Pattern Recognition 108 (2020) 107542

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Bringing semantics into word image representation

Praveen Krishnan*, C.V. Jawahar

CVIT, Kohli Center on Intelligent Systems (KCIS), IIIT Hyderabad, INDIA

ARTICLE INFO

Article history:
Received 6 April 2019
Revised 14 March 2020
Accepted 11 July 2020
Available online 12 July 2020

Keywords:
Word image embedding
Word spotting
Semantic spotting

ABSTRACT

The shift from one-hot to distributed representation, popularly referred to as word embedding has changed the landscape of natural language processing (NLP) and information retrieval (IR) communities. In the domain of document images, we have always appreciated the need for learning a holistic word image representation which is popularly used for the task of word spotting. The representations proposed for word spotting is different from word embedding in text since the later captures the semantic aspects of the word which is a crucial ingredient to numerous NLP and IR tasks. In this work, we attempt to encode the notion of semantics into word image representation by bringing the advancements from the textual domain. We propose two novel forms of representations where the first form is designed to be inflection invariant by focusing on the approximate linguistic root of the word, while the second form is built along the lines of recent textual word embedding techniques such as Word2Vec. We observe that such representations are useful for both traditional word spotting and also enrich the search results by accounting the semantic nature of the task. We conduct our experiments on the challenging document images taken from historical-modern collections, handwritten-printed domains, and Latin-Indic scripts. For the purpose of semantic evaluation, we have prepared a large synthetic word image dataset and report interesting results for the standard semantic evaluation metrics such as word analogy and word similarity.

1. Introduction

What's in a word? According to the linguist Ferdinand de Saussure [1], a word is like a coin which has two sides to it that can never be separated. On one side, we have the *form* of the word, composed of sounds and letters that combine to make a spoken or written word. While on the other side we have the *meaning* which gives us the concept or an intuition on the word usage. In the domain of document images, for decades we have appreciated the need for learning a holistic word level representation which is popularly used for the task of word spotting. However, most of the previous works in this space have restricted the representations which only respect the word form, while ignoring its meaning. In this work, we attempt to bridge this gap by encoding the notion of semantics by introducing novel forms of word image representations.

Learning representation from the data forms the basis of any pattern recognition problem. With a good representation, one achieves both better performance and insights into the underlying problem. In the text domain, the shift from one-hot to distributed representations learned using techniques such as Word2Vec [2], and GloVe [3] have changed the entire landscape for information retrieval (IR) and natural language processing (NLP) problems. Such distributed representations, popularly referred as word embeddings, essentially embed the relationships among words by formulating a proxy task such as language modeling. The embedding process projects each word into a vectorial space where the distances among the words define its semantic relationships. The learning process exploits the cue that: *the words which are related in semantics keep the company (context) of similar words in a document*. The widespread use of word embeddings as the underlying representation brought a revolution in the NLP domain. In this work, we would like to capitalize this success to document images where the community largely restricts itself to image representations which captures only the visual properties. Fig. 1(a) presents a couple of word image pairs and their similarity values projected along visual and semantic axes. As mentioned before, the traditional word image representations captures only the visual properties (top-axis), while in our proposed work, we would like to additionally encode the semantic properties (bottom-axis) of the word which further enriches the representation.

One of the interesting aspects in the domain of document images is its close relationship with the text domain. A document image essentially contains text, and hence the "semantics" that are

* Corresponding author.
E-mail address: praveen.krishnan@research.iiit.ac.in (P. Krishnan).

https://doi.org/10.1016/j.patcog.2020.107542
0031-3203/

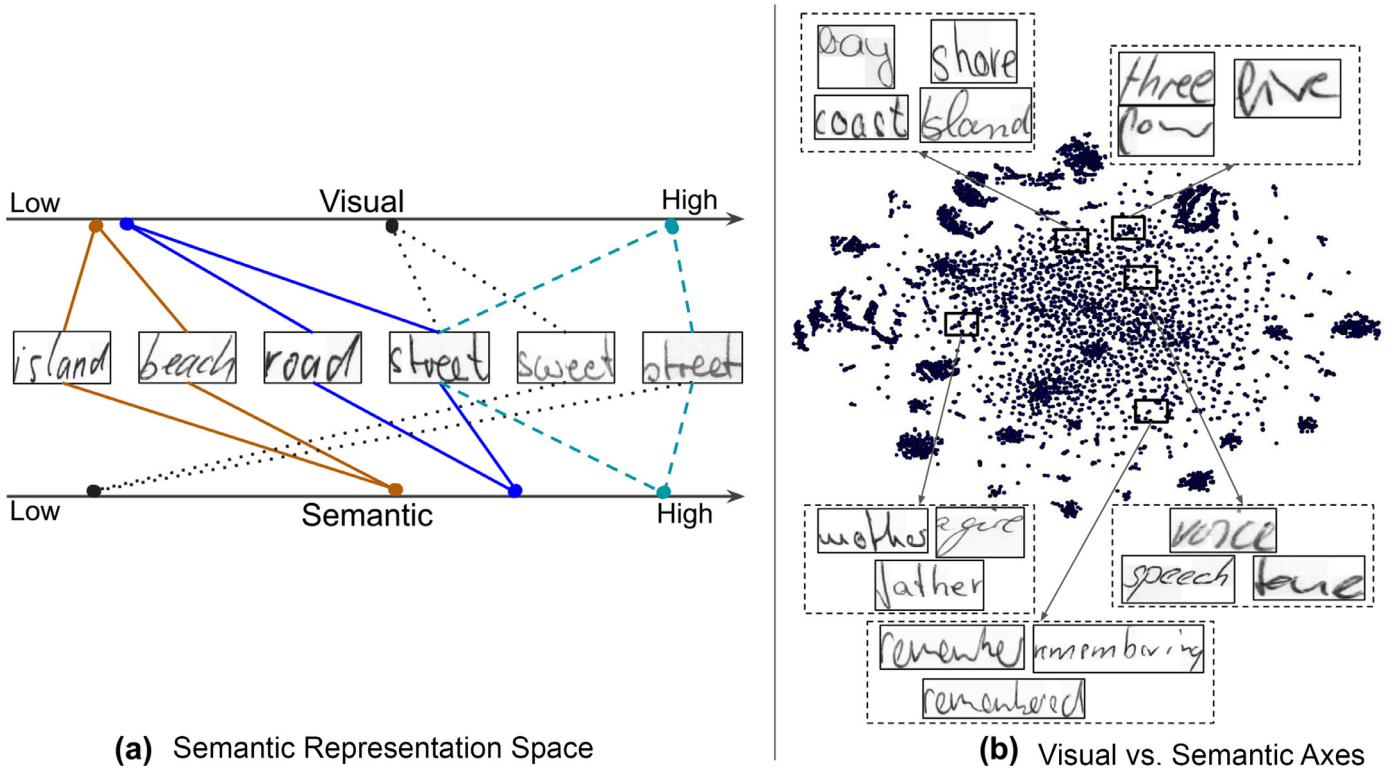

Fig. 1. (a) Projection of word similarity scores onto visual and semantic axes as shown in top and bottom locations respectively. Here the top axis uses word image representation [4] which only focus on the visual space, while the bottom axis uses the proposed semantic representation which focus on the meaning of the word. (b) Visualization of the proposed semantic representation space using t-SNE.

involved in creation of the original text are seamlessly transferable to it. In linguistics, "semantics" refers to a broad area which investigates the meaning and relationship among words, phrases, sentences and documents. In this work, we restrict the scope to individual words (word images) which can be treated as a fundamental and meaningful unit for a document. We explore two major relationships among words:- (i) at the level of individual word usages caused by "inflections", which expresses different grammatical categories such as tense, case, number, etc., and (ii) semantic relatedness among words which associates words with similar meaning in a given context. We address both these problems by bringing external linguistic knowledge from textual domain in the form of algorithms (stemming, lemmatization, etc.) and learned statistical models (such as Word2Vec). Although most of our discussions are for English, we will also present results for Indic scripts which demonstrate the generic nature of the proposed framework. Note that, in this work, we limit our discussions to segmentation based settings where we obtain word segmentation information in the form of ground truth. Fig. 1(b) presents a snapshot of the proposed word image semantic representation space visualized using the t-SNE algorithm. Here the neighboring word images belong to semantically related entities and are invariant of handwriting styles.

1.1. Recognition vs. representation

Broadly, there are two paradigms of solution in the direction of our problem: (i) *recognition* (or transcription) of a word image into its constituent textual string and later transferring it into the semantic space using existing textual IR algorithms, and (ii) a direct embedding of word images into the semantic representation space through a learned model.

