Textual Visual Semantic Dataset for Text Spotting

Ahmed Sabir[1] Francesc Moreno-Noguer[2] Lluís Padró[1]
[1] Universitat Politècnica de Catalunya, TALP Research Center, Barcelona, Spain
[2] Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain

Abstract

Text Spotting in the wild consists of detecting and recognizing text appearing in images (e.g. signboards, traffic signals or brands in clothing or objects). This is a challenging problem due to the complexity of the context where texts appear (uneven backgrounds, shading, occlusions, perspective distortions, etc.). Only a few approaches try to exploit the relation between text and its surrounding environment to better recognize text in the scene. In this paper, we propose a visual context dataset[1] for Text Spotting in the wild, where the publicly available dataset COCO-text [40] has been extended with information about the scene (such as objects and places appearing in the image) to enable researchers to include semantic relations between texts and scene in their Text Spotting systems, and to offer a common framework for such approaches. For each text in an image, we extract three kinds of context information: objects in the scene, image location label and a textual image description (caption). We use state-of-the-art out-of-the-box available tools to extract this additional information. Since this information has textual form, it can be used to leverage text similarity or semantic relation methods into Text Spotting systems, either as a post-processing or in an end-to-end training strategy.

1. Introduction

Recognition of scene text in images in the wild is still an open problem in computer vision. There exist a number of difficulties in recognizing texts in images due to the many possible lighting conditions, variations in textures, complex backgrounds, textual font types and perspective distortions. The ability to automatically detect and recognize text in natural images, a.k.a. *text spotting or OCR in the wild* is an important challenge for many applications such as visually-impaired assistants [21] or autonomous vehicles [30]. In recent years, the interest of Computer Vision community in Text Spotting has significantly increased [1, 36, 16, 8, 7, 32]. However, state-of-the-art scene text recognition methods do not leverage object and scene recognition. Therefore, in this work, we introduce a visual semantic context textual dataset (*e.g.* object, scene information) for text spotting tasks. Our goal is to fill this gap, bringing closer vision and language, by understanding the scene text and its relationship with the environmental visual context.

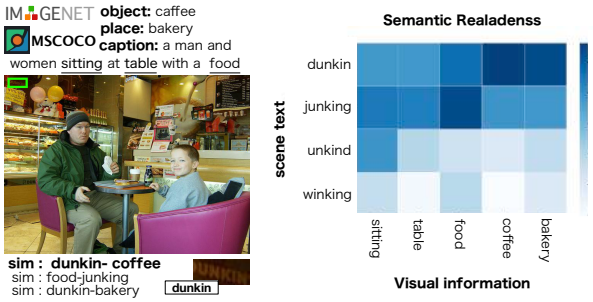

Figure 1. Overview of the natural language understanding visual context information. The word "dunkin" has a strong semantic relation with "bakery", "food", and "coffee", thus it will be more likely to appear in this image than other similar words such as "junking" or "unkind". Note that this relation is computed by pretraind word2vec [25] cosine similarity.

The relation between text and its surrounding environment is very important to understand text in the scene. While there are some publicly available datasets for text spotting, none includes information about visual context in the image. Therefore, we propose a visual context semantic knowledge dataset for the text spotting pipeline, as our aim is to combine natural language processing and computer vision. In particular, we exploit the semantic relatedness between the spotted text and its image context. For example, as shown in Figure 1 the word "dunkin" has a stronger semantic relation with "coffee", thus it will be more likely to appear in a visual context than other possible candidates such as "junking" or "unkind".

Departing from [34, 33], in this paper we describe in depth the construction of the visual context dataset. This dataset is based on the COCO-text [40], which uses Mi-

[1]Our dataset is publicly available at: https://git.io/JeZTb

Table 1. Several word recognition datasets. The images are cropped word images useful only for the recognition task only.

Text Recognition Dataset				
Label	Description	Dictionary	# bbox	# text †
IC17-T2	from COCO-text dataset ICDAR 17 Task-2 [9]	-	46K	-
Synth90K	Synthetic dataset with 90K dict [15]	90K	9M	-
SVT	Street View Text [42]	-	647	-
IC13	ICDAR 2013 [19]	-	1K	-
IC17-V	Image+Textual dataset from IC17 Task-3 (ours)	-	10K	25K
COCO-Text-V	Image+Textual dataset from COCO-text (ours)	-	16K	60K
COCO-Pairs	Only Textual dataset from COCO-text (ours)	-	-	158K

† Each sample have bounding box and its full image and the textual visual context (object, scene and caption).

crosoft COCO dataset [22] and annotates texts appearing in the images. We further extend the dataset using out-of-the-box tools to extract visual context or additional information from images. Our main contribution is this combined visual context dataset, that provides the unrestricted-OCR research community the chance to use semantic relatedness between text and image to improve the results. The computer vision community tackles this problem by dividing the task into two sub-models one for text, and other for object [46, 18, 29]. Our approach uses existing state-of-the-art visual context generation approaches and thus, it can be used as a visual context with text spotting as post-processing (OCR correction) or end-to-end training.

2. Related Work

While there are some publicly available datasets for text spotting, none of them includes visual context information such as objects in the scene, location id or textual image descriptions. In this section we describe several publicly available text spotting datasets.

