What Does Scene Text Tell Us?

Seiichi Uchida and Yuto Shinahara
Kyushu University, Fukuoka, Japan
uchida@ait.kyushu-u.ac.jp

***Abstract*—Scene text is one of the most important information sources for our daily life because it has particular functions such as disambiguation and navigation. In contrast, ordinary document text has no such function. Consequently, it is natural to have a hypothesis that scene text and document text have different characteristics. This paper tries to prove this hypothesis by semantic analysis of texts by word2vec, which is a neural network model to give a vector representation of each word. By the vector representation, we can have the semantic distributions of scene text and document text in Euclidean space and then determine their semantic categories by simple clustering. Experimental study reveals several differences between scene text and document text. For example, it is found that scene text is a semantic subset of document text and several semantic categories are very specific to scene text.**

I. Introduction

Scene text tells something to us and we fully utilize it for our life. Intuitively, scene text might have a different function from ordinary document text. Some of these particular functions of scene text are *disambiguation* and *navigation*. For example, text on a shop sign disambiguates the type of the shop. Text on a street sign disambiguates our location. The text on a traffic sign suggests where to go. Text on a door suggests suitable operations, such as "push" and "pull." Both of those functions are strongly related to the environmental context around the scene text. In contrast, document text is not expected to have such functions and is not related to any environmental context.

The purpose of this paper is to understand what scene text tells us by analyzing its typical semantic categories. For this purpose we use a large scene text dataset containing 16,517 English words. To the authors' best knowledge, this paper is the first trial to analyze the semantic categories of scene texts. If we know the semantic categories, we can quantify the textual information from scene. In addition to this scientific contribution, we can utilize the quantification result in several applications, especially in scene text recognition. Specifically, the frequency of each semantic category can be used as a general (or even language-independent) prior probability for scene text recognition.

In our semantic analysis of scene text, each word is embedded into an Euclidean space as a vector, while reflecting their meaning. In general, if two words have a similar meaning, their corresponding vectors are similar to each other; for example, *Vec*("avenue") $\sim$ *Vec*("street"). Furthermore, the semantic difference between two words is quantified by their difference vector; we therefore can expect that *Vec*("father")$-$*Vec*("mother") is similar to *Vec*("son")$-$*Vec*("daughter"). This vectorization is realized by word2vec [1], [2], whose mechanism will be reviewed later.

Fig. 1. Examples of scene text images retrieved with the keyword "sign."

Consequently, the vector representation will result in more substantial analysis than a simple word histogram or a wordle[1] representation of the dataset.

Semantic categories of scene texts are finally analyzed by clustering the vectors. Each cluster is a set of words with similar meaning and therefore representing a semantic category. By observing large clusters, it is possible to understand major semantic categories of scene texts. In addition, by comparing the clusters with those of document texts, it is possible to analyze the semantic difference between scene text and document text.

Since the definition of scene text is somewhat vague, we will use scene texts from *sign*. More precisely, the scene texts of our dataset (i.e., the above-mentioned 16,517 words) are collected from 3,000 scenery images retrieved by the keyword "sign". With this keyword-based selection, we can ignore texts in ordinary paper documents captured in scenes. The selected images are similar to the images used in recent competitions on scene text detection and recognition [3], [4], where scene text from signs is the main target. Figure 1 shows several image examples. It is noteworthy that we do not exclude complete sentences from our target, although text on signs is often a short phrase or just a single word.

A. Contributions of This Paper

The main contributions of this paper are summarized as follows:

- This is the first experimental trial to analyze the semantic categories of scene text through a comparison

[1] http://www.wordle.net/

with that of document text.

- It is proved that scene text is a *semantic subset* of document text by combining vector representation by word2vec [1], [2] and clustering.
- It is determined that several semantic categories are very specific to scene text and tend to be more concrete than document text.