A perfect recognition is all we want for building any type of IR or NLP application around document images. However, the document images that we are interested in this work are handwritten, historical manuscripts and degraded printed books where traditional printed Optical Character Recognizer (OCR) based methods would result in noisy text and could lead to inferior results. This leads us to the complementary method, where the idea is to formulate the problem from a retrieval perspective. In this case, the focus is to learn an appropriate *representation* space which preserves the similarity among the content words irrespective of their variations due to style and degradation. One of the classical works in this space is the development of word spotting problem, originally proposed in the speech community [5] and later introduced in the document community [6]. Here, the basic idea is to represent the underlying word images into a representation space where matching between the query and the candidate is posed as a retrieval problem. In literature, there exist many successful representations which are learned either in an unsupervised setting or in a supervised fashion. These representations are successful in capturing the lexical similarity at the level of word-forms, while remaining invariant to a large extent to different (writers) styles and degradations. More details on these methods are presented in the related works under Section 2.1.

1.2. Why semantic representation?

Our motivation to learn a semantic representation for word images is two-folded. The first and foremost reason is that, this enables the user to retrieve words based on the meaning of the query. The second important motivation is driven by the fact that these representations could open newer research directions where one could attempt to solve certain NLP problems directly on document images. This draws parallels to how textual embedding techniques became the de facto representation scheme in NLP tasks such as machine translation, sentiment analysis, POS tagging,

Fig. 2. The top row shows the nearest neighbors of a query "watched" in the traditional word spotting setting, while the middle row shows the word images that are related to query in terms of inflections or word morphology, while the bottom row shows the nearest neighbors which are either related in terms of inflections, or synonyms, or related in terms of semantics.

named entity recognition (NER) and also to certain multi-modal problems such as image captioning and visual question answering. In similar spirits, in the domain of document images, there has been a recent interest [7,8] in solving some of these NLP tasks. Some of the attempted problems are finding document similarity [9], Named Entity Recognition (NER) [7,8], automatic essay grading [10], etc.

1.3. Traditional spotting vs. semantic spotting

A traditional word spotting method respects the similarity among the word images on the basis of their *form* by embedding them into an approximate verbatim space. Note that, in this work we refer to the traditional word representation space as "verbatim" as coined in the work of Wilkinson and Brun [11] in order to differentiate it with semantic space. For example, in the verbatim space the representation for word images written/printed in different styles or handwriting are supposed to lie close to each other. The top row of the Fig. 2 shows the traditional spotting system for the example query "watched" along with its top nearest neighbors. In this work, we propose two novel form of representations which project the word images into a semantic space where distances among the representations define the semantic relationships. We refer this task as semantic spotting as shown in the middle and bottom rows of Fig. 2. Here, the middle row presents the top nearest neighbors for the same query, which includes other related word images that are invariant to word form inflections. We refer this representation as *normalized word embedding*. We further generalize this to include broader semantics along the lines of textual word embedding techniques such as Word2Vec, and GloVe. The bottom row of the Fig. 2 shows the result from the proposed *semantic representation*.

1.4. Contributions

The underlying architectures used in this work are first presented in our previous works [4,12] and were primarily used for word spotting. These are summarized in Section 3 and 4 respectively for completeness. The primary focus of this work is to bring semantic into the learned representations. More specifically, the major contributions of this work are: (i) We propose a normalized word representation which is invariant to word form inflections. The representation achieves state of the art performance as compared to a conventional verbatim based representation, (ii) We introduce a novel semantic representation for word images which respects both its *form* and *meaning*, thereby reducing the vocabulary gap that exists between the query and its retrieved results, (iii) We demonstrate semantic word spotting task and evaluate the proposed representation on standard IR measures such word analogy and word similarity, and (iv) The proposed representation is successfully demonstrated on both historical and modern document collections in printed and handwritten domains across Latin and Indic scripts.

2. Related works

We broadly categorize our literature review into three major subsections. We first explore the prominent methods in the domain of word image representations. We then discuss the existing literature of semantic embedding from the IR or NLP communities where the problem is more fundamental and has numerous applications. Finally, we discuss some of the recent papers in the domain of document images that address the problem of learning semantic embeddings from word images.

2.1. Word image representation

Learning holistic representation for word images was popularized with the problem of word spotting [6]. Given a corpus of word images, segmentation based word spotting involves retrieving all the relevant instances of a given query from the corpus. Most of the traditional methods only consider the lexical correctness of the word while performing the retrieval. Initial methods in this space used variable length representation schemes [13], to exploit the temporal information in words. These methods were mostly based on profile features [6,14] which essentially summarize the pixel level statistics. However these features are less robust to wide variations in style and demand robust pre-processing stages such as binarization and noise removal. With the popularization of bag of words (BOW) [15] method in the vision community, such frameworks were adapted for learning holistic representations for word images [16–18]. These methods were based on encoding local patch level descriptors such as SIFT [19] using a visual code book learned in an unsupervised fashion. Given an image with a set of encoded code words, pooling is performed to compute a fixed-length vectorial representation. In [16], the representation is further projected onto a topic space using latent semantic indexing (LSI) [20], where the latent topic space is assumed to preserve the lexical content of word images. Although the authors use a semantic model (LSI) in their work, the purpose was to exploit the semantic relatedness in visual word space, thereby improving the quality of holistic representation for lexical matching. For further details on BOW based word representation schemes for document images, readers can refer to [18].

More recently, learning representation in a supervised setting has gained a lot of interest. One of the prominent methods in this space is [21] which uses the concept of word attributes using a representation called pyramidal histogram of characters (PHOC). A PHOC representation is built by concatenating the histogram of characters at multiple spatial regions in a pyramidal fashion. Unlike previous methods (e.g., gradient features, BOW, etc.) which do not use supervised learning, these attributes are learned from images in a supervised setting.

With the advancements in deep learning and its wide success in feature learning for natural images, numerous methods have been proposed which use deep features for learning word image representations [9,11,12,22–24]. In our earlier works [9,12], we proposed a deep convolutional network (HWNet) for word image rep-

resentation learning by formulating a proxy task of word classification. Here, the holistic features are taken from the penultimate layer of the network. We further extend our architecture in [4,25] to embed word images and textual representations onto a common subspace which enable both query-by-string and query-by-image word spotting. Along the lines of PHOC attributes, Poznanski and Wolf [22] adapted VGGNet [26] for recognizing these attributes by having multiple parallel fully connected layers where each one predicts a PHOC attribute at a particular level. One of the prominent works in this space is from Sudholt and Fink [23] which proposes an architecture to directly embed image features to PHOC attributes by having sigmoid activation in the final layer. It is referred as PHOCNet, which uses the final layer activations to derive a holistic representation for word spotting. In continuation, the authors adapt PHOCNet with a temporal pooling layer (TPP-PHOCNet) [24,27] which enriches the temporal features by preserving information at multiple regions (similar to spatial pyramids) from a word image. The above methods learn representations which respect lexical similarity, thereby ignoring the semantics of these words. In this work, we propose novel form of semantic representations for word images which provides a platform for semantic word spotting.

2.2. Textual word embedding

Learning continuous representation for textual words (word embedding) which respects their semantic properties has been a fundamental quest in the text processing community. Such word embeddings are useful in many downstream NLP and IR applications such as text classification [28], named entity recognition [29], etc. The term word embedding was coined in the classical work of Bengio et al. [30] where the problem is formulated in terms of language modeling. Here, the underlying task is to learn a probability distribution of word sequences present in a context. This implicitly learns a distributed representation of words. The task is achieved using a neural architecture with a cross entropy based loss function, however the complexity in the output softmax layer, which is equivalent to the size of vocabulary, presented the major computational bottleneck. One of the most popular and cited works in the field of word embedding is by Mikolov et al. and is referred as Word2Vec [2,31]. Word2Vec uses a shallow neural architecture and a negative sampling loss function which is computationally far efficient than softmax. Another popular formulation referred as GloVe proposed by Pennington et al. [3] uses word-to-word co-occurrence statistics from the corpus to learn word representations. Here, the basic idea is to exploit the semantic information that is the present in the ratio of word co-occurrence probabilities. Both Word2Vec and GloVe learn high quality word representations which can be readily used in many downstream NLP tasks. However, one of the major limitation in the above formulations is that one cannot extract the word embedding for an out of vocabulary (oov) word. This is because the underlying formulation treats the individual unit as a word and hence, the word embeddings are only learned for the words in training vocabulary. To avoid such limitations, recent methods [32,33] were proposed at sub-word level while keeping the overall formulation similar. Such models can now also define representations for oov words, and are also quite useful for morphologically rich languages.

