicon Reduction

Once the solutions are obtained from all the selected candidate words, they are used to reduce the large lexicon and compute pairwise potentials iteratively. We prefer to use the diverse solutions over the MAP solution as they maximize the chances of inferring the correct solution. Our first iteration involves shrinking the lexicon to a smaller size, i.e., 50. This is done by re-ranking the lexicon words using group edit distance (described below) to the solutions obtained, and retaining the top matches. This iteration reduces the lexicon size significantly and retains a small subset with a high recall of ground truth words. From the second iteration onwards, we use the new pairwise potentials (computed from the reduced lexicon) and re-infer the diverse solutions. Thereafter, we remove the word in the lexicon with maximum group edit distance from the diverse solutions. This lexicon reduction procedure is summarized in Algorithm 1.

Group Edit Distance. The standard way of re-ranking a lexicon using a single solution is by computing the edit distance between the solution and all the lexicon words. However in a multiple solution scenario, where diverse solutions from multiple words come into the picture, the correct inferred label is most likely to be present in the solution set. To be able to compute the edit distance between a solution set and lexicon, we find the minimum edit distance for each lexicon word from the solution set. This modification ensures that if the ground truth is very close to one of the diverse solutions, it will be ranked higher than others in the lexicon.

3   Recognition and Retrieval

The method described so far reduces the size of the lexicon by alternating between the two steps of estimating candidate words and refining the lexicon. We then use this lexicon for the recognition and retrieval tasks.

Input: Candidate words, Initial lexicon L_i, Reduced lexicon size r
Output: Reduced lexicon L_r
Initialization: $L_r = L_i$
while $size(L_r) > r$ **do**

> **1:** Perform inference on all the candidate words to obtain M diverse solutions (Section 2.3)
> **2:** Remove the lexicon word w with the maximum group edit distance from M diverse solutions
>
> $$L_r = L_r - \{w\}$$
>
> **3:** Compute new pairwise priors from the reduced lexicon

end

Algorithm 1: The lexicon reduction process alternates between removing words from the lexicon and re-computing the pairwise potentials.

Recognition. In the recognition task, our goal is to associate a text label to a given word image. The process begins by forming multiple candidate words using the graph construction described in Section 2.2. Candidate words are re-ranked and k optimal candidate words are retained. We reduce the lexicon (using the method in Section 2.4) to a size of 10 words and obtain diverse solutions with the newly computed pairwise potentials from this reduced lexicon. We now select a word from the original lexicon with the minimum group edit distance from the diverse solutions as our result.

Retrieval. In a retrieval task, our objective is to retrieve word images for a given text query word from a dataset. The traditional approach would be to reduce the lexicon for each word to size one (hereafter referred to as singleton lexicon), and search for the query word in the singleton lexicons of all the words in the dataset. However, since this approach is prone to failures in recognition, we relax the constraint of reducing the lexicon to size one, and instead reduce the lexicon to a very small size, say five words. This allows us to overcome recognition errors and retrieve word images where the ground truth in present in the reduced lexicon but not in the singleton lexicon. Word images with reduced lexicons having low similarity among their constituent words are further reduced to a singleton lexicon. We measure the similarity of words in the lexicon with a measure called average edit distance (AED) which is defined as,

$$\text{AED} = \frac{1}{P} \sum_{w_i, w_j \in L^P} ED(w_i, w_j), \tag{13}$$

where L^P is the lexicon with P words and $ED(w_i, w_j)$ is the edit distance between words w_i and w_j. A low AED implies that the reduced lexicon has similar words and hence, one more lexicon reduction iteration may result in arbitrary loss of ground truth from the reduced lexicon. On the other hand, in cases with high AED score, the words in the reduced lexicon are different from each other.

As a preprocessing step to our retrieval task, we prepare the dataset by reducing the lexicons for each word image to either a singleton or a reduced lexicon. The lexicon is reduced iteratively to a size n and the AED score is computed. If the score is found to be less than θ (i.e., showing high similarity among the words in the lexicon) we terminate the lexicon reduction process and associate the reduced lexicon of size n with the word image. We continue the process to get a singleton lexicon otherwise. For a given query word, we find all the word images in the dataset that have the query word in their respective singleton or reduced lexicons. All the selected images are then ranked using a combined score computed as the weighted sum of: (i) the lexicon size (one or n), and (ii) the position of the query word in the ranked lexicon. Note that in each iteration of the lexicon reduction process, the lexicon is ranked by group edit distance from the diverse solutions (Section 2.4). The intuition behind this combined score is that words retrieved from a small lexicon and words that rank better in the lexicon are more likely to be the correct retrieval, i.e., a low combined score. We give more weightage to the first term, as word images with smaller lexicons are more likely to retain the ground truth.