II. Related work

A. Scene Text Detection and Recognition

Recently, scene text gets much attention from pattern recognition researchers. The most active task is scene text detection and recognition. It is a similar task of ordinary OCR for document text – however, scene text detection and recognition is much more difficult than document text. This is because scene text undergoes more image distortions, more variations in font shape, and less regularities in layout. Consequently, it is necessary to develop new detection and recognition techniques specialized for scene text [5], [6]. MSER [7] and SWT [8] are common techniques for the text detection task. For the scene text recognition task, Convolutional Neural Networks (CNN) have shown outstanding performance [9], [10], like in other image recognition tasks (e.g., ImageNet Classification [11]).

We can find other research topics around scene text. Visual attention of scene text is a classical topic in visual psychology [12] and also a new topic in pattern recognition [13]. Distribution of foreground-background color contrast in scene text was recently inspected [14]. The relationship between scene text and its environmental context (i.e., its surrounding) is also analyzed and utilized [15]–[17]. To the authors' best knowledge, however, the textual message from scene text, i.e., semantics of scene text, has not been researched yet.

B. Scene Text Dataset

For analyzing the semantic categories of scene text, we need to have a large dataset of scene texts – however, there is no dataset that satisfies this need. One remedy is to use public datasets for scene text recognition, such as [3], [4], [18], but even the largest dataset (ICDAR Robust Reading Competition 2015 [4]) contains only about 4,200 words[2]. IIIT-5K [18] contains 5,000 words but it is comprised of not only scene text but also texts in born-digital images. Consequently, it is necessary to prepare a larger dataset of scene texts for our purpose.

C. Vector Representation of Words

The semantic analysis of scene text can be realized by the recent progress of the vector representation of words (i.e., continuous representation of words). The purpose of vector representation is to embed words into Euclidean space. The simplest realization is so-called 1-of-K representation, where K denotes the vocabulary size. It represents the word of the i-th entry in the vocabulary as a K-dimensional vector where only the i-element is 1 and the others are 0. For realizing a more continuous embedding, i.e., a more semantic embedding, Latent Semantic Analysis (LSA) has been used as a traditional method [19]. LSA is sometimes called a global model because it is based on occurrence of individual words in an entire document.

After the proposal of word2vec [1], [2] in 2013, vector representation has become a hotter and more practical topic than before in natural language processing. The main idea of word2vec is skip-gram, a rather simple neural network model, which will be outlined in Section IV-A. Word2vec is called a local model because it is based on occurrence of individual words in a small window, which can be shorter than a sentence. GloVe [20] is its alternative where a weighing technique originated from the global model is introduced into the local model. Word2vec has been extended to handle a sequence of words, i.e., a paragraph [21] or a document [22].

Word2vec has a drawback in its ability on word sense disambiguation (WSD). For example, the word "plant" can refer to organisms in the kingdom Plantae or can be another word for "factory" and therefore is ambiguous. In this paper, however, we still use the original word2vec implementation, because WSD did not become a serious problem. It will be worthy to replace word2vec with its recent version with better WSD ability, such as [23]–[25], in our future work.

A recent trend around word vectorization is its application to image recognition tasks. The key idea is that the class labels of visual object are a set of words and thus their semantic relationship can be evaluated by using word2vec. The semantic relationship is then utilized for visual object recognition task [26]–[29]. For example, a CNN is trained to output similar vectors for object classes with similar meaning [26]. This is more natural than assuming 1-of-K output and extensible to so-called zero-shot learning.

Gordo et al. [30] have proposed a method which can obtain a semantic vector directly from a scene text image, without any word recognition step. If this method can be more accurate with further improvement (e.g., a combination with query expansion [31]), it is promising for larger-scale semantic analysis of scene text; this is because we do not need to annotate scene text images any more.

III. Dataset of Scene Text from Signs

A. The Original Dataset

Scene text was extracted from 3,000 scene images collected from Flickr [3] under the copyright license of creative commons BY 3.0. As noted in Section I, the keyword "sign" was used for searching Flickr for images containing some text. From those images, texts were manually extracted and gathered as a dataset. The number of words in the dataset was 16,517. To the authors' best knowledge, this is the largest scene text (i.e., word) dataset created manually.