2.3. Word image semantic embedding

In the document image community, there has been only very few pursuits in learning the semantic aspects of the word images. One of our initial work [34] highlights the need for semantics, and demonstrated a two-stage approach for performing semantic retrieval. The basic idea is to perform a query expansion by matching the query to an annotated corpus of word images (semantic index) which contains the associated relationships among words defined by WordNet [35]. The work could not be considered as semantic embedding since we didn't embed word images into a semantic space, however the final goal for semantic retrieval remains similar to this work. In [36], the authors proposed an end2end representation learning using a chosen subset of concepts from WordNet using a deep convolutional architecture. Here, each word image is annotated with a semantic attribute vector generated from the WordNet and the network is trained using a weighted ranking loss function. The work demonstrated the results on synthetic scene text dataset in both query-by-concept and query-by-image formulations. Both the above mentioned works [34,36] utilize the lexical database WordNet which is limited to the available human annotations, supports very few languages and defines only specific semantic relationships. However, the recent textual embedding techniques such as Word2Vec or GloVe have the ability to learn both generic and specific semantic relationships in an unsupervised fashion. The recent work from Wilkinson and Brun [11] is one of the first work which uses these textual embeddings to learn word image representation. The authors propose a two-stage neural architecture which uses a cosine embedding ranking loss to project word images into the semantic space. However, [11] only analyzes the performance of the traditional word spotting task and does not study the semantic aspects of such a representation. We draw inspiration from this work and systematically present a framework which can learn a better semantic representation, and also evaluate it under both semantic tasks and the conventional word spotting task.

3. HWNet: word image representation

Given our interest in bringing semantics into word image representation, one of the foremost design challenge is to come up with an architecture, which can embed word images into an appropriate feature space. In this work, we use our previous proposed architecture referred as HWNet v2 [12] for learning representation which is given to the embedding architecture (originally proposed in [4]) which brings both text and images into a common subspace. In this section we present the HWNet style of representation, while in the next section we present the joint image and text embedding framework. In both these sections, we limit our discussion to the lexical properties of an image, while in Section 2.3, we adapt these architectures for the purpose of semantic embedding. Note that for simplicity, we would refer HWNet v2 as HWNet in this paper.

One of the key observation from a trained deep CNN network using a large dataset (e.g. AlexNet [37] trained on ImageNet corpus) is that the feature maps obtained at the intermediate layers are generic in the nature. The design of HWNet architecture [12] follows the inspiration from this recent form of feature learning scheme. However, one of the major challenges to be solved first is the need for a huge annotated corpus in our domain which could substitute the role ImageNet played for vision tasks. Fortunately there exists an unique solution for generating huge synthetic dataset from open source fonts which could practically provide a large annotated corpus of word images for free. IIIT-HWS dataset[1] is a collection of synthetic word images rendered from handwritten style fonts with suitable post-processing for adding enough realism to the generated dataset. In this work, we would use this dataset for pre-training our networks in order to obtain a better initialization of weights.

The top part of the Fig. 3 shows the HWNet architecture. It uses a ResNet34 architecture along with spatial pyramid pooling.

[1] http://cvit.iiit.ac.in/research/projects/cvit-projects/hwnet.

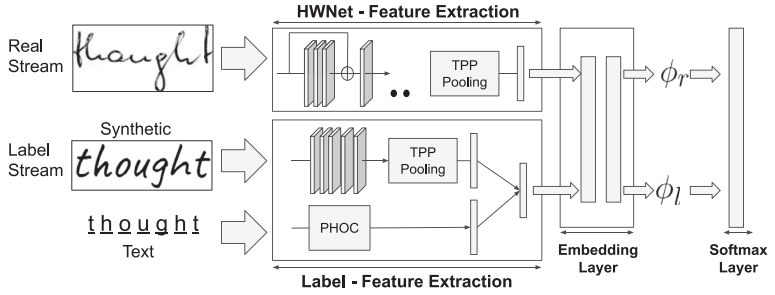

Fig. 3. Figure depicts two streams in the End2End word image embedding architecture, where the real stream inputs the handwritten/printed word image through the HWNet network for feature extraction. The label stream contains two parts which takes a synthetic image and a PHOC representation for the corresponding textual label. Both the streams converge to a common representation space shown as embedding layer which is implemented using a multi-layer perceptron. The layer contains softmax units which computes the class probabilities for each word in the vocabulary.

Instead of using global average pooling after convolutional layers (as used in the original ResNet architecture [38]), we found fully connected (FC) layers learn better features at the penultimate layer. In order to support variable length word images as input, which also preserve the aspect ratio of the input images, we use a spatial pyramid pooling layer (SPP) after the final convolutional layer. SPP is a multi-scale generalization for region of interest pooling (ROI). SPP takes a variable length representation as input and produces a fixed length output depending on number of grids. In our usage of SPP along HWNet, we create pyramid levels only vertically with levels set at 1, 2 and 3 similar to the one used in [24]. This would essentially capture the temporal properties present in the word image. This is referred as temporal pooling layer TPP. For much detailed analysis on HWNet architecture and the training process, the readers are advised to refer to [12].

4. Joint image and text embedding

The idea for joint image and text embedding, is to learn a common subspace for both word images and the corresponding textual strings. Fig. 3, presents our architecture which was first proposed in [4], where we achieve the joint embedding using a two stream (real and label) network with a multi-task loss function. In this work, we refer this as End2End embedding architecture. The real stream is essentially the HWNet network which takes real word images as input, while the label stream incorporates the supervised label information. The label stream is further split into two modalities, which inputs the synthetic image rendered in one single font and the textual stream which is fed using PHOC [21] representation. The features of both real images and synthetic images are captured using convolutional layers, where we use HWNet architecture for real images and a shallower CNN network similar to HWNet for the synthetic image stream. These choices are motivated by the complexity level of data variations in individual streams. In both the CNN networks (real and synth), we use TPP pooling after the last convolutional layer to operate on variable length images. The textual stream, where the inputs are fed using the PHOC extractor, is appended to the synthetic stream and is treated as a conditional label. This is achieved by concatenating the flattened activations from the synthetic network with the PHOC based textual representation. After concatenation, we use a fully connected network to merge both information. We believe this preserves the complementary information from both the modalities (synth and text). Note that the weights of individual streams as part of the convolutional layers are not shared because:- (i) we need both real and synthetic (along with its text) streams to learn complementary features, and (ii) to learn real stream without any conditional label information. Given the two sets of features, one from the real stream and another from the label stream, we perform label embedding by projecting both these features into a common subspace. We achieve this using an embedding layer as shown in the Fig. 3 (right-most block), which is a typical Siamese style network implemented as a multi-layer perceptron. Here, the weights are shared since we want to identify the common subspace where the correlation among similarly paired data is maximized. Finally, the embedding features are given to the softmax layer to compute the class-wise probability of each word in the training vocabulary. To train the network, we use a multi-task loss function as given below:-

$$\mathcal{L}(\phi_r, \phi_l, y) = \mathcal{L}_1(\phi_r, y) + \mathcal{L}_2(\phi_l, y) + \mathcal{L}_3(\phi_r, \phi_l) \tag{1}$$

Here, ϕ_r, ϕ_l are the embeddings obtained from the real and label streams respectively as shown in the Fig. 3, while y is the ground truth label represented using one hot representation. The first two components $(\mathcal{L}_1, \mathcal{L}_2)$ of the loss function are cross entropy based classification loss functions computed on the soft-max scores for real and label embeddings respectively. The third component $(\mathcal{L}_3)$ is a similarity loss function, which is defined using the cosine similarity between the pairs of features belonging to the same label, and is given as:-

$$\mathcal{L}_3(\phi_r, \phi_l) = 1 - \cos(\phi_r, \phi_l) \tag{2}$$

The choice of multi-task loss was done following our experience with training HWNet [12], which convinced us that the features learned while training a word classification network are robust enough to perform word spotting. More details on the training strategy such as pre-training, data augmentation and architectural details will be discussed in Section 6.3 as part of implementation details. Given the trained embedding network, in order to compute embedding for the test images and strings, we extract the L_2 normalized activation from the penultimate layer of the network, which in our case are ϕ_r and ϕ_l respectively. As mentioned earlier, the original motivation of this network and learning procedure is to learn a generic representation for word images and text, suitable for spotting lexically similar words with respect to query. In next section, we adapt our architecture and the loss functions to induce semantics into the representation.