4   Experimental Analysis

4.1   Datasets

We used three public datasets, namely IIIT 5K-word dataset [7], ICDAR 2003 [16] and Street View Text (SVT) [17, 18] in our evaluations.

IIIT 5K-word. The IIIT 5K-word dataset contains 5000 cropped word images from scene texts and born-digital images, harvested from Google image search engine. This is the largest dataset for natural image word spotting and recognition currently available. The dataset is partitioned into train (2000 word images) and test (3000 word images) sets. It also comes with a large lexicon of 0.5 million words. Further, each word is associated with two smaller lexicons, one containing 50 words (known as small lexicon), another with 1000 words (known as medium lexicon).

ICDAR 2003. The test dataset contains 890 cropped word images. They were released as a part of the robust reading competitions. We use small lexicons provided by [17] of size 50 for each image in this dataset.

SVT. The SVT dataset contains images taken from Google Street View. Since we focus on the word recognition task, we used the SVT-WORD dataset, which contains 647 word images and a 50-word sized lexicon for each image.

4.2   Multiple Candidate Word Generation

We binarize the image using Otsu's method [19] with ten thresholds equally spaced over the grayscale range. This provides a good set of potential character locations, which are used to construct the graph (Section 2.2). The overlap,

Word Image	Iteration 1	Iteration 2	Iteration 3	Iteration 4
KINGFISHER	FGAIEESHER	FGAIERSHER	KINGFISHER	KINGFISHER
that	NHAI	AHAI	AHAI	THAT
Mammoth	MAITOTA	MAITOTA	MACTOTH	MAMMOTH
THIS	THTL	THEL	THEL	THIS

Fig. 3. Effect of the lexicon reduction technique on the inferred label. Here we show four iterations for each example. We observe that with stronger pairwise potentials the method recovers from the errors in the MAP solution.

aspect ratio and width/height range parameters associated with the graph construction are chosen by cross-validating on an independent validation set. We add an edge between two overlapping windows if their X-axis projection intersection is less than 25% of the left window width. If they are non-overlapping, they must be no farther than 50% of the left window width. We remove edges with window width ratio or height ratio more than a factor of 4. For non-maximal suppression, we use 80% overlap as our threshold. Once the graph is constructed, all candidate words are found (Section 2.2). We then re-rank them using their energy score (1) normalized by word length and select the top-10 candidate words for the lexicon reduction phase.

4.3   Diversity Preserving Inference

We train one-vs-all character classifiers with linear SVM for unary potentials, as described in [20], with dense HOG features [21] from character images. To obtain multiple CRF solutions we infer the top-5 diverse labels by modifying the unary potentials in each iteration (Section 2.3). The λ parameter in (12) is set by cross validation. We found $\lambda = 0.1$ to be an optimal value to moderate the influence of diversity. Note that with a small λ, the unary potentials will be modified by a very small amount in the next iteration, which will result in inferring the same solution. On the other hand, a large λ gives very diverse solutions, and in some cases words that are significantly different from each other.

4.4   Recognition

In our recognition experiment, we stop the lexicon reduction process when the reduced lexicon reaches a size of 10, and then find the nearest word in the original lexicon with minimum group edit distance from the most recently inferred solution set. We evaluate the performance of the system by checking if the nearest word is the ground truth or not.

For the large lexicon experiments, the group edit distance re-ranking becomes computationally expensive due to the lexicon size. To speed up the process, we represent each word by its character histogram and build a k-NN classifier. Now, for a given solution set and a lexicon, we first find the top-100 nearest neighbours

Table 1. Word recognition accuracy comparison between various CRF and non-CRF methods. A word is said to be correctly recognized if the word nearest to result of a method in the lexicon is the ground truth. We compute top-5 diverse solutions and select one solution from the full lexicon with minimum group edit distance as the proposed method. We see that in the large and medium lexicon setting of IIIT 5K-word dataset, our method outperforms the existing ones. We also obtain similar performance as compared to the other CRF methods on small lexicons.