Figure 1 shows several examples of scene text. Most text on signs is a short phrase or just a word, although some are complete sentences. We, therefore, need to analyze semantics

[2]Those words are contained in the training set of "Task 4 (Incidental Scene Text)" of the competition [4]. The test set of this task contains 2,000 words but its word list is not publicly available. The dataset for "Task 2 (Focused Scene Text)" contains much less words than Task 4. Note that the vocabulary size of those 4,200 words was 488 by the screening process of Section III-B. This size is less than half of our dataset (1,102).

[3]www.flickr.com

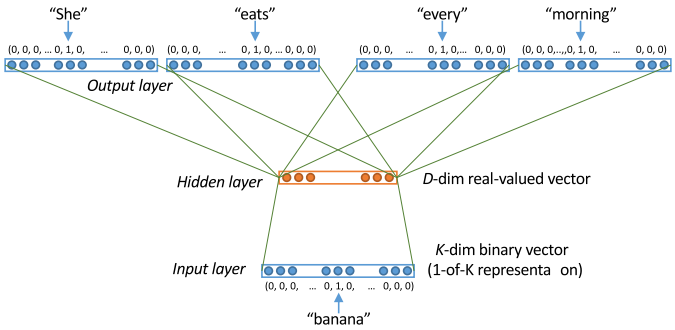

Fig. 2. Skip-gram model in word2vec. Window size $w = 2$.

of scene text in a word-wise manner, rather than a sentence-wise or image-wise manner. Note that an image may contain a part of a word, like "GHLANDS" in Fig. 1. We treat it as a word in the original dataset and then remove it by a screening process before the semantic analysis.

We also use a dataset of document texts in the experimental analysis of Section VI. The role of this dataset is twofold. First, it is used as a training dataset for vectorizing words by word2vec (as described in Section IV-B). Second, it is used for analyzing the semantic characteristics of document text and then clarifying that of scene text via comparison. We use a public dataset *text8*. The dataset is provided along with the source code of word2vec and comprised of ordinary document text. More precisely, *text8* contains English sentences collected from Wikipedia and the number of words is about 16 million.

B. Screening

A screening process was applied to the original scene text dataset for removing words which do not fit to our analysis purpose and then getting the word vocabulary of the dataset. First, if a word is not listed in the GSL (The General Service List) dictionary[4], it is considered as named entity or a part of a word and then removed from the dataset. Similarly, if a word is in the English stop word list[5], it is removed. Then, a stemming operation is applied to remove linguistic variations. (For example, "fishing" and "fishes" will become "fish" by stemming.) PorterStemmer implemented in NLTK[6] was used for the operation. Finally, a "uniquifying" operation is applied to remove duplicated words from the dataset. In Section VI-A, we will observe the result of those screening steps for understanding the characteristics of scene text.

IV. Vectorizing Words by Word2vec

A. Word2vec – A Brief Overview

Word2vec [1], [2] is a neural network-based technique for converting a word into a vector. Assume a sentence with a missing word, "She eats [ ] every morning". We can imagine several candidates for the missing word – "banana", "egg", etc. Simply speaking, the key idea of word2vec is to increase the probability that "banana" and "egg" are represented by similar vectors because they are exchangeable in this *context*, i.e., between the preceding words "She eats" and the succeeding words "every morning". In fact, the probability should be high because they are still exchangeable in many different contexts. The context can be wider; that is, it is possible to use w preceding words and w succeeding words as the context. The parameter w is called *window size*.

In word2vec, the above idea is implemented by the *skip-gram* model[7]. The skip-gram is a three-layer neural network, as shown in Fig. 2. It is similar to an auto-encoder because its hidden layer will output a compressed expression of the input. More precisely, the K-dimensional binary vector representing the target word by the 1-of-K manner is fed to the input layer and then a D-dimensional real-valued vector is generated from the hidden layer. The output layer is expected to generate $2w$ context words as $2w$ K-dimensional binary vectors, whereas the output layer of the conventional auto-encoder is expected to reproduce the input word. After training the skip-gram, the D-dimensional real-valued vector from the hidden layer is treated as a vector representing the semantics of the given input word.