5. Word image semantic embedding

In this work, we propose a word image representation which respects both the lexical and the semantic properties of a word. While the definition of semantics is quite broad, in this work we restrict our scope in two forms:- (i) words which are related to each other in the form of “common linguistic root”, and (ii) the words which are related to each other in terms of “common context”. In the following section, for each form, we propose representations which respect these semantic relationships.

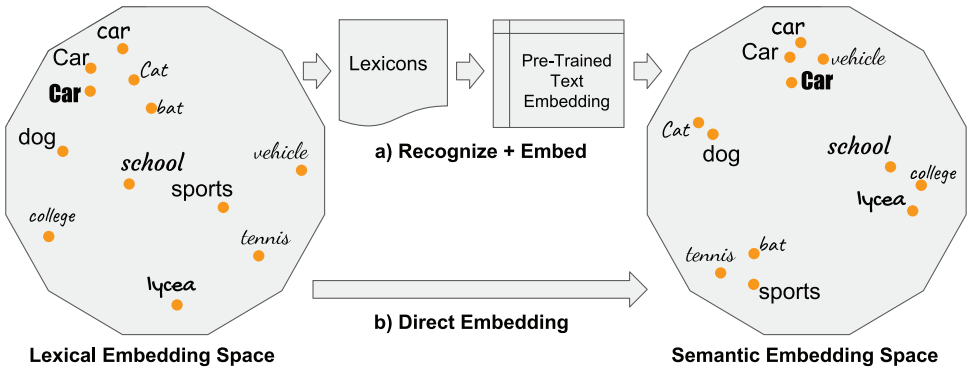

Fig. 4. Intuition behind the semantic embedding process. Here each point refers to a word image embedding either shown in the lexical space (shown in left) or the semantic space (shown in right). We propose two schemes to embed word images from lexical to semantic space (a) Recognize+Embed, and (b) Direct Embedding.

5.1. Normalized word embedding

In textual search systems, one of the first pre-processing stage is word normalization, which includes normalizing the case of the word to lower case and reducing each word to its root/stem form. For example, words such as *look, looking, looked, looks* will all reduce to its root form *look*. In terms of linguistic rules, these individual words convey different grammatical categories, however from a search perspective they all convey the same information. We follow a similar paradigm where we would like to learn normalized word embeddings for word images which are invariant to such word forms in a language.

In the English language, we observe variations in the form of inflections to a word which expresses different grammatical categories. These are expressed in the form of either prefix or a suffix (more generally called as affixes). The popular inflectional ending in English are "-s" (plural), "-ed" (past tense), "-ess" (adjective), "-ing" (continuous form), etc. These affixes are added to the root word, thereby resulting in a semantically related word. In text domain, there are two ways to normalize such words to their root forms:- (i) use of stemming algorithms which use heuristics to reduce the word to a stem which is very close to the actual root form, and (ii) use of lemmatizers or morphological analyzers which are more of linguistic rule based systems which use the parts of speech (POS) information to deduce the root for the word.

In this work, we imitate the process of stemming and lemmatization in visual domain using our image embedding network by formulating the problem of normalized word embedding. Here we use standard and popular linguistic tools such as Porter stemmer [39] and WordNet based Lemmatizer [40] which are provided as part of the NLTK toolkit [41] to strip out common affixes. This generates a normalized representation of words with common roots. Here, the Porter stemmer uses multiple heuristic rules in stages to strip the affixes, while the lemmatization algorithm uses the knowledge of WordNet [35], a large lexical database of English annotated by humans, along with the POS tag information of the word to derive the root form of the word. Note that POS computation for isolated words could be ambiguous without the context information. Given the reduced number of word classes obtained from stemmer and the lemmatizer, we train our embedding network (as presented in Section 4) with the modified classification losses $(\mathcal{L}_1, \mathcal{L}_2)$ which now computes the loss in recognizing the root word from the word image. Note that the $\mathcal{L}_3$ loss remains the same as mentioned earlier. We empirically observe that such a trained network gives lesser weights to popular word suffixes and learns a feature space where both the root of the word and its inflections lie close to each other.

5.2. Semantic embedding

We now generalize our notion of semantic representation by taking inspiration from the word embedding techniques proposed in the text/NLP community, such as Word2Vec [2] and GloVe [3] algorithms. The distributed representations learned using these word embedding techniques establish relationships among words which occur in a similar context in a language. In this work, we use a pre-trained textual word embedding model and project word image features into the learned textual embedding space with no loss of generality in representation space. We achieve this transfer in two ways: a) recognize+embed, and b) direct embedding. This is visually demonstrated in Fig. 4. In both schemes the target semantic embedding space remains the same.

5.2.1. Lexicon based recognition and embedding

We consider lexicon-based recognition and embedding scheme as a two-stage semantic embedding process, where first we recognize the word image into a string $w \in D$, and then transfer it into a continuous representation space $w_s = F_w$ defined by a pre-trained textual semantic look-up table such as Word2Vec. Here D is the list of lexicon words from a dictionary, F is the semantic look-up table, and F_w is the representation for word w.

We achieve the first task using our lexical word-image embedding architecture as presented in Section 4. We project each textual word in the dictionary D into the lexical embedding space through the label stream of the architecture which utilizes both the synthetic rendering of the textual word and its PHOC representation. In order to recognize a handwritten word image, we perform a k-nearest neighbor operation with $k = 1$ in the lexical embedding space after projecting each word image through the real stream. Once we have mapped a word image to a textual label w, we use a pre-trained semantic look-up table to transfer w to w_s which lies in the semantic space.

There exists a limitation in this scheme which enables only in-vocabulary semantic transfers due to the usage of a pre-defined list of words in the dictionary D. However, as discussed in the experiments section, we observe that the proposed two-stage scheme provides a powerful representation which enables an efficient semantic spotting in the domain where we have a fair knowledge on target lexicons. We also perform an ablation study to validate the robustness of such a scheme under varying sizes of lexicon. The idea of recognition followed by embedding is generic and the performance is directly proportional to the recognition performance. Also, one can substitute an unconstrained recognition system instead of the constrained one as used in this work.

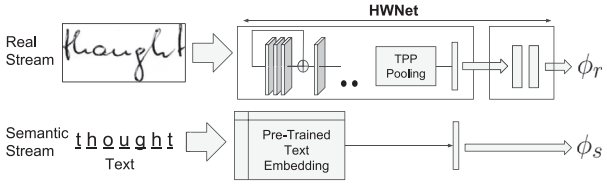

Fig. 5. Direct word image embedding architecture. Here, we use a semantic stream which provides the textual word embedding for the current word image being passed through the real stream.

5.2.2. Direct embedding

One of the limitations of the previous scheme is that if the recognition is wrong, we transfer the wrong semantics which is non-recoverable in the embedding space. In this alternative scheme, we propose to directly embed into the target semantic space. To enable such a formulation, we adapt our embedding network as shown in Fig. 5 by incorporating a semantic stream instead of the label stream. In the real stream, the final fully connected layer has now a dimension equivalent to the textual word embedding size (typically $\sim 100 - 300$).

To train the network, we use two loss function variants. The first one is the mean square error (MSE) loss as given in Eq. 3 which minimizes image embedding with respect to the pre-trained textual embedding.

$$\mathcal{L}_{mse}(\phi_r, \phi_s) = \| \phi_r - \phi_s \|_2^2 \tag{3}$$

Here ϕ_r and ϕ_s refer to the word image embedding and its pre-trained textual word embedding obtained from real and semantic streams respectively. The MSE loss function only considers the distance between the target word image representation and its corresponding semantic feature while completely ignoring its difference from the negative samples which could lead to inferior ranking. Therefore we update the loss function to include this property which could potentially alleviate the problem by simultaneously minimizing the distance between both the positive samples and maximizing the distance between the negative ones. This is done through a cross domain triplet ranking loss as given below:

$$\mathcal{L}_{rank}(\phi_r, \phi_{s_p}, \phi_{s_n}) = \max(d(\phi_r, \phi_{s_p}) - d(\phi_{s_p}, \phi_{s_n}) + \alpha, 0) \tag{4}$$

Here, ϕ_{s_p} refers to textual word embedding for current word image whose representation is ϕ_r, ϕ_{s_n} refers to the embedding corresponding to a negative sample, and α is the margin (typically $\sim 0 - 1$). In the above equation, $d(.,.)$ refers to the L_2 distance between the two embeddings. Here the embedding variables, $\phi(.)$ are normalized with details given in the implementation Section 6.3. We train the network using similar training strategies [2] such as sub-sampling the words while forming the triplets so that commonly occurring words are sampled fewer times and do not bias the training. Similarly, while choosing negative samples, we sample the words from the distribution $w \sim P_n(w)$, where $P_n(w)$ is computed using unigram frequency of the word. The above triplet based ranking loss enables capturing the relationships among samples in the embedding much better as compared to the MSE loss.