2.1. Synthetic Dataset

Table 1 summarizes the number of examples in different datasets. Sizes of real image datasets with annotated texts are in the order of thousands, and have a very limited vocabulary, which makes them insufficient for deep learning methods. Therefore, [15] introduced a synthetic data generator without any human label cost. Words are sampled from a 90K-words dictionary, and are rendered synthetically to generate images with complex background, fonts, distortions, etc. It contains 9 million cropped word text images. All current state-of-the-art text spotting algorithms [36, 16, 8, 7] are trained on this dataset.

2.2. ICDAR Dataset

Text Spotting shared tasks carried out at ICDAR conferences released several relevant datasets:

ICDAR 2013 (IC13) [19]. The ICDAR 2013 dataset consists of two sections for different text spotting subtasks: (1) text localization and (2) text segmentation. Text localization consists of 328 training images and 233 test images. Given its reduced size, ICDAR 2013 dataset is typically used for evaluation of scene text understanding tasks: localization, segmentation, and recognition.

ICDAR 2017 (IC17) [9]. ICDAR 2017 is based on COCO-text [40] and aims for end-to-end text spotting (*i.e.* detection and recognition). The dataset consists of 43,686 full images with 145,859 text instances for training, and 10,000 images and 27,550 instances for validation.

Street View Text (SVT) [42]. This dataset consists of 349 images downloaded from Google Street View. For each image, only one word-level bounding box is provided. This is the first dataset that deals with text image in real scenarios, such as shop signs in a wide range of fonts styles.

COCO-Text [40]. This dataset is based on Microsoft COCO [22] (Common Objects in Context) and consists of 63,686 images, 173,589 text instance (annotations of the images). The COCO-text dataset differs from the other datasets in three aspects. First, the dataset was not collected with text recognition in mind. Thus, the annotated text instances lie in their natural context. Second, it contains a wide variety of text instances, such as machine-printed and handwritten text. Finally, the COCO-text has a much larger scale than other datasets for text detection and recognition.

3. Source Data

We use state-of-the-art tools to extract textual information for each image. In particular, for each image we extract: 1) spotted text candidates (text hypotheses), and 2) surrounding visual context information.

3.1. Text Hypotheses Extraction

To extract the text associated with each image or bounding box we employ several off the self pre-trained Text Spotting baselines to generate k text hypotheses. All the pre-trained models are trained on a synthetic dataset [15]. We build out text hypotheses dataset for each image as the

union of the predictions of all baselines. We next describe these models.

Convolutional Neural Network-90K-Dictionary [16]. The first baseline is a CNN with fixed lexicon based recognition, able to recognize words in a predefined 90K-word dictionary W. Each word w is corresponds to a word (class) in the 90K dictionary W (multi-class classification). The dictionary is composed of various forms of English words (*e.g.* nouns, verbs, adjectives, adverbs, etc) In short, the model classifies each input word-image into a pre-defined word in the 90K fixed lexicon. Each word $w \in W$ in the dictionary corresponds to one output neuron. The final output $word$ for a given image x is written as:

$$word = \arg\max_{w \in \mathcal{W}} P(w|x, lexicon) \quad (1)$$

Convolutional Recurrent Neural Network (CRNN) [36]. The second baseline is a CRNN that learns the words directly from sequence labels, without relying on character annotations. The encoder uses a CNN to extract a set of features from the image. The CNN has no fully connected layers and extracts sequential feature representations of the input image, which are fed into a bidirectional RNN. A Connectionist Temporal Classification [10] based method is used to convert the per-frame predictions made by the RNN into a label sequence as following:

$$\mathbf{I}^* \approx \mathcal{B}\left(\arg\max_{\pi} p(\boldsymbol{\pi}|\mathbf{y})\right) \quad (2)$$

where $\mathcal{B}$ is sequence-to-sequence mapping function, $\boldsymbol{\pi}$ sequence label and $\mathbf{I}^*$ the sequence.

LSTM-Visual Attention (LSTM-V) [8]. The third baseline also generates output words as probable character sequences, without relying on lexicon. The network is based on encoder-decoder architecture with visual attention mechanism. In particular, they use the CNN pre-trained model [16] mentioned above as encoder, but without the final layer, to extract the most important feature vectors from each text image. That feature vector is used to reduce model complexity through the soft attention model [43], which focus on the most relevant parts of the image at every step. The LSTM [12] decoder computes the output character probability y for each time step and the visual attention α, where L are the deep model output parameters of each layer:

$$P\left(y_t|\alpha, y_{t-1}\right) \sim \exp\left(\mathrm{L}_0\left(Ey_{t-1} + \mathrm{L}_\mathrm{h} h_t + \mathrm{L}_\mathrm{z}\hat{z}_t\right)\right) \quad (3)$$

CNN-Attention [7]. Finally, we employ one of the most recent state-of-the art systems, which also produces the final output words as probable character sequences, without any fixed lexicon. The model is based on a CNN encoder-decoder with attention and a CNN character language model. The final character prediction is a element-wise addition of the attention and language vector as:

$$p\left(y_k|y_{k-1}, \ldots, y_1\right) = p_a + p_l \quad (4)$$

where p_a and p_l are softmax functions that convert the attention and language vectors to predicted characters separately.