B. Training Word2vec

The original open source code[8] is used as the implementation of word2vec in this paper. By word2vec, each scene word is represented as a 200-dimensional vector (i.e., $D = 200$). For training word2vec, *text8* dataset is used.

It is important to note that we analyze semantics categories of scene text by word2vec trained with not scene text but ordinary document text (i.e., *text8*) because of the following four reasons. (i) Comparison between the semantic categories of scene text and document text becomes easy by using the same word2vec model. (ii) The scene text dataset is not large enough for training word2vec by itself. (iii) Scene text is often a short phrase or just a single word and thus sufficient context (i.e., preceding and succeeding) words are not always available at every word for training skip-gram[9]. (iv) The vocabulary of our scene text dataset is, fortunately, a subset of that of *text8* — this means that we can always have the corresponding vector for any word from our scene text dataset. In other words, we cannot have a vector representation of a word which is not contained in the training dataset.

The window size w on training skip-gram is an important parameter. Bansal et al. [32] pointed out that a small w (say, $w = 1$) makes a pair of words with high syntactical similarity closer and a larger w (say, $w = 10$) makes a pair with high topical similarity closer. We therefore set $w = 5$ as the best compromise for our semantic analysis.

V. Clustering to Obtain Typical Semantic Categories of Scene Texts

To understand the semantic categories of scene text, the 200-dimensional vectors of the words from the screened scene text dataset are clustered by simple k-means. Words of a cluster are expected to share similar semantics. Accordingly, observation of a cluster with many words is helpful to understand

[4] http://www.eapfoundation.com/vocab/general/gsl/frequency/

[5] http://www.ranks.nl/stopwords

[6] http://www.nltk.org/

[7] Word2vec has another model called CBOW but we use skip-gram in this paper.

[8] https://code.google.com/p/word2vec/

[9] The window size w does not matter for vectorizing a word by the trained skip-gram. It does matter in the training step.

TABLE I. CHANGE OF THE NUMBER OF WORDS BY SCREENING STEPS.

	scene text		document text	
original datasets(A)	16,517	(100)	15,943,811	(100)
non-GSL words (B)	7,134	(43.2)	6,291,288	(39.5)
stop words (C)	4,264	(25.8)	6,275,372	(39.4)
after removing non-GSL and stop words (D=A-B-C)	5,119	(31.0)	3,377,151	(21.2)
after stemming an uniquifying (=unique(stemming(D)))	1,102	(6.67)	2,535	(0.016)

The number in parentheses indicates the ratio (%) to the original dataset.

Fig. 3. Example of scene texts (highlighted by a blue box) belonging to the four largest semantic categories.

a typical *semantic category* of scene text. Before clustering, every vector was normalized to have a norm of 1. The metric in k-means was cosine similarity.

The number of clusters, k, is an important parameter for semantic analysis. If k is too small, a cluster will contain words having different semantics. If k is too large, a cluster will contain too few words. Accordingly, semantic category of the cluster becomes ambiguous with inappropriate k. After several preliminary experiments, k was set to 50 as one of the best compromises. When $k = 50$, the maximum, minimum, and average cluster sizes are 63, 5, and 23, respectively, for our scene text dataset.

For each cluster, a label representing the semantic category was attached manually by observing the words belonging to the cluster. For example, the label "Food" is attached for a cluster containing "meal", "carrot", "cake", etc. If we have two clusters of a similar semantic category, we attach a number for each, such as "Food(1)" and "Food(2)." The semantic category of a cluster was sometimes ambiguous or unclear even with the best compromise $k = 50$. The semantic category becomes especially ambiguous when its cluster size was small. In such cases, a special label of "Misc" was used. In fact, it is not always easy to attach a label that covers the meaning of all words in a certain cluster. Nevertheless, manual attachment was still much better than automatic attachment by choosing the word closest to the cluster center as the label.