6. Experiments

In this section, we first present our results on handwritten datasets which are popular in the community. In our work, we also use a synthetic dataset in order to evaluate the models in a controlled fashion thereby allowing to simulate evaluation tasks used in text embedding. We later present our results on printed datasets from Indic scripts.

6.1. Datasets

The IAM Handwriting Database [42]: It includes contributions from 657 writers making a total of 1539 handwritten pages comprising of 115,320 words. The database is labeled at the sentence, line and word levels. We use the official partition for writer independent text line recognition that splits the pages into training, validation and test sets which are writer independent.

George Washington (GW) [13]: It contains 20 pages of historical letters written by George Washington and his associates in 1755. The images are annotated at the word level and contain approximately 5K words. Since there is no official partition, we use a random set (similar to [21]) of 75% for training and validation and the remaining 25% for testing.

HW-SYNTH [43]: We also use an in-house synthetic word image dataset rendered from open source handwritten fonts as proposed in [43]. We use a vocabulary of 12K most frequent words in the English language. For each word in the vocabulary, we render 50-4 train-test word images which are sampled from nearly 710 fonts. The train and test data come from mutually exclusive font sets. This gives us nearly 0.6M word images.

6.2. Evaluation measures

In order to evaluate performance under word spotting tasks, we use the standard information retrieval evaluation measure, mean Average Precision (mAP) [44]. The selection of queries for each dataset follows the protocol used in [21], where we filter the stopwords from the test corpus while all words (including stopwords as distractors) are kept in the retrieval dataset in which the search is performed. We perform evaluation under both query-by-example (QBE) and query-by-string (QBS) setting. In QBE setting, since the query image is taken from the corpus, the first retrieved image is not included in the mAP calculation. While in QBS setting, the queries in test set are the unique strings (words). Also, note that all evaluations are done in a case-insensitive manner.

For evaluating the learned semantic representation, we follow the standard measures used in the IR community which are: (i) word similarity and (ii) word analogy. In word similarity measure, there exists human similarity judgements on a pre-defined set of word pairs as part of the dataset. For English, we use the popular dataset WS353 [45] which contains word pairs along with human similarity judgements, where 0 means totally unrelated words and 10 is very much related or identical words. We compute Spearman's rank correlation coefficient [46] between the human judgement score and the cosine similarity score between the pair of word image representation. Since the words in the WS353 dataset are specific ones which may not be present completely in the handwritten image datasets, we evaluate this measure only on the synthetic dataset where we have the control to render the appropriate word images.

For word analogy task, we use dataset introduced in [31]. The dataset contains questions of the form *A* is to *B* as *C* is to *D*, where *D* must be predicted by the model. We evaluate this task on HW-SYNTH and IAM datasets. Note that the questions which contain words which are not present in the test image corpus are excluded from the evaluation. Here we report mean accuracy of correct prediction on the questions.

6.3. Implementation details

We train our network and its modified version for semantic embedding using stochastic gradient descent algorithm with momentum. We set the momentum factor to be 0.9 and the learning rate is set as 0.01 while training from scratch on synthetic data. While performing fine tuning, the learning rate is initialized

Table 1
Quantitative results of normalized word spotting evaluated under mean Average Precision (mAP) and reported in percentages. Here we evaluate the learned embeddings in three different scenarios (Exact, Stem and Lemma).

Concept GT.	Evaluation	HW-SYNTH		IAM		GW	
		QBE	QBS	QBE	QBS	QBE	QBS
Exact	Exact	**98.49**	**98.96**	**92.54**	**96.54**	**99.37**	**99.46**
	Stem	74.62	88.31	86.35	90.79	97.46	98.61
	Lemma	81.54	91.32	86.11	92.09	97.60	98.69
Stem	Exact	87.70	87.31	90.65	92.15	98.14	96.31
	Stem	**97.56**	**98.73**	**92.05**	**95.69**	**99.45**	**99.12**
	Lemma	93.94	95.12	90.19	94.40	99.07	98.82
Lemma	Exact	93.18	92.70	91.08	93.34	98.14	96.89
	Stem	90.88	95.85	90.76	93.95	98.97	98.13
	Lemma	**97.56**	**98.62**	**91.67**	**95.01**	**99.01**	**98.20**
Exact	PHOC-Exact [23]	-	-	72.51	82.97	-	-
	PHOC-Stem	-	-	70.37	34.05	-	-
	PHOC-Lemma	-	-	68.18	60.22	-	-

from 0.001 and reduced by a factor of 2 once the loss does not change within a certain threshold in last two epochs. The weights are initialized using He initialization [47]. We perform extensive data augmentation [12] while training the network which includes elastic distortion, and affine transformations (scaling, translation, rotation and shear). The augmentations are done on-the-fly with 50% probability whether to augment the current sample from the mini-batch. For elastic distortion, we set the hyper-parameters $\alpha = 0.8$ and $\sigma = 0.08$, denoted as scaling and smoothing parameters [48] respectively, which regulate the amount of distortion. For affine transformation, we randomly pick whether to rotate, shear or pad. The rotation and shear angles are sampled in the range of $(-5, 15)$ and $(-0.5, 0.5)$ degrees respectively. For bringing translation in-variance, we randomly insert padding in the four boundaries within a range of 0–20 pixels.

As mentioned earlier, for the purpose of embedding word images into semantic space, we used fastText pre-trained models [49] for English, Hindi and Telugu. Given the textual embedding from fastText, we further perform normalization of the features by subtracting the mean and dividing by the standard deviation as computed from the training corpus. We found this to help in better training of the network and the same values are used while computing the test set features. For using the synthetic dataset, pre-trained network files along with codes for extracting features, please visit the project page[2].

6.4. Normalized spotting

In normalized word spotting task, we evaluate our representation presented in Section 5.1 which is invariant to word form variations due to inflections. Here we validate whether the representation essentially captures the root form of word. Table 1 presents the quantitative results evaluated with the proposed representation under the word spotting task using our image embedding network, and also compares it with PHOCNet representation introduced in [23]. Although PHOCNet is not primarily trained for capturing the linguistic root of the word, we consider this evaluation to verify whether this happens implicitly due to the sharing of attributes in the final layer of PHOCNet. As discussed in the Section 5.1, we have utilized two major linguistic knowledge sources, stemming and lemmatization to learn our embedding. We trained our embedding in three different scenarios (Concept GT): “exact”, “stem” and “lemma”. Here “exact” corresponds to the actual ground truth of the word taken for training, while “stem” and “lemma” corresponds to the approximate linguistic root of the given word. In either case, the training of the network is formulated as a classification problem while feature/representation is taken from the penultimate layer of the network after performing L_2 normalization. The evaluation criteria for the representation models trained under each of three Concept GT scenarios is also designed in the similar fashion. Here, we evaluate each model under all three scenarios. For example, the evaluation criteria of ‘Stem’ under the Concept GT model ‘Lemma’ would be the setting where the model is trained on lemma of the word obtained using the lemmatizer, and during evaluation we consider true prediction to be all the words which are related with common stem.

In the table, we first evaluate (2nd row) the traditional word spotting performance (“exact” criteria) across all the datasets. Note that the networks trained for IAM and GW are first pre-trained on the HW-SYNTH dataset. The obtained results are slightly better to the ones reported in the original work [4] due to the usage of the TPP layer along with better implementation strategies as explained in Section 6.3. As shown in the table, we first observe that using the exact representation as the Concept GT, the evaluation of retrieving words having common stem and lemma is inferior since the original representation is a holistic feature focusing on the entire word. In the other two scenarios where we use Concept GT as either stem or lemma, we observe that performance of exact evaluation drops. This is because for “exact” retrieval, the words having common roots are deemed false positives. However, the results on the original training criteria where the Concept GT and Evaluation are same are quite promising and better than their corresponding counterparts where the Concept GT is exact. This demonstrates that the network is able to focus on the root form of the words and put lesser importance to the affixes (prefix and suffix) that are common across language. In the last row, we compare the performance of the original PHOCNet [23] architecture using the publicly available pre-trained model on IAM dataset. Much similar to our model trained on Exact Concept GT, we see drop in performance under stem and lemma evaluation criteria. However, the drop in PHOCNet is much higher as compared to our model. The qualitative results of our method are presented in Section 6.5.2.