3.2. Visual Context Information

To extract the *visual context* from each image, we use out-of-the-box state-of-the-art classifiers. We obtain three kinds of contextual information: objects in the image, location/scenario labels, and a textual description or caption.

3.2.1 Object information

The output of the following classifiers is a 1000-dimensional vector with the probabilities of 1000 object types. We retain the top-5 most likely objects.

GoogLeNet [38]. The design of this network is based on an inception module, which uses 1-D convolutions to reduce the number of parameters. Also, a fully connected layer is replaced with a global average pooling at the end of the network. The network consists of 22 layer Deep CNN with reduced parameters. It has a top-5 error rate of 6.67%.

Inception-ResNet [37]. Inspired by the breakthrough ResNet performance, a hybrid-inception module was proposed. Inception-ResNet combines the two architectures (Inception modules and residual connection) to boost performance even further. We use Inception-ResNet-v2, with a top-5 error rate 3.1%.

The object hypotheses are obtained by extracting top-5 error class labels from each classifier and re-ranking them based on the cosine distance.

3.2.2 Scene information

To extract scene information, we considered just one scene classifier [45]. This is a pre-trained scene classifier able to recognize 365 different scenario classes. The original model is based on Places365-Standard as deep convolutions network that trained on 1.8 million images from 365 scene categories. The same work proposed a better model, which we use, consisting of a fine-tuned model *Places365-ResNet*[2] based on ResNet architecture.

3.2.3 Image description

Finally, we use a caption generator to extract more visual context information from each image, as a natural language description. Image caption generation approaches can use either top-down or bottom-up approaches. The bottom-up approach consists of detecting objects in the image and then attempting to combine the identified objects into a caption [20]. On the other hand, the top-down approach learns the semantic representation of the image which is then decoded into the caption [41]. Most current state-of-the-art systems

[2] http://places2.csail.mit.edu/

Table 2. Sample from the dataset. The text hypothesis comes from existing Text Spotting baselines and the visual context information comes from out-of-the-box computer vision classifiers. The **bold** font is the ground-truth.

Text hypothesis	Object	Scene	Caption
11, il, j, m, ...	railroad	train	a train is on a train track with a train on it
lossing, docile, dow, **dell**, ...	bookshop	bookstore	a woman sitting at a table with a laptop
29th, 2th, **2011**, zit, ...	parking	shopping	a man is holding a cell phone while standing
happy, hooping, happily, nappy, ...	childs	bib	a cake with a bunch of different types of scissors
coke, gulp, slurp, fluky,...	plate	pizzeria	a table with a pizza and a fork on it
will, **wii**, xviii, wit,....	remote	room	a close up of a remote control on a table

adopt the *top-down* approach using RNN-based architectures. In this work, we use the latter top-down model to extract the visual description of the image.

The caption generator encoder of [41] uses a ResNet architecture [11] trained on ILSVRC competition dataset for general image classification task, and the decoder is tuned on COCO-caption [22], the same dataset for which we extract all visual context information. Table 3 shows that the caption has richer semantic.

4. Dataset Construction

4.1. Text hypothesis selection

As described above, the output of several text spotting systems is included in the dataset as text hypotheses or possible candidates for each image. However, some filtering is applied to remove duplicates and unlikely words:

First, we use a unigram language model (ULM) to filter out rare words (*e.g. pretzel*), non-words (*e.g. tolin*), or very short words (*e.g. inc*) unlikely to be in the image. The ULM [34] was built from Opensubtitles [23][3], a large database of movie subtitles containing around 3 million unique word forms, including numbers and other alphanumeric combinations that make it well suited for our task. We combined this corpus with google-ngrams[4] that contains 5 million tokens from English literature books. The combined corpora contain around 8 million tokens as shown in Table 4.

Secondly, we add the *ground-truth* if it was removed by the filter or if it was not included in the hypothesis list generated by the baselines. Note that this may occur often, since according to the author of COCO-text [40] the significant shortcoming of this dataset is a bounding box detection recall. Therefore, in about 40% of the images, the text is not properly detected and thus the classifiers fail to recognize it.

4.2. Visual context selection

Despite we extract the top-5 objects from each image, we use a semantic similarity measure and threshold to filter out predictions where the object classifier is not confident enough. We use two approaches to filter out duplicated cases and false positive example.

Threshold measure. First, we consider a threshold $P(w|class) < 0.5$ to extract the most likely classes in the images, and eliminate low confidence predictions.

Semantic alignment. We use the cosine similarity to select the most likely visual context in the image. Concretely, we use a general text word-embedding [25, 27] to compute the similarity score between different visual context elements, and then we select objects or places detected with: 1) a high confidence and that have 2) strong semantic similarity with other image elements. The underlying assumption is that if two objects in the image are related, the classifier prediction we are relying on will be more likely to be correct.

4.3. Object and text co-occurrence database

Finally, we enrich the dataset with text-object co-occurrence frequencies. Since this information is not associated to each image, but is an aggregated of the whole dataset, it is provided in a separate table. This co-occurrence information may be useful when the text hypotheses and the scenes are not close in the semantic space but they are in the real world (*e.g. delta* and *airliner* or the sports TV channel *kt* and *racket* may not be close according to a general word embedding model, but they co-occur often in the image dataset). A sample of these co-ocurrence frequencies is shown in Figure 3 (b).