VI. ANALYSIS RESULTS

A. Observation of Screening Result

Before analyzing the semantic categories, several basic differences between scene text and document text are observed from the result of the screening steps of Section III-B. As mentioned before, about 16 million words in *text8* were used as the original dataset of document texts. Table I shows how each screening step reduces the number of words from the original dataset. From this table, the following facts are confirmed:

- Scene text contains slightly more non-GSL words than document text. In fact, scene text often contains named entity, such as location names and shop names, and word fragments.
- Scene text contains far less stop words than document text. More precisely, 25.8% of scene text are stop words, whereas 39.4% of documents text are stop words. This supports that scene texts are often a short phrase or just a single word.
- The difference between scene text and document text in their word vocabulary size (1,102 and 2,535) is far smaller than the difference in their original dataset size (1.65×10^4 and 1.59×10^7). This suggests that because our scene text dataset has a comparable vocabulary variety to the document text vocabulary, we can analyze the semantic categories of scene text.

B. Semantic Categories of Scene Text

The 1,102 words in the screened scene text dataset were fed to the trained word2vec and then k-means clustering was performed for dividing the resulting vector set into $k = 50$ semantic categories. Table II shows the top 10 largest categories. The category label was determined manually by observing the words as noted in Section V. In the table, the word closest to the cluster center and nine randomly-chosen words are shown as examples of the words from the cluster. Readers can refer to the supplementary material where all the words of each cluster are listed.

Not all but most categories contain words consistent with its label. This indicates that (i) word2vec could give similar vectors to the words with similar meaning successfully, and, more importantly, (ii) scene text has several clear semantic categories. The latter point is more detailed in Section VI-C. Figure 3 shows example images from the top 4 largest categories. Those top categories (except for the first one being comprised of two sub-categories, "Tool" and "Body") contain frequent words in scene.

The distribution of the 50 semantic categories is visualized in Fig. 4. Multi-dimensional scaling (MDS) was used for this two-dimensional visualization of the categories, where the distance between two categories are measured by Euclidean distance between their centroid vectors. The label size is relative to the cluster size. Since individual words with similar meaning have similar vectors, similar semantic categories are also located closely. For example, the categories related to "life" ("Animal/Body", "Animal" and "Tool/Body") are close to each other.

TABLE II. RANDOM EXAMPLES OF WORDS IN TOP 10 LARGEST SEMANTIC CATEGORIES OF SCENE TEXT.

Tool/Body	Behavior	Geography	Transportation	Atmosphere	Public organization	Social/Temporal	Action with object	Food(1)	Misc.
44(4.0%)	41(3.7%)	38(3.5%)	37(3.4%)	37(3.4%)	36(3.3%)	36(3.3%)	32(2.9%)	30(2.7%)	30(2.7%)
circular	begin	bridge	air	belt	act	article	accept	bread	active
finger	follow	canal	busy	close	admission	century	apply	butter	change*
teeth	immediately	island	industrial	deep	appointment	development	deliver	eat	control
weight	join	lake	level	dry*	force	headquarter	develop	food	current
corkscrew*	leave*	mount	lower	extreme	foreign	history*	enjoy	greasy	mark
lock	left	ocean	network	rainy	government*	late	include	juice	open
ear	limit	river*	rail*	side	national	main	provide*	meat	past
scrape	spread	road	railway	wet	official	recent	receive	milk	preference
straight	tray	south	station	wind	police	science	share	roast*	standard
drive	turn	town	upper	winter	president	world	yield	taste	unfair

The top row is the category label. The second row is the number of words belonging to the category with its ratio (percentage) to the whole vocabulary.
The symbol "*" indicates the word closest to the cluster center.

TABLE III. RANDOM EXAMPLES OF WORDS IN TOP 10 LARGEST SEMANTIC CATEGORIES OF DOCUMENT TEXT.