6.5. Semantic spotting

Table 2 presents the quantitative analysis of the proposed semantic representation learned under different schemes. As mentioned in Section 6.2, we use standard measures from textual word embedding literature such as word similarity and word analogy to evaluate our representation. Note that, under both these schemes we report mean performance along with the standard deviation of the results across multiple trial runs. In text literature, one does not see such trials because for each textual word, there exists only one representation. However, in case of image corpus, for each textual word one can sample images under different styles. Therefore, to be fair in evaluation, we conduct multiple trials, where in each trial and for each textual word, we sample a random style image from our corpus and take its representation for evaluation purposes. In addition, we also compute the mAP-Exact similar to Normalized Spotting evaluation (as shown in Section 6.4). This will help to understand the performance of the semantic representation for lexical word spotting where the goal is to only retrieve exact similar images. Note that we have avoided evaluating GW dataset for semantic tasks (WS353 and Word Analogy), since the corpus has very few images which intersect with the words in WS353 and word analogy datasets.

In this work, for the purpose of embedding word images into semantic space, we used fastText pre-trained models [49] for English. The first method reported in the table (fastText) evaluates our ground truth representation under semantic evaluation measures. Here, for the analogy task, we only take those words (anal-

[2] https://cvit.iiit.ac.in/research/projects/cvit-projects/sem-embed.

Table 2
Quantitative evaluation of word image semantic representation. Following shortened notation are used: word analogy (WA), query-by-example (QBE), query-by-string (QBS) and in-vocabulary QBS (In-QBS). As mentioned in Section 6.2, we use Spearman's rank correlation co-efficient (0,1) to report word similarity measure in WS353 dataset. While, WA uses word accuracy (0-100%) and semantic spotting reports the mean Average Precision (mAP) (0-100%). In each of the performance measure higher the value, better the method.

Method	HW-SYNTH				IAM				GW		
	WS353	WA	mAP-Exact		WA	mAP-Exact			mAP-Exact		
			QBE	QBS		QBE	QBS	In-QBS	QBE	QBS	In-QBS
fastText [49]	**0.7300**	81.97	-	-	83.45	-	-	-	-	-	-
NormSpot-Exact	0.0799 ± 0.018	17.15 ± 0.4	**98.49**	**98.96**	23.17 ± 1.1	**92.54**	**96.54**	**97.01**	**99.37**	**99.46**	**99.67**
RecEmbed-90K	**0.6841 ± 0.053**	70.67 ± 0.5	96.85	98.19	60.19 ± 2.6	79.78	88.31	89.43	95.37	93.29	92.02
Sem-MSE	0.5660 ± 0.05	67.73 ± 1.1	92.82	94.76	63.22 ± 3.1	84.63	69.72	88.12	97.63	94.38	98.94
Sem-Rank	0.6498 **± 0.04**	**79.90 ± 1.1**	93.41	95.53	**65.64 ± 3.1**	83.28	71.26	86.13	97.83	93.72	98.79
Sem-Rank+NormSpot	0.5912 ± 0.038	67.88 ± 0.6	97.96	98.78	61.53 ± 2.8	90.61	94.28	95.11	99.35	98.78	99.55
Triplet [11] (Semantic)	-	-	-	-	-	81.58	75.74	-	96.91	69.81	-

ogy question pairs) which are present in our test corpus of each dataset. The numbers reported under fastText could be taken as the upper bound for our proposed semantic representation for word images. Next, we measure the performance of our normalized word representation (Table 1, ConceptGT:Exact) which is assumed to contain only the lexical representation. Here, we are interested in understanding whether such a representation encodes semantics or not? As one can clearly notice, the performance on both WS353 and Analogy is considerably low in all the datasets, which states the need for dedicated encoding of semantics while training. The next three rows in the table present the results of the proposed semantic representation as discussed in Section 5.2. Here, RecEmbed-90K refers to our two stage recognition+embedding scheme under the lexicon size of 90K which is significantly larger than the test lexicons in these datasets and thereby most of the words acts as distractors. The methods Sem-MSE and Sem-Rank refer to learned representations under direct embedding scheme using the mean square error loss and ranking loss respectively. As compared to the performance of NormSpot, we observe that is that there is a significant improvement under semantic evaluation criterion (WS353 and Word Analogy), however there is also a drop of nearly 8 – 12% in IAM mAP-Exact evaluation while for GW and HW-SYNTH it in range of 2 – 5%. As mentioned earlier, the loss in performance for "exact" evaluation is due to inducing semantics. However, in order to strike a balance between semantic and exact word spotting without compromising much on either, in the next row (Sem-Rank+NormSpot), we evaluate the performance of representation obtained by concatenating both the semantic embedding and normalized word embedding. Here, we observe a huge improvement of performance in exact word spotting which is now comparable to NormSpot-Exact while there is some drop in semantic evaluation measures (WS353,WA). However this drop is still better or comparable to Sem-MSE method and much better than original NormSpot-Exact method.

The other set of observation to notice is between recognition-based and recognition-free setting. We see that in HW-SYNTH dataset, which contains synthetic images and thereby easier to recognize, RecEmbed (recognition-based) method works better in analogy and mAP-Exact than Sem-MSE and Sem-Rank (recognition-free). However, for real handwritten datasets, direct embedding is better in most of the scenarios. This emphasizes the importance of direct embedding schemes where actual recognition is difficult. Within direct embedding schemes, we observe that although both obtain a comparable mAP-Exact performance, ranking based loss provides better results under semantic measures of word similarity using WS353 and word analogy tasks. This validates that a triplet based ranking loss captures the relationships among samples in the embedding space much better as compared to the MSE loss. Another important point to note here is that the mAP performance of query-by-string (QBS) for both Sem-MSE and Sem-Rank is inferior than its query-by-example (QBE) counterpart. The reason is the out-of-vocabulary (OOV) words in the test set. Since these OOV words did not exist while training, their embeddings through real stream does not occupy the expected semantic space where the textual strings from semantic stream exist. Due to this reason, we also provide a In-QBS measure which only measures the in-vocabulary performance.

The last row in the table compares our representations from the embedding proposed in Wilkinson and Brun [11] which is the closest method in literature which does semantic spotting similar to ours. However, the paper does not report semantic evaluation results as done in this work, but presents the results qualitatively. As shown, the table reports their performance for exact (verbatim) spotting using their semantic representation which is learned using a character level language model. Although, the proposed representations of this work obtain better results on mAP-Exact measure, since [11] have not reported semantic evaluation results on word similarity and analogy, we could not directly compare among the two methods.

6.5.1. Ablation study

Fig. 6 (a-b) presents an ablation study done for lexicon based recognition and embedding on the IAM dataset. The chart on the top shows semantic spotting performance in terms of mAP for exact evaluation, while the chart on the bottom reports the word (WER) and character error rates (CER) of the recognizer. In both the cases, the x axis shows the varying lexicon size of order 0K-90K words. Here, 0K refers to only having lexicon words taken from the test set. Each lexicon set is the union of test set words and other words from English vocabulary acting as distractors. As one can clearly notice, in both the scenarios the drop in performance is moderate and even on using an extremely large lexicon of size 90K, we obtain an mAP of 88.31% for QBS, word error rate of 8% and character error rate of 3.9%.

6.5.2. Qualitative analysis

Inspired by the popular example of $king - man + woman = queen$ in the textual word embedding literature [2], Fig. 6(c) presents few such example analogical questions and their answers obtained using the proposed semantic representation of word images. Here, we present two scenarios where the top block shows the results when the query $B - A + C =?$ was formulated using its textual representation (analogous to QBS), whereas in the middle block, the query was entirely formulated using the word image representation (analogous to QBE). In both the cases, we show the result as the nearest word image representation (top-1) in the image corpus to its corresponding query formulation. The top and the middle blocks are taken from the IAM dataset where the