The co-occurrence information [34] can be used to estimate the conditional probability $P(w|c)$ of a word w given that object c appears in the image:

$$P(w|c) = \frac{freq(w,c)}{freq(c)} \tag{5}$$

where $freq(w,c)$ is the number of training images, w appears as the gold standard (ground truth) annotation for recognized text, and the object classifier detects object label c in the image. Similarly, $freq(c)$ is the number of training images where the object classifier detects object class c.

4.4. Resulting datasets

In this section, we outline in more detail our textual visual context dataset, which is an extension to COCO-text. First, we explain the original dataset and then we describe our proposed textual visual context.

[3] https://www.opensubtitles.org
[4] https://books.google.com/ngrams

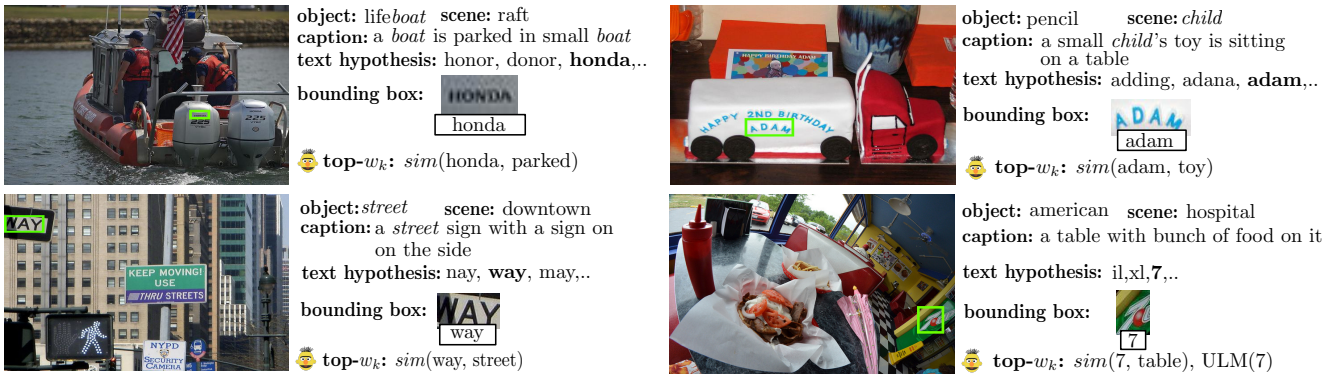

Figure 2. Examples of our proposed dataset. For each bounding box there are list of text hypotheses (w_k) and visual context information, object, place and caption. The **bold** and *italic* font shows the ground-truth and the overlapping visual information, respectively. The top-w_k indicates the top re-ranking score based on Bert [6] similarity or Unigram Language Model (ULM).

Table 3. Data statistic for training dataset that publicly available for caption and text spotting.

Unique Count for Textual Dataset								
Dataset	image #	bbox	caption	object	words	nouns	verb	adjectives
Conceptual [35]	3M	-	3M	-	34219,055	10254,864	1043,385	3263,654
MSCOCO [22]	82k	-	413k	-	3732,339	3401,489	250,761	424977
Flickr 30K [44]	30k	-	160k	-	2604,646	509,459	139128	169158
SVT [42]	350	✓	-	-	10,437	3856	46	666
COCO-Text [40]	66k	✓	-	-	177,547	134,970	770	11,393
COCO-Text-V (ours)	16k	✓	60k	120k	697,335	246,013	35,807	40,922
IC17-V (ours)	10k	✓	25k	50k	296,500	96,371	15,820	15,023
COCO-Pairs (ours)	66k	-	-	158k	319,178	188,295	6,878	46,983

Table 4. Total count of unique words - Dictionary.

Unique Count of Textual Data				
Dictionary	words	nouns	verb	adjectives
Dic-90K [15]	87,629	20,146	6,956	15,534
ULM [33]	8870,209	2695,906	139,385	824,581

4.4.1 COCO-text without visual context

As we described in Section 2.2, the COCO-text dataset is much larger than other text detection and recognition. It consists of 63,686 images, 173,589 text instances (annotations of the images).

4.4.2 COCO-text with visual context

We propose three different visual textual datasets for COCO-text as shown in Table 1: 1) training dataset (COCO-Text-V), 2) benchmark testing (IC17-V) and 3) object and text co-occurrence database (COCO-Pairs).

COCO-Text-V: It consists of 16K images with associated bounding boxes, and 60K textual data, each line have a caption, object and scene visual information. As shown in Table 2, for each bounding box we extract k=10 text hypotheses, and each of them have different or same visual context information depending on the semantic alignment.

ICDAR17-Task3-V (IC17-V) is based on ICDAR17 task 3 end-to-end text recognition dataset. Similar to COCO-Text-V, we only introduce the visual context (textual dataset) for each bounding box. It consists of 10K images with 25k textual data for testing and validation.