Method	IIIT 5K-word			ICDAR 03	SVT
	Large	Medium	Small	Small	Small
non-CRF based					
Wang et al. [3]	-	-	-	76.0	57.0
Bissacco et al. [2]	-	-	-	82.8	**90.3**
Alsharif et al. [22]	-	-	-	**93.1**	74.3
Goel et al. [5]	-	-	-	89.6	77.2
Rodriguez et al. [6]	-	57.4	**76.1**	-	-
CRF based					
Shi et al. [10]	-	-	-	87.4	73.5
Novikova et al. [9]	-	-	-	82.8	72.9
Mishra et al. [4]	-	-	-	81.7	73.2
Mishra et al. [7]	28.0	55.5	68.2	80.2	73.5
Our Method	**42.7**	**62.9**	71.6	85.5	76.4

in the lexicon for each word in the solution set. We then consider the union of all top-100 nearest lexicon words to be the new lexicon and perform the group edit distance based re-ranking on it. This speeds up the process by around 200 times and reduces the computation time to less than a second.

Discussion. Fig. 3 shows that lexicon reduction (and re-computation of priors using diverse solutions) corrects solutions in the first four iterations. We observe that the inferred label changes by one or more characters as the priors get stronger over iterations by assigning a lower pairwise cost to the bigrams from the ground truth.

Table 1 compares the performance of the proposed method with the state of the art over the three datasets. We see that our method outperforms the state of the art in the large lexicon setting. We obtain 14% improvement over [7] because of stronger priors.[1] As a baseline, to evaluate the effectiveness of the diversity constraint, we searched for multiple candidate words using the CRF energy without using the diversity constraint. For example, on the IIIT 5K-word dataset (with medium lexicon), this resulted in an accuracy of 55.6 without diversity compared to 62.9 (with diversity, shown in Table 1), when considering the top-5 candidate words.

---

[1] It should also be noted that [7] follows an open vocabulary lexicon, i.e., it does not assume that the ground truth is present in the lexicon. We find that around 75% of the ground truth words from the IIIT 5K-word dataset are present in the large lexicon by default. The rest of the ground truth words are language-specific and proper nouns like city and shop names.

Table 2. Top-1 precision for retrieval experiment on various datasets. We compare the results between two reduction methods, each with and without diverse solutions. The partial reduction method leaves some lexicons with around 5 words, while the full reduction method reduces all lexicons to size one. We see that our proposed method of partial reduction with diverse solutions works the best for the IIIT 5K-word dataset.

Method	IIIT 5K-word			ICDAR 03
	Large	Medium	Small	Small
Without diversity				
Full Reduction	27.5	51.9	65.0	**81.7**
Partial Reduction	35.1	35.6	60.7	76.9
With diversity				
Full Reduction	23.1	52.0	65.0	78.9
Partial Reduction	**42.1**	**59.0**	**66.5**	79.5

For the small lexicon setting, non-CRF methods, like beam search on a graph in [2] perform well on the SVT dataset because of training the classifiers with millions of character images. This is around ten times larger than the amount of training data we use, and is unavailable to the public. The structured SVM formulation [6] shows a good performance on the small lexicon of IIIT 5K-word but deteriorates as the lexicon size increases. This is due to the model being incapable of effectively minimizing the distance between the label and image features in the embedded space for larger lexicons.

4.5   Retrieval

In this experiment we retrieve a word image for a given query word from the dataset. The dataset comprises of a singleton or a reduced lexicon for each image which is used for the task as described in Section 3. As our proposed method, we preprocess the dataset by reducing the lexicon to singleton if the AED value θ at the $5th$ (last) iteration is less than 3.5. We call this process the *partial reduction* method as it reduces the lexicon to size one only for some word images, and for the rest, the lexicon contains 5 words. As a baseline method, we also do a *full reduction*, reducing lexicons for all the word images to one corresponding word. Both, the proposed and the baseline methods, are performed with and without the diversity constraint, thus creating four different variations. The parameter θ that gives the best precision for the proposed method is selected after cross validation over an independent query set. For quantitative evaluation, we compute the precision of the first retrieved word image as the datasets do not have a significant number of repeating ground truth labels (i.e., word images with the same text).

We show quantitative results in Table 2, where we clearly see that partial reduction of lexicons with diversity outperforms full reduction without diversity on the IIIT 5K-word dataset. The diverse solutions improve the performance as they retain the ground truth in reduced lexicon after the lexicon reduction process in many cases. We also notice that on IIIT 5K-word dataset, the performance gap

Query	Retrieved Image	Reduced Lexicon: diversity + partial red.	Reduced Lexicon: diversity + full red.
BRADY	Brady	MY, BRADY, ANY, A	MY
SPACE	SPACE	HOT, SPACE, LACEY, SALE	HOT
HAHN	HAHN	BUENA, HANDA, HAHN, PIPE	BUENA
DAILY	Daily	PEARL, MOUNTS, DAILY, NIKE	PEARL
TIMES	TIMES	TIME, TIMES, WINE, MED	TIME
THREE	three	THE, THREE, THERE, USED	THE