Personality	Communication(1)	Social/Temporal	Behavior	Body condition	Spatial	Judgment	Technology	Art/Feeling	Action with object
122(4.8%)	97(3.8%)	88(3.5%)	83(3.3%)	72(2.9%)	71(2.8%)	68(2.7%)	68(2.7%)	66(2.6%)	65(2.6%)
congratulate	argument	boast	accept	accidental	angle*	cruelly	extra	confidentially	crack
cowardice	common	current	admit	anxiety	edge	deceit	improvement	crowd	draw
curiosity	commonsense	heavily	bring	excess	empty	forbid	level	favorite	dug
dishonor*	detail	influential	build	medical	end	permission	load	goodnight*	float
doubtless	direct	list	entrust*	pain	gap	police	mileage*	grateful	ground
eager	prefer	modern*	give	pressure	log	punish	multiply	joy	hung
foolish	qualification	popular	interfere	reduction	multiplication	refusal	refresh	laugh	scrape*
misery	question	solidly	offer	severe	roughly	suspicion	test	show	stairs
shopkeeper	recommend	total	operate	sleepiness*	row	trial	wrapper	story	sweep
stupidity	sense*	umbrella	succeed	sudden	shape	urgent	yield	theater	touch

The top row is the category label. The second row is the number of words belonging to the category with its ratio (percentage) to the whole vocabulary.
The symbol "*" indicates the word closest to the cluster center.
The underlined words can be found in scene text.

C. Comparison between Scene Text and Document Text

Table III shows the top 10 largest semantic categories of document text. The comparison between Tables II and III reveals the following important semantic characteristics of scene text relative to document text:

- Several semantic categories are very specific to scene texts. Especially, the third ("Geography"), the fourth ("Transportation"), and the ninth ("Food(1)") categories are clear examples.
- Scene text has more *concrete* categories comprised of concrete nouns than document text. This suggests that scene texts need to be concrete for disambiguation and navigation. Especially, words in the category "Geography" are used for disambiguating the location, as shown in Fig. 3. In contrast, document text has more abstract categories (such as "Personality" and "Communication").
- The difference between scene text and document text also appears in the words nearest to the cluster center. Specifically, it is difficult to guess the category label from the center word for document text, while it is rather easier for scene text. This suggests that scene text has a smaller variance in semantic space.

D. Scene Text Is a Semantic Subset of Document Text

Figure 5 shows the two-dimensional visualization of 50 semantic categories of document text. MDS is used for the visualization like Fig. 4, whereas the color of each label represents the percentage of words also found in scene text. A warmer (colder) color indicates that the semantic category contains more (less) words from scene text. The average percentage is about 45% as indicated by the star mark in the figure.

Figure 5 reveals the most important fact:

- Semantic categories of scene text exist locally in the entire semantic distribution of document text. Specifically, semantic categories labeled with warm colors exist only around the center to top-right area in Fig. 5. In other words, scene text is a *semantic subset* of document text.

This is much more meaningful than the case that scene text is just a *vocabulary subset* of document text. The above fact allows us to grasp general prior probability of scene text. For example, any word related to "City/Building" will have higher prior probability than any word related to "Personality" or "Communication."

VII. CONCLUSION

This paper proved several semantic characteristics of scene text, through experimental and comparative analysis using large scene text and ordinary document text datasets. The semantic analysis is done by representing each word as a vector by word2vec [1], [2]. Qualitative evaluations by watching the Euclidean clustering results of the vectors support that word2vec can vectorize the meaning of words appropriately. Main results found by the analysis are as follows: (i) Scene text is a semantic subset of document text. (ii) Several semantic categories such as "Geography" and "Transportation") are very specific to scene text. (iii) Scene text has more concrete semantic categories, which are comprised of concrete nouns, and thus has the functions of disambiguation and navigation.

In the future, we can utilize the prior probability of semantic categories for scene text recognition directly. Furthermore, like [26], we can expect a semantic level combination of words and word images toward a novel formulation of scene text recognition.