To be able to use other type of word embedding, knowledge based embedding, we use external knowledge BabelNet [26] to extract multiple senses for each word. BabelNet[5] is the largest semantic network with a multilingual encyclopedic dictionary, comprising approximately 16 million entries for named entities linked by semantic relations and concepts. Each class label in ResNet has sense or meaning that is extracted from the predefined sense inventory (BabelNet). This allows the model to learn more accurate semantic relations between the spotted text and its visual. That sense ID can be used to extract any word vector from any pre-trained sense embedding [14, 28, 31, 4, 13]. It consists of 1800 images with id senses (*e.g.* orange$^1_{bn:00059249n}$ as *fruit* and orange$^2_{bn:15347402n}$ as *color*) that can be used to compute the similarity vector. Some of the words can be used multiple times because they have only one meaning. For example, an "umbrella" means the same in all contexts; meanwhile, the word "bar" has multiple meanings, such as a steel bar or bar that serves alcoholic beverages.

[5] https://babelnet.org/

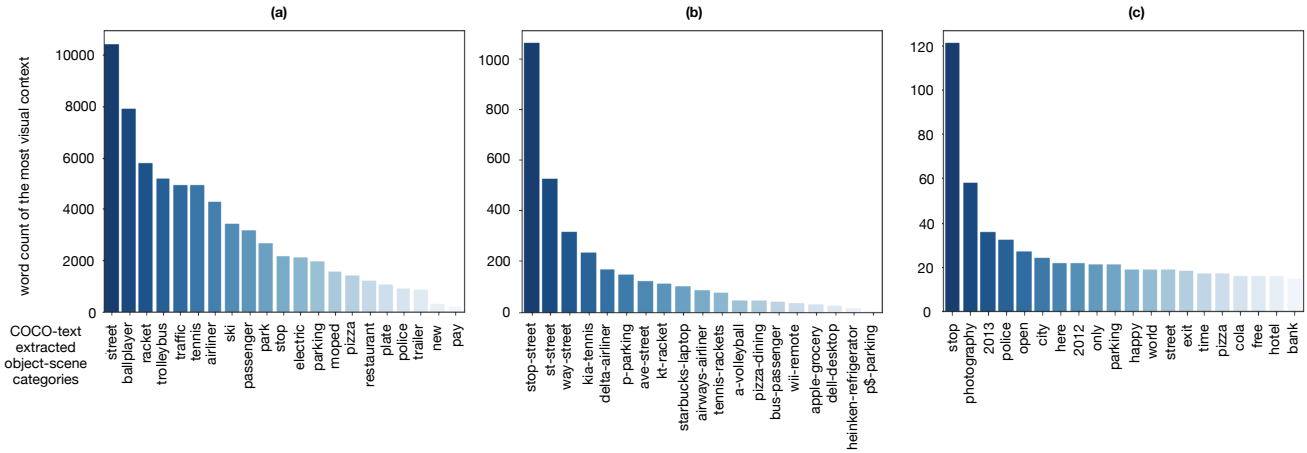

Figure 3. (a) Frequency of objects in COCO-text images. (b) Most common pair (text-object) in the training dataset (c) Frequency count of the most visual context in the testing dataset.

COCO-Pairs: This textual dataset has no bounding box, only the textual information. The dataset consists of only a pair of object-text extracted from each image. It consists of 158K word-visual context pairs. We combined the output from the visual classifier with the ground truth to create the pairs (*e.g.* text-scene, text-object).

Table 3 shows unique word count of part-of-speech tagging (nouns, verb, etc.) of our dataset. Our proposed textual datasets have more semantic than the original COCO-text dataset. Also, as seen in Figure 4 real text in the wild is very challenging problem and thus, current state-of-the-art including our dataset struggle to detect the correct coordination of bounding box. Thus, we use the dataset, COCO-text, ground truth annotation to overcome this shortcoming in this inaccurate bounding box coordination.

5. Experimental Evaluation

5.1. Task

To evaluate the utility of the proposed dataset, we define a novel task, consisting of using the visual context in the image where the text appears to re-rank a list of candidates for the spotted text generated by some pre-existing model.

More specifically, the task is to use different *similarity* or *relatedness* scorers to reorder the k-best hypothesis produced by a trained model with a softmax output. This candidate word re-ranking should filter out false positive and eliminate low frequency short words. The softmax score and the probabilities of the most related elements in the visual context are then combined by simple algebraic multiplication. In this work, we experimented extracting and re-ranking k-best hypotheses for $k = 1 \dots 10$.

5.2. Evaluation remarks

For evaluation, we used a less restrictive protocol than the standard one proposed by [42] and adopted in most state-of-the-art benchmarks, which does not consider words with less than three characters. This protocol was introduced to overcome the false positives on short words that most current state-of-the-art struggle with, including our Baseline. Instead, we consider all cases in the dataset, and words with less than three characters are also evaluated.

Since our task is re-ranking, we use the Mean Reciprocal Rank (MRR) to evaluate the quality of re-ranker outputs. MRR is computed as $\mathrm{MRR} = \frac{1}{|Q|}\sum_{k=1}^{|Q|}\frac{1}{\mathrm{rank}_k}$, where rank k is the position of the first correct answer in the candidate list. MRR is only looking at the rank of the first correct answer; hence it is more suitable in cases such ours, where for each candidate word there is only a single right answer.