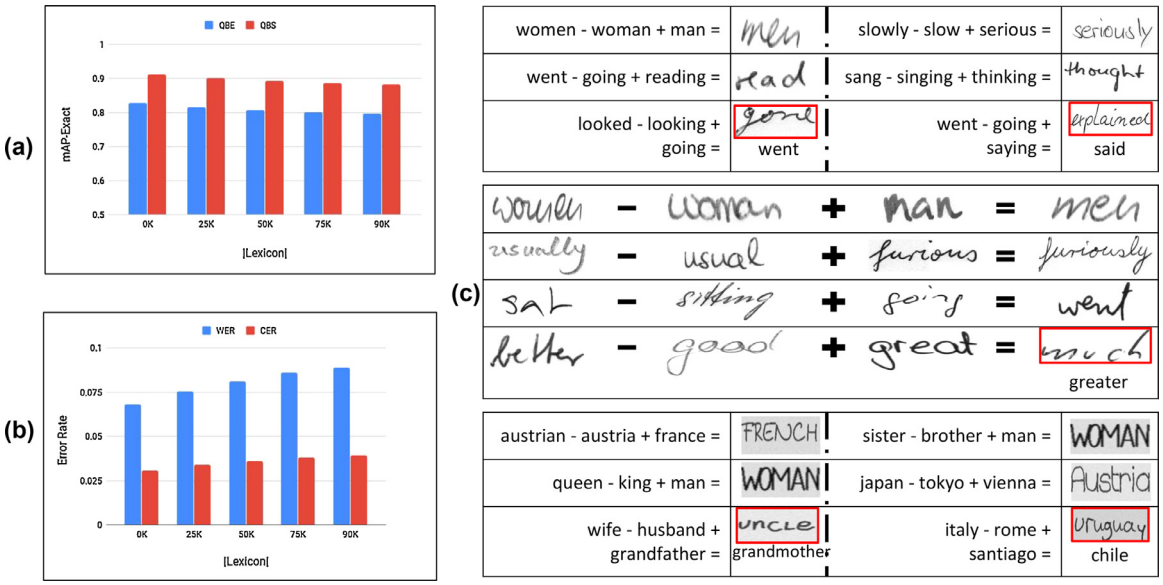

Fig. 6. (a-b) Rate of the performance change while increasing the lexicon size in lexicon based recognition and embedding for the IAM dataset. (a) shows semantic spotting performance in terms of mAP for exact evaluation, while the chart (b) reports the word and character error rates of the recognizer. (c) Qualitative results of analogical questions and their answers obtained using the proposed semantic representation of word images.

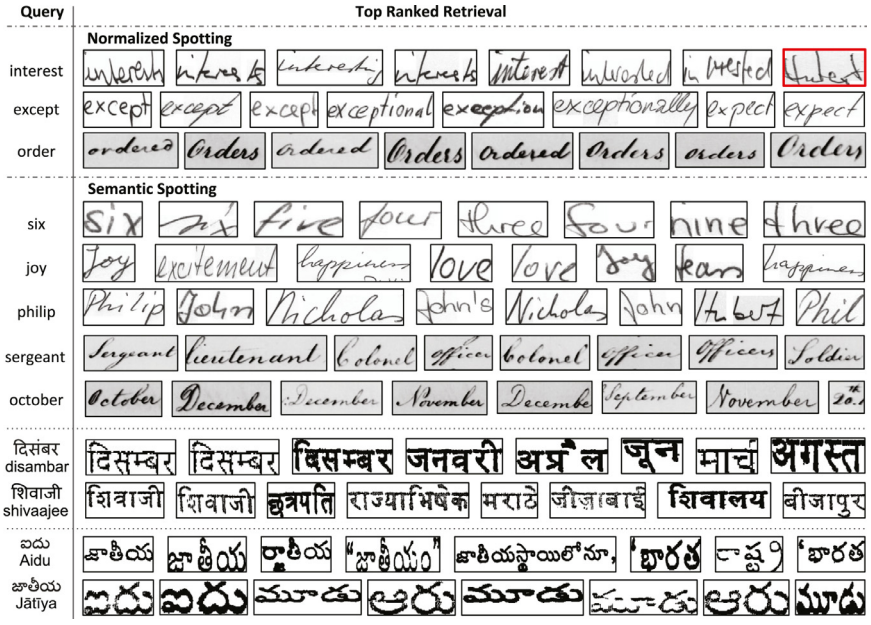

Fig. 7. Qualitative results of normalized and semantic word spotting. Note that, while showing the nearest neighbors, we have removed the consecutive similar word images to emphasize the distances among different lexical word images.

queries were mostly syntactic in nature due to the limited vocabulary coverage of the dataset. In order to demonstrate semantic queries (for e.g. relationships in *country, family*), we utilize our synthetic dataset HW-SYNTH as shown in the bottom block. One can clearly notice that the learned features preserve the linear semantic relationships of the original word embedding on which it was trained. These features are largely invariant to handwritten styles and degradations. In the Fig. 6(c), the last rows of each block shows the failure scenarios (marked within a red bounding box) where the nearest image retrieved for the particular query was wrong. E.g., for the query *looked* − *looking* + *going* = *gone* is a wrong match, whereas the correct match is *went* as per the ground truth. We observe that the failure scenarios are reasonable and not far from its actual answer.

Fig. 7 presents the qualitative results from the learned representation for the word spotting task. The first three rows show the results of the proposed normalized word embedding where the representation is invariant to the word form inflections. The results are shown for the network trained using stem information. Here, one can notice that for the query *interest* we obtain top results which include *interest, interests, interesting, interested* irrespective of multiple handwriting styles. The first two rows belong to the IAM dataset, while the third row presents results from the GW dataset. The next five rows present results using the proposed se-

Table 3
Semantic evaluation on printed datasets from English, Hindi and Telugu languages. Here also we report the mean average precision (mAP) for semantic spotting and word analogy (WA) reports the word level accuracy. Both these values are reported in percentages (0-100%).

Method	Supervision	English			Hindi			Telugu	
		WA	mAP-Exact		WA	mAP-Exact		mAP-Exact	
			QBE	QBS		QBE	QBS	QBE	QBS
fastText [49]	-	81.40	-	-	57.14	-	-	-	-
Yalniz et al. [17]	No	-	93.00	-	-	-	-	-	-
Krishnan et al. [50]	No	-	-	-	-	60.55	-	74.38	-
HWNet v2 (TPP) [12]	Yes	-	95.70	-	-	95.09	-	95.82	-
Sem-Rank (This Work)	Yes	79.01 ± 0.3	99.05	95.66	53.21 ± 1.8	94.28	93.59	95.38	94.61

mantic embedding where we have showcased the results for retrieving semantically relevant word images. E.g., the query *six* retrieves *six* and other nearby numbers, the query *joy* retrieves *joy, excitement, happiness, love* etc. Here, the query *philip* is an interesting case where we observe that the query being a named entity, which may not have any semantic sense, the top similar results are from other named entity words such as *john, nicholas, hubert, phil*, etc. Please note that while showing the semantic spotting results, we have avoided few successive results which have same ground truth label in order to show the diversity. Also the false positives are marked inside red bounding box.

6.5.3. Results on printed documents

In order to validate the proposed framework on different scripts and modalities, we now present our results on scanned document images taken from printed books in both English and Indic scripts.

English-1601 [17]: The dataset contains a single book in English titled "Adventures of Sherlock Holmes" written by Arthur Conan Doyle. This was first used in [17] for comparing OCR based results with image search.

DLI Hindi and Telugu [50]: These two datasets belonging to Hindi and Telugu languages from Indic scripts are part of the Digital Library of India (DLI) [51] project. DLI has emerged as one of the largest collections of document images in Indian scripts. Many of the pages present in DLI contain serious forms of document degradation which restricts present day OCRs and text spotting systems to work efficiently. We take one such subset [50] which was annotated at the level of lines and words, and are referred to as HS1 and TS1 datasets.

Table 3 presents the results on these datasets using the same metrics as used earlier. Similar to Table 2, fastText method refers to text based representation which we use for semantic embedding and its evaluation on analogy tasks is depicted in column WA. There are very few prior works which have reported results on these datasets. We take three such works among which the methods proposed in [17,50] uses BoWs based features which are computed in an unsupervised setting and could not be directly compared with supervised methods as shown in this work. As one can notice, the supervised methods result in better representation learning. We also compare our earlier work HWNet v2 [12] with the Sem-Rank method as proposed in this work. We observe that in terms of analogy performance (WA), the gap between the fastText and Sem-Rank is quite closer than the one reported for handwritten domain. Also, one can see a similar trend in the performance of mAP-Exact between Sem-Rank and HWNet v2 which was trained in a lexical manner. Here, interestingly for the printed English dataset, we obtain a better result than HWNet v2 on the lexical evaluation. The bottom four rows in Fig. 7 show qualitative results for Hindi and Telugu language word images. Note that the existing dataset is binary in nature and contains degradations due to the binarization process. Here the top results are meaningful and semantically coherent in nature.

