PSYCHOLOGY AND EDUCATION (2019) 56(1): 33-40

ISSN:1553-6939

BREAKING NEWS ARTICLE ANNOTATION USING IMAGE AND TEXT PROCESSING

B. Bhaskar Reddy[1], P. Imran Khan[2], Dr. B. Dhananjaya[3]
Associate Professor[1,3], Assistant Professor[2]
Department of Electronics and Communication Engineering
Bheema Institute of Technology and Sciences, Adoni-518301.[1,2,3]

ABSTRACT

Building upon recent Deep Neural Network architectures, current approaches lying in the intersection of Computer Vision and Natural Language Processing have achieved unprecedented breakthroughs in tasks like automatic captioning or image retrieval. Most of these learning methods, though, rely on large training sets of images associated with human annotations that specifically describe the visual content. In this paper we propose to go a step further and explore the more complex cases where textual descriptions are loosely related to the images.

We focus on the particular domain of news articles in which the textual content often expresses connotative and ambiguous relations that are only suggested but not directly inferred from images. We introduce an adaptive CNN architecture that shares most of the structure for multiple tasks including source detection, article illustration and geolocation of articles. Deep Canonical Correlation Analysis is deployed for article illustration, and a new loss function based on Great Circle Distance is proposed for geolocation. Furthermore, we present BreakingNews, a novel dataset with approximately 100K news articles including images, text and captions, and enriched with heterogeneous meta-data (such as GPS coordinates and user comments).

We show this dataset to be appropriate to explore all aforementioned problems, for which we provide a baseline performance using various Deep Learning architectures, and different representations of the textual and visual features. We report very promising results and bring to light several limitations of current state-of-the-art in this kind of domain, which we hope will help spur progress in the field.

INTRODUCTION

In recent years, there has been a growing interest in exploring the relation between images and language. Simultaneous progress in the fields of Computer Vision (CV) and Natural Language Processing (NLP) has led to impressive results in learning both image-to-text and text-to-image connections. Tasks such as automatic image captioning [8], [16], [37], [41], [75], [82], image retrieval [21], [31], [45], [48] or image generation from sentences [6], [88] have shown unprecedented results, which claimed to be similar to the performance expected from a three-year old child1. One of the main reasons behind the success of these approaches is the resurgence of deep learning for modeling data, which has been possible due to the development of new parallel computers and GPU architectures and due to the release of new large datasets, used to train deep models with many parameters.

The popularity of crowd sourcing tools has facilitated the proliferation of a number of these datasets combining visual and language content. Among them, most widely known are the UIUC Pascal Sentence Dataset [66], the SBU captioned photo dataset [61], Flickr8K [31], Flickr30K [84] and MS-COCO [51]. All these datasets consist of a number of images (from 1K to 1M), each annotated by human written sentences (between 1 and 5 per image). The annotations are typically short and accurate sentences (of less than 20 words) describing the visual content of the image and the action taking place. In contrast, other and more complex types of documents, like illustrated news articles, have been barely explored. We believe that current successes in the crossroads between NLP and Computer Vision indicate that the techniques are mature for more challenging objectives than those posed by existing datasets.

The NLP community has been addressing tasks such as sentiment analysis, popularity prediction, summarization, source identification or geolocation to name a few that have been relatively little explored in Computer Vision. In this paper we propose several learning schemes. Specifically, for the source detection, article illustration and

geolocation prediction, we consider an adaptive CNN architecture, that shares most of the structure for all the problems, and just requires replacing and retraining the last layers in order to tackle each particular problem. For the

option generation task, image and text representations are combined in a Long Short-Term Network (LSTM). In order to evaluate these algorithms, we have collected BreakingNews, a large-scale dataset of news articles with rich meta-data. Our dataset consists of approximately 100K news articles, illustrated by one to three images and their corresponding captions. Additionally, each article is enriched with other data like related images from Google Images, tags, shallow and deep linguistic features (e.g. parts of speech, semantic topics or outcome of a sentiment analyzer), GPS latitude/longitude coordinates and reader comments. The articles cover the whole year 2014 and are collected from various reference newspapers and media agencies like BBC News, The Guardian or the Washington Post.

This dataset is an excellent benchmark for taking joint vision and language developments a step further. In contrast to existing datasets, the link between images andtextinBreakingNewsisnotasdirect,i.e.,theobjects, actions and attributes of the images may not explicitly appear as words in the text (see examples in Fig. 1). The visual-language connections are more subtle and learning them will require the development of new inference tools able to reason at a higher and more abstract level. Furthermore, besides tackling article illustration or image captioning tasks, the proposed dataset is intended to address new challenges, such as source/media agency detection or estimation of GPS coordinates.

Fig. 1: Breaking News

LITERATURE SURVEY

Tasks Description And Related Work:

We next describe the tasks that will be tackled in this article and the related work for each of them, as well as the existing datasets. Source detection This tasks deals with analyzing the content of the news articles,anddetectinginwhichnewsmediaagencyithas been originally published. It can also be used for more sociological-oriented research. As an example, we tackle the task of quantitatively assessing the intuition that different news agencies have their clear, distinctive style.

This problem can be addressed by modeling the type of language, images and topical preference for each news agency, but also from correctly modeling the sentiment in each news article based on the political orientation of the agency, or other similar refinements. Althoughwearenotawareofotherapproachesexplicitly tackling source identification in news articles, there exists a vast amount of related works, mostly motivated bydetectionofplagiarismordealingwiththeauthorship identification problem [44], [58], [72].

Text Illustration:

This task deals with automatically retrieving the appropriate image (or a small subset) from a pool of images given a textual query of a news story.Some approaches tackle this problem by learning classifiers to represent images through intermediate semantic concepts, that can then be easily assigned to individual keywords or to multi-attribute textual descriptions [3], [4], [45], [48], [67], [17]. Richer textual queries are allowed in [21], [31], [29], [19], [76] by mapping both image and sentences to intermediate spaces where direct comparison is possible. For all these methods, the textual input consists on keywords or short sentences at most. There have been some efforts specifically addressing the domain of news articles.

For instance [22] builds a system based on joint topic models to link text to images, and evaluates the results in the BBC News dataset[23].In[9],itis assumed that there exist short text descriptions and tags accompanying the images, which can then be easily matched to text documents represented by means of word frequencies. Other approaches perform article illustration by exploiting