Human Evaluation as an Upper Bound. To calibrate the difficulty of the task we picked 33 random pictures from the test dataset and had 16 human subjects try to select the right word among the top $k = 5$ candidates produced by the baseline text spotting system. We observed that human subjects more familiar with ads and commercial logos obtain higher scores. Average human performance was 63% (highest 87%, lowest 39%). Figure 5 shows the user interface for human annotation.

5.3. Baselines

To generate the list of candidate words that will be re-ranked, we rely on two baseline pre-trained systems: a CNN [16] and an LSTM [8]. Each baseline takes a text image bounding box Bb as input and produces k candidate words $w_1 \dots w_k$ plus a probability for each prediction $P(w_i|\mathrm{Bb}) \quad i = 1 \dots k$.

The CNN baseline uses a closed lexicon and can not recognize any word outside its 90K-word dictionary. The LSTM baseline uses a visually soft-attention mechanism which performs unconstrained text recognition without relying on a lexicon.

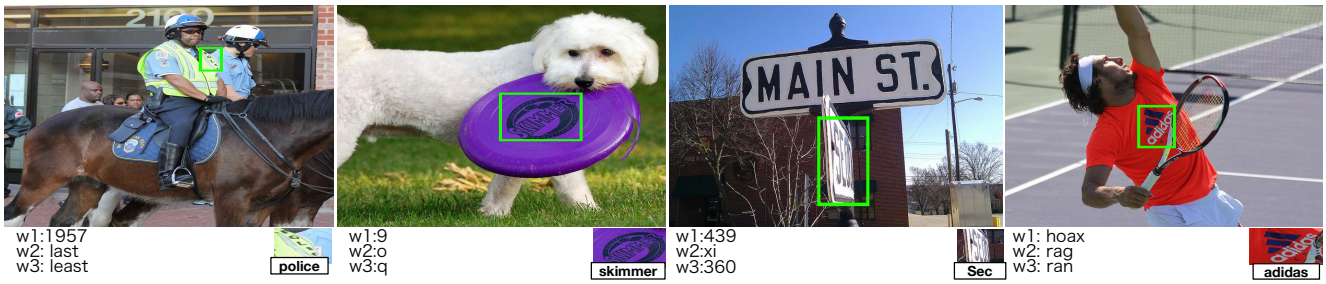

Figure 4. Some random examples extracted from COCO-text with poor detection. The poor detection effect the accuracy of our baseline. Thus, we use the **ground truth annotation** to overcome this shortcoming in this dataset COCO-text.

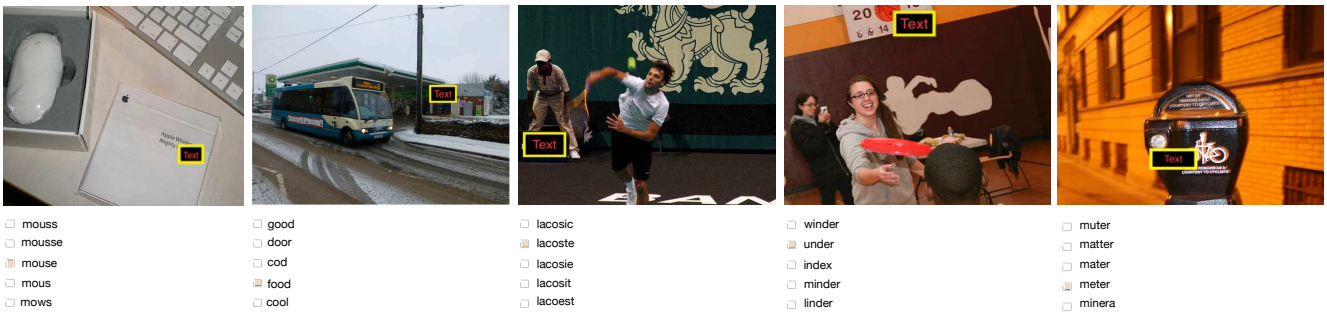

Figure 5. The user interface presented to our human subjects through the survey website asking them to re-rank the text hypothesis based on the visual information. This figure show samples of the variety of images in the wild in COCO-text such as outdoor and indoor images. In this figure, the k=5 text hypothesis has been generated by our baselines and lets the human subject have to choose the most related text to its environmental context.

5.4. Experiments

We performed two experiments, and in each of them we compared the performance of several existing semantic similarity/relatedness systems.

The first experiment consists of re-ranking the text hypotheses produced by the baseline spotting system using only word-to-word similarity metrics. In this experiment each candidate word is compared to objects and places appearing in the image, and re-ranked according to the obtained similarity scores. In the second experiment, we re-rank the candidate words comparing them with an automatically generated caption for the image. For this, we require semantic similarity systems able to produce word-to-sentence or sentence-to-sentence similarity scores.

5.4.1 Experiment 1: Re-ranking using word-to-sentence metrics

We used different off-the-shelf semantic similarity systems to compare the candidate words with the visual context in the image (objects and places), and evaluated the performance of each of them. The used systems are:

Glove [27]: Word embedding system that derives the semantic relationships between words from the co-occurrence matrix. The advantage of Glove over Word2Vec [25] is that it does not rely on local word-context information, but it incorporates global co-occurrence statistics.