6.6. Application to POS tagging

In order to validate the utility for NLP tasks, we formulated a simple classification task using our learned representation to perform parts of speech (POS) tagging from word images. We used the IAM dataset which already has POS annotations. The original annotations are fine-grained with nearly 200+ different tags. We manually mapped these to around 36 tags taking parts-of-speech tags of Penn Treebank project as reference. We trained a simple two-layer perceptron with hidden dimensions of 512 weights. We used ReLU as the activation function and also added batch normalization. We evaluate the classification performance in terms of F_1 score since the number of words in each class is skewed in nature. Note that the learned representations are taken from the models presented in Section 6.5. We obtain a F_1 score of 70.20% for the Sem-Rank model while for the NormSpot-Exact model the F_1 score is 68.60%. This reinforces our assumption that semantic features would be a better choice for performing NLP tasks on document images. Note that our experiment is about proving the validity of semantic features and the considered model architecture may not be the optimum design choice.

7. Conclusion and future work

In this work, we have presented new forms of holistic representations for word images which preserve the semantic properties of words along with their forms. We introduced two such frameworks: a) a normalized word embedding, which is invariant to multiple word inflections, and b) semantic embedding that draws its properties along modern textual word embedding schemes and are more generic in capturing the relationships among semantically similar words. Our experiments, presented promising results for semantic evaluation tasks such as word analogy and word similarity. In our analysis, we also performed a direct comparison with the traditional word spotting task, with a belief that an ideal semantic representation should also be optimum for the purpose of exact evaluation. However in this case, we observed that there existed a gap in the performance. One of the major reasons for this gap is that, due to the infusion of semantically similar words along with the exact matches, there is a reduction in precision for the exact evaluation. One can notice this issue for the query *joy* in Fig. 7 (Row: 5), where we see the exact matches are at rank 1, and 6, while the top results also contain other related words which are relevant in semantic sense.

Our work presented two ways for doing semantic embedding: (i) lexicon based recognition and embedding, and (ii) direct embedding. Here we observe that the direct embedding gives better

performance for semantic tasks (analogy and similarity) whereas the recognition scheme gives better performance for exact spotting. The limitation in the direct embedding scheme is that its semantic coverage is limited to training vocabulary, whereas for the recognition scheme, it is limited by the lexicon words which are independent of the training vocabulary. In our experiments, we tested the limits of recognition based scheme by having a huge lexicon of size 90K words where we observed the drop in performance to be marginal. However, there exists a typical issue in the recognition based framework when the recognition of word image fails. In these cases, we typically embed into a wrong semantic representation which may be completely unrelated with query. We believe that a direct embedding scheme is relatively better in such setting of failures since it can take partial cues from the root of words (if such roots existed in training corpus).

As a future work, we notice that there exists more scope in fusing the traditional lexical representation along with semantic representation, which could exploit their complementary properties. We performed one basic experiment (Sem-Rank+NormSpot) in this direction, as shown in Section 6.5 where we concatenated both the representations. The initial results proves the direction worth to pursue. Another interesting direction of work which takes upon one of the limitation of the current work is dealing with out of vocabulary words (OOVs). We indeed believe that char/n-gram level embeddings could partly solve the problem of OOVs as demonstrated in the NLP community [32,33].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowlgedgments

This work is partly supported by IMPRINT. Praveen Krishnan is supported by Amazon Alexa Graduate Fellowship.

References

[1] C. Anderson, Essentials of linguistics, Last Retrieved 2019-01-19 (https://essentialsoflinguistics.pressbooks.com/).
[2] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, NIPS, 2013.
[3] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, EMNLP, 2014.
[4] P. Krishnan, K. Dutta, C.V. Jawahar, Word spotting and recognition using deep embedding, DAS, 2018.
[5] J.R. Rohlicek, W. Russell, S. Roukos, H. Gish, Continuous hidden markov modeling for speaker-independent word spotting, ICASSP, 1989.
[6] R. Manmatha, C. Han, E.M. Riseman, Word spotting: A new approach to indexing handwriting, CVPR, 1996.
[7] C. Adak, B.B. Chaudhuri, M. Blumenstein, Named entity recognition from unstructured handwritten document images, DAS, 2016.
[8] J.I. Toledo, S. Sudholt, A. Fornés, J. Cucurull, G.A. Fink, J. Lladós, Handwritten word image categorization with convolutional neural networks and spatial pyramid pooling, S+SSPR, 2016.
[9] P. Krishnan, C.V. Jawahar, Matching handwritten document images, ECCV, 2016.
[10] A. Sharma, D.B. Jayagopi, Automated grading of handwritten essays, ICFHR, 2018.
[11] T. Wilkinson, A. Brun, Semantic and verbatim word spotting using deep neural networks, ICFHR, 2016.
[12] P. Krishnan, C.V. Jawahar, HWNet v2: an efficient word image representation for handwritten documents, IJDAR (2019).
[13] T.M. Rath, R. Manmatha, Word spotting for historical documents, IJDAR (2007).
[14] U. Marti, H. Bunke, Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system, IJPRAI (2001).
[15] J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in videos, ICCV, 2003.
[16] M. Rusiñol, D. Aldavert, R. Toledo, J. Lladós, Browsing heterogeneous document collections by a segmentation-free word spotting method, ICDAR, 2011.
[17] I.Z. Yalniz, R. Manmatha, An efficient framework for searching text in noisy document images, DAS, 2012.
[18] D. Aldavert, M. Rusiñol, R. Toledo, J. Lladós, A study of bag-of-visual-words representations for handwritten keyword spotting, IJDAR (2015).
[19] D.G. Lowe, Distinctive image features from scale-invariant keypoints, IJCV (2004).
[20] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inform. Sci. (1990).
[21] J. Almazán, A. Gordo, A. Fornés, E. Valveny, Word spotting and recognition with embedded attributes, PAMI (2014).
[22] A. Poznanski, L. Wolf, CNN-N-Gram for handwriting word recognition, CVPR, 2016.
[23] S. Sudholt, G.A. Fink, PHOCNet: A deep convolutional neural network for word spotting in handwritten documents, ICFHR, 2016.
[24] S. Sudholt, G.A. Fink, Attribute CNNs for word spotting in handwritten documents, IJDAR (2018).
[25] P. Krishnan, K. Dutta, C.V. Jawahar, Deep feature embedding for accurate recognition and retrieval of handwritten text, ICFHR, 2016.
[26] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR (2014).
[27] S. Sudholt, G.A. Fink, Evaluating word string embeddings and loss functions for CNN-based word spotting, ICDAR, 2017.
[28] Y. Kim, Convolutional neural networks for sentence classification, EMNLP, 2014.
[29] A. Passos, V. Kumar, A. McCallum, Lexicon infused phrase embeddings for named entity resolution, CoNLL (2014).
[30] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, JMLR (2003).
[31] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, ICLR (2013).
[32] Y. Kim, Y. Jernite, D. Sontag, A.M. Rush, Character-aware neural language models., AAAI, 2016.
[33] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, TACL (2017).
[34] P. Krishnan, C. Jawahar, Bringing semantics in word image retrieval, ICDAR, 2013.
[35] G.A. Miller, Wordnet: a lexical database for english, Commun. ACM (1995).
[36] A. Gordo, J. Almazán, N. Murray, F. Perronin, LEWIS: latent embeddings for word images and their semantics, ICCV, 2015.
[37] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, NIPS, 2012.
[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR, 2016.
[39] M.F. Porter, An algorithm for suffix stripping, Program (1980).
[40] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.
[41] E. Loper, S. Bird, Nltk: The natural language toolkit, ETMTNLP, 2002.
[42] U. Marti, H. Bunke, The IAM-database: an english sentence database for offline handwriting recognition, IJDAR (2002).
[43] P. Krishnan, C.V. Jawahar, Generating Synthetic Data for Text Recognition, arxiv, 2016.
[44] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge university press, 2008.
[45] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, E. Ruppin, Placing search in context: the concept revisited, ACM Trans. Inf. Syst. (2002).
[46] C. Spearman, The proof and measurement of association between two things, Am. J. Psychol. (1904).
[47] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, ICCV, 2015.
[48] P.Y. Simard, D. Steinkraus, J.C. Platt, Best practices for convolutional neural networks applied to visual document analysis, ICDAR, 2003.
[49] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Learning word vectors for 157 languages, LREC, 2018.
[50] P. Krishnan, R. Shekhar, C.V. Jawahar, Content level access to Digital Library of India pages, ICVGIP, 2012.
[51] V. Ambati, N.Balakrishnan, R. Reddy, L. Pratha, C.V. Jawahar, The Digital Library of India Project: Process, Policies and Architecture, ICDL, 2007.

Praveen Krishnan is a PhD candidate in Computer Science at IIIT Hyderabad. His primary research interests include document image analysis and computer vision.

C. V. Jawahar is a Professor at IIIT Hyderabad, India. His areas of research include computer vision, machine learning and document image analysis.