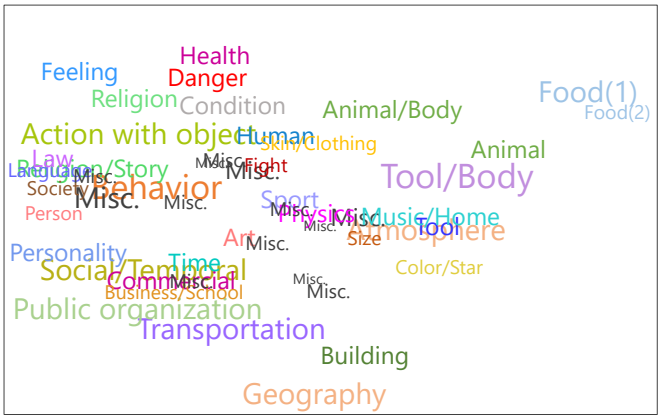

Fig. 4. Two-dimensional visualization of the 50 semantic categories of scene text. Various colors are used just for better visibility.

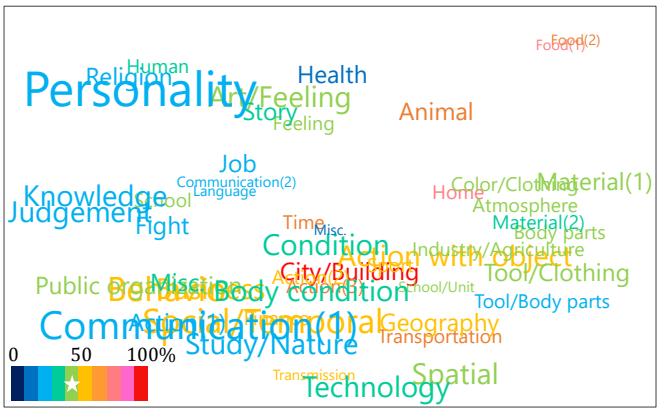

Fig. 5. Two-dimensional visualization of the 50 semantic categories of document text. Colors are used for representing the percentage of words also found in the scene text dataset.

Acknowledgment

This research was partially supported by JST-CREST, MEXT-Japan (Grant No. 26240024), and Kakihara Foundation. The pictures in Fig. 1 and Fig. 3 are taken from Flickr under the copyright license of creative commons “BY 3.0”. The contributors of Fig. 1 are (from left-top to right-bottom): Mykl Roventine, kevinspencer, Fibonacci Blue, Robbie1, phozographer [doing a 365], dno1967b, drcorneilus, rick, Ethan Prater, MelvinSchlubman, josef.stuefer, Daquella manera, adactio, David Boyle, Jonas B, The Last Cookie, caesararum, cogdogblog, lorentey, hsivonen, jack dorsey, acme, LeonIngul, Mykl Roventine, shawnzrossi, shawnzrossi, acme, jo9ce4line0, cogdogblog, and Joel Washing. The contributors of Fig. 3 are (from left-top to right-bottom): upyernoz, acme, Stilgherrian, Mykl Roventine, lincolnlog, cogdogblog, bennylin0724, Johnny Jet, drcorneilus, striatic, Joe Shlabotnik, Joe Shlabotnik, ttarasiuk, taberandrew, TooFarNorth, Mountain/\Ash, Orin Zebest, Jonas B, Oran Viriyincy, and Samuel Mann.

References

[1] T. Mikolov, K. Chen, G. Corrado and J. Dean. Efficient Estimation of Word Representations in Vector Space, *arXiv*, 2013.

[2] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, 2013.

[3] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. Gomez, S. Robles, J. Mas, D. Fernandez-Mota, J. Almazan, and L. -P. de las Heras. ICDAR 2013 Robust Reading Competition. In *ICDAR*, 2013.

[4] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. Ramaseshan Chandrasekhar, S. Lu, F. Shafait, S. Uchida, E. Valveny. ICDAR 2015 Competition on Robust Reading. In *ICDAR*, 2015.