Fasttext [17]: Extension of Word2Vec that instead of learning directly the word, it learns a n-gram representation. Thus, it can deal with rare words not seen during training, by breaking them down into character n-grams.

Relational Word Embeddings [3] (RWE): Enhanced version of Word2Vec that encodes complementary relational knowledge into the standard word-embedding in the semantic space. This enhanced embedding is still learned from pure co-occurrence statistics and not relying on any external knowledge. The model intends to capture and combine new knowledge complementary to standard similarity-centric embeddings.

TWE [34]: Semantic Relatedness with Word Embeddings. Word embedding trained using Word2Vec, but instead of general corpus, it is trained on the presented dataset, so it can learn associations between candidate words and their visual context that are uncommon in general text. The model is trained on a *Skip-gram* model [25] that works well with small amounts of training data and is able to represent low-frequency words.

LSTMEmbed [13]: LSTMEmbed is the most recent model in sense embeddings. It utilizes a BiLSTM architecture to learn the word and sense embeddings from annotated corpora. We use the same approach than in [13]: 200-

dimension embeddings trained on the English portion of BabelWiki and English Wikipedia.

Once the similarity between the candidate word and the most closely related element in the visual context is computed, we need to convert that score to a probability in order to combine them in the re-ranking process. Following [34], we use two different methods to obtain the final probability:

- For TWE, we use $P_{TWE}(w|c) = \frac{\tanh(\text{sim}(w,c))+1}{2P(c)}$ where, since $\tanh(x) \in [-1,1]$, then $\tanh(x)+1 \in [0,2]$, and thus $\frac{\tanh(x)+1}{2} \in [0,1]$ is our approximation of $P(w,c)$, which is then divided by $P(c)$ to obtain the conditional probability.
- For all other word-level similarity methods, we combine the obtained cosine similarity $sim(w,c)$, the probability $P(c)$ of the detected context (provided by the object/place classifier), and the probability $P(w)$ of the candidate word (estimated from a 8M token corpus [23]). The final probability is computed following [2] with confirmation assumption $p(w|c) \geqslant p(w)$ as:

$$P(w|c) = P(w)^{\alpha} \quad \text{where } \alpha = \left(\frac{1-sim(w,c)}{1+sim(w,c)}\right)^{1-P(c)}$$

Results of experiment 1 are shown in Table 5-top.

5.4.2 Experiment 2: Re-ranking using word-to-sentence metrics

In the second experiment we used sentence-level semantic similarity. For this, we resorted to state-of-the-art sentence embedding models fine-tuned using the caption dataset.

USE-Transformer [5]: Universal Sentence Encoder (USE) is the current state-of-the-art in Semantic Textual Similarity (STS). The model is based on the transformer architecture USE-T [39] that targets high accuracy at the cost of complexity and resource consumption. We experimented with USE-T fine tuning and feature extraction to compute the semantic relation with cosine distance.

Bert[6] [6]: Bidirectional Encoder Representations from Transformers has shown groundbreaking results in many semantics-related NLP tasks.

Fine-tuned Bert: According to Bert authors, it is not suited for Semantic Textual Similarity (STS) task, since it does not generate a meaningful vector to compute the cosine distance. Thus, we also evaluated a fine-tuned version of the model with one extra layer to compute the semantic score between caption and candidate word. In particular, we fed the sentence representation into a linear layer and a softmax for sentence pair tasks (Q&A re-ranking task).

[6]We use the basic bert-base-uncased model.

Table 5. Experimental results. Row BL shows the baseline performance, without any visual context information. Gray-shaded indicate the models has been trained or fine-tuned using the presented dataset. Star ★ indicate that the model relies on a predefined sense inventory and annotated data. Accuracy (*Acc.*) is the percentage of images in which the right word is ranked in the first place. Column k shows the number of k-best hypotheses re-ranked to obtain the shown accuracy. MRR is computed using $k = 8$ for CNN and $k = 4$ for LSTM.

Model	**CNN**			**LSTM**		
	Acc.	*k*	MRR	*Acc.*	*k*	MRR
Baseline (BL)	**Acc.:19.7**			**Acc.:17.9**		
Experiment 1						
BL+Word2vec [25]	21.8	5	44.3	19.5	4	80.4
BL+ Glove [27]	22.0	7	44.5	19.1	4	78.8
BL+Sw2v [24] ★	21.8	7	44.3	19.4	4	80.1
BL+Fasttext [17]	21.9	7	44.6	19.4	4	80.3
BL+TWE [34]	22.2	7	44.7	19.5	4	80.2
BL+RWE [3]	21.9	7	44.5	19.6	4	80.7
BL+ LSTMmebed [13] ★	21.6	7	44.0	19.2	4	79.6
Experiment 2						
BL+USE-T [5]	22.0	6	44.7	19.2	4	79.5
BL+ BERT-feature [6]	21.7	7	45.0	19.3	4	**81.2**
BL+ BERT (fine-tune) [6]	**22.7**	8	**45.9**	**20.1**	9	79.1

Results for the second experiment are shown in Table 5-bottom. Fine-tuned Bert outperforms all other models. BL+TWE ranks second in accuracy.