Fig. 4. Cases where retrieval results are correct. The reduced lexicon from partial reduction method (partial red.) retains the ground truth word. The words in the reduced lexicon are similar to each other, and any further reduction could have resulted in loss of ground truth.

increases as the lexicon size increases, suggesting potential applicability to larger lexicon based query systems. Correct retrievals (in Fig. 4) show that a higher AED threshold based lexicon association has the ground truth in the reduced lexicon associated with it, as compared to its singleton lexicon. The method is less successful in cases (Fig. 5) where the ground truth is lost in the early stages of lexicon reduction leading to a reduced lexicon without the ground truth in it. This happens due to failure of the binarization method used to segment out the characters, which leads to abrupt short/long candidate word formation.

	Query	Retrieved Image	Reduced Lexicon
1	CLEAR	SLEDS	CLEAR
2	HOME	9:00am	HOME, 900AM, 9080, 90
3	BAR	DALLAS	BAR
4	FOR	SUPER	AND, ARTS, FOR, INN
5	311		311
6	JOIN	one	ONE, JOIN, OUT, OUR

Fig. 5. Failure cases for retrieval experiment with reduced lexicons after partial reduction. Some word images have reduced lexicons with no ground truth (rows 1, 3, 4, 5). Other cases have the ground truth word, but are retrieved for the wrong query word (rows 2, 6).

5   Summary

In this paper we proposed a novel framework for recognition and retrieval tasks in the large lexicon setting. We identify potential character locations and find words contained in the image. We reduce the large lexicon to a small image-specific lexicon. The lexicon reduction process alternates between recomputing priors and refining the lexicon. We evaluated our results on public datasets and show superior performance on large and medium lexicons for recognition and retrieval tasks.

References

1. Weinman, J., Butler, Z., Knoll, D., Feild, J.: Toward Integrated Scene Text Reading. TPAMI (2014)
2. Bissacco, A., Cummins, M., Netzer, Y., Neven, H.: Photoocr: Reading text in uncontrolled conditions. In: ICCV. (2013)
3. Wang, K., Babenko, B., Belongie, S.: End-to-End Scene Text Recognition. In: ICCV. (2011)
4. Mishra, A., Alahari, K., Jawahar, C.V.: Top-down and bottom-up cues for scene text recognition. In: CVPR. (2012)
5. Goel, V., Mishra, A., Alahari, K., Jawahar, C.V.: Whole is Greater than Sum of Parts: Recognizing Scene Text Words. In: ICDAR. (2013)
6. Rodriguez, J., Perronnin, F.: Label embedding for text recognition. In: BMVC. (2013)
7. Mishra, A., Alahari, K., Jawahar, C.V.: Scene text recognition using higher order langauge priors. In: BMVC. (2012)
8. Weinman, J.J., Learned-Miller, E., Hanson, A.R.: Scene Text Recognition Using Similarity and a Lexicon with Sparse Belief Propagation. TPAMI (2009)
9. Novikova, T., Barinova, O., Kohli, P., Lempitsky, V.: Large-lexicon attribute-consistent text recognition in natural images. In: ECCV. (2012)
10. Shi, C., Wang, C., Xiao, B., Zhang, Y., Gao, S., Zhang, Z.: Scene Text Recognition Using Part-Based Tree-Structured Character Detection. In: CVPR. (2013)
11. Batra, D., Yadollahpour, P., Guzman-Rivera, A., Shakhnarovich, G.: Diverse m-best solutions in markov random fields. In: ECCV. (2012)
12. Sheshadri, K., Divvala, S.K.: Exemplar driven character recognition in the wild. In: BMVC. (2012)
13. Tian, S., Lu, S., Su, B., Tan, C.L.: Scene text recognition using co-occurrence of histogram of oriented gradients. In: ICDAR. (2013)
14. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM journal on computing (1972)
15. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. TPAMI (2006)
16. ICDAR 2003 datasets, http://algoval.essex.ac.uk/icdar.
17. Wang, K., Belongie, S.: Word Spotting in the Wild. In: ECCV. (2010)
18. Street View Text dataset, http://vision.ucsd.edu/∼kai/svt.
19. Otsu, N.: A Threshold Selection Method from Gray-level Histograms. IEEE Trans. Systems, Man, and Cybernetics (1979)
20. Mishra, A., Alahari, K., Jawahar, C.V.: Image retrieval using textual cues. In: ICCV. (2013)
21. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR. (2005)
22. Alsharif, O., Pineau, J.: End-to-end text recognition with hybrid HMM maxout models. arXiv preprint arXiv:1310.1811 (2013)