[5] Q. Ye and D. Doermann, Text Detection and Recognition in Imagery: A Survey. *TPAMI*, 2015.

[6] S. Uchida, Text Localization and Recognition in Images and Video. *Handbook of Document Image Processing and Recognition*, Springer-Verlag, 2014.

[7] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *BMVC*, 2002.

[8] B. Epshtein, E. Ofek, and Y. Wexler. Detecting Text in Natural Scenes with Stroke Width Transform. In *CVPR*, 2010.

[9] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet. Multi-Digit Number Recognition from Street View Imagery Using Deep Convolutional Neural Networks. *arXiv*, 2013.

[10] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading Text in the Wild with Convolutional Neural Networks. *IJCV*, 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv*, 2015.

[12] H. -C. Wang and M. Pomplun. The Attraction of Visual Attention to Texts in Real-World Scenes. *J. Vision*, 2012.

[13] R. Gao, S. Uchida, A. Shahab, F. Shafait, and V. Frinken. Visual Saliency Models for Text Detection in Real World. *PLoS ONE*, 2014.

[14] R. Gao, S. Eguchi, and S. Uchida. True Color Distributions of Scene Text and Background. In *ICDAR*, 2015.

[15] Y. Kunishige, Y. Feng and S. Uchida. Scenery Character Detection with Environmental Context. In *ICDAR*, 2013.

[16] V. Frinken, Y. Iwakiri, R. Ishida, K. Fujisaki, and S. Uchida. Improving Point of View Scene Recognition by Considering Textual Data. In *ICPR*, 2014.

[17] Y. Movshovitz-Attias, Q. Yu, M. C. Stumpe, V. Shet, S. Arnoud, and L. Yatziv. Ontological Supervision for Fine Grained Classification of Street View storefronts. In *CVPR*, 2015.

[18] A. Mishra, K. Alahari and C. V. Jawahar. Scene Text Recognition using Higher Order Language Priors. In *BMVC*, 2012.

[19] S. Deerwester, S. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 1990.

[20] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global Vectors for Word Representation. In *EMNLP*, 2014.

[21] Q. V. Le and T. Mikolov. Distributed Representations of Sentences and Documents. In *ICML* 2014.

[22] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From Word Embeddings To Document Distances. In *ICML*, 2015.

[23] X. Chen, Z. Liu, and M. Sun. A Unified Model for Word Sense Representation and Disambiguation. In *EMNLP*, 2014.

[24] I. Iacobacci, M. T. Pilehvar, and R. Navigli. SENSEMBED: Learning Sense Embeddings for Word and Relational Similarity. In *ACL*, 2015.

[25] A. Trask, P. Michalak, and J. Liu, sense2vec - A Fast and Accurate Method for Word Sense Disambiguation In Neural Word Embeddings. *ArXiv*, 2015.

[26] A. Frome, G. Corrado, and J. Shlens. DeViSE: A Deep Visual-Semantic Embedding Model. In *NIPS*, 2013.

[27] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of Output Embeddings for Fine-Grained Image Classification. In *CVPR* 2014.

[28] J. Almazan, A. Gordo, A. Fornes, and E. Valveny. Word Spotting and Recognition with Embedded Attributes. *TPAMI*, 2014.

[29] J. A. Rodriguez-Serrano, A. Gordo, and F. Perronnin. Label Embedding: A Frugal Baseline for Text Recognition. *IJCV*, 2015.

[30] A. Gordo, J. Almazan, N. Murray, and F. Perronnin. LEWIS: Latent Embeddings for Word Images and their Semantics. In *ICCV*, 2015.

[31] P. Krishnan and C. V. Jawahar. Bringing Semantics in Word Image Retrieval. In *ICDAR*, 2013.

[32] M. Bansal, K. Gimpel, and K. Livescu. Tailoring Continuous Word Representations for Dependency Parsing. In *ACL*, 2014.