6. Conclusions and Further Work

We have proposed a dataset that extends COCO-text with visual context information, that we believe useful for the text spotting problem. In contrast to the most recent method [29] that relies on limited classes of context objects and uses a complex architecture to extract visual information, our approach utilizes out-of-the-box state-of-the-art tools. Therefore, the dataset annotation will be improved in the future as better systems become available. This dataset can be used to leverage semantic relation between image context and candidate texts into text spotting systems, either as post-processing or end-to-end training. We also use our dataset to train/tune an evaluate existing semantic similarity systems when applied to the task of re-ranking text hypothesis produced by a text spotting baseline, showing that it can improve the accuracy of the original baseline between 2 and 3 points. Note that there's a lot of room for improvement up to 7.4 points in a benchmark dataset.

Acknowledgments

This work is supported by the KASP Scholarship Program and by the Spanish government under projects HuMoUR TIN2017-90086-R and María de Maeztu Seal of Excellence MDM-2016-0656.

References

[1] Alessandro Bissacco, Mark Cummins, Yuval Netzer, and Hartmut Neven. Photoocr: Reading text in uncontrolled conditions. In *CVPR*, 2013. 1

[2] Sergey Blok, Douglas Medin, and Daniel Osherson. Probability from similarity. In *AAAI*, 2003. 8

[3] Jose Camacho, Luis Espinosa-Anke, and Steven Schockaert. Relational word embeddings. *arXiv preprint arXiv:1906.01373*, 2019. 7, 8

[4] José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 2016. 5

[5] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018. 8

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5, 8

[7] Shancheng Fang, Hongtao Xie, Zheng-Jun Zha, Nannan Sun, Jianlong Tan, and Yongdong Zhang. Attention and language ensemble for scene text recognition with convolutional sequence modeling. In *ACMMM*, 2018. 1, 2, 3

[8] Suman K Ghosh, Ernest Valveny, and Andrew D Bagdanov. Visual attention models for scene text recognition. *arXiv preprint arXiv:1706.01487*, 2017. 1, 2, 3, 6

[9] Raul Gomez, Baoguang Shi, Lluis Gomez, Lukas Numann, Andreas Veit, Jiri Matas, Serge Belongie, and Dimosthenis Karatzas. Icdar2017 robust reading challenge on coco-text. In *ICDAR*, 2017. 2

[10] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997. 3

[13] Ignacio Iacobacci. Lstmembed: Learning word and sense representations from a large semantically annotated corpus with long short-term memories. In *ACL*, 2019. 5, 7, 8

[14] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Sensembed: Learning sense embeddings for word and relational similarity. In *ACL*, 2015. 5

[15] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. 2, 5

[16] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 2016. 1, 2, 3, 6

[17] Armand Joulin, Edouard Grave, and Piotr Bojanowski Tomas Mikolov. Bag of tricks for efficient text classification. *EACL*, 2017. 7, 8

[18] Chulmoo Kang, Gunhee Kim, and Suk I Yoo. Detection and recognition of text embedded in online images via neural context models. In *AAAI*, 2017. 2

[19] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere de las Heras. Icdar 2013 robust reading competition. In *ICDAR*, 2013. 2

[20] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 3

[21] Christos Liambas and Miltiadis Saratzidis. Autonomous ocr dictating system for blind people. In *GHTC*, 2016. 1

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 4, 5

[23] Pierre Lison and Jörg Tiedemann. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *LREC*, 2016. 4, 8

[24] Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. Embedding words and senses together via joint knowledge-enhanced training. In *CoNLL*, 2017. 8

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013. 1, 4, 7, 8

[26] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 2012. 5

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 4, 7, 8

[28] Mohammad Taher Pilehvar and Nigel Collier. De-conflated semantic representations. *arXiv preprint arXiv:1608.01961*, 2016. 5

[29] Shitala Prasad and Adams Wai Kin Kong. Using object information for spotting text. In *ECCV*, 2018. 2, 8

[30] Satria Priambada and Dwi H Widyantoro. Levensthein distance as a post-process to improve the performance of ocr in written road signs. In *ICIC*, 2017. 1

[31] Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. Automatic construction and evaluation of a large semantically enriched wikipedia. In *IJCAI*, 2016. 5

[32] Arnau Ramisa, Fei Yan, Francesc Moreno-Noguer, and Krystian Mikolajczyk. Breakingnews: Article annotation by image and text processing. *TPAMI*, 2017. 1

[33] Ahmed Sabir, Francesc Moreno, and Lluís Padró. Semantic relatedness based re-ranker for text spotting. In *EMNLP*, 2019. 1, 5

[34] Ahmed Sabir, Francesc Moreno-Noguer, and Lluís Padró. Visual re-ranking with natural language understanding for text spotting. In *ACCV*, 2018. 1, 4, 7, 8

[35] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 5

[36] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *TPAMI*, 2016. 1, 2, 3

[37] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 3

[38] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 3

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 8

[40] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016. 1, 2, 4, 5

[41] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 3, 4

[42] Kai Wang and Serge Belongie. Word spotting in the wild. *Computer Vision–ECCV*, 2010. 2, 5, 6

[43] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 3

[44] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014. 5

[45] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 2017. 3

[46] Anna Zhu, Renwu Gao, and Seiichi Uchida. Could scene context be beneficial for scene text detection? *Pattern Recognition*, 2016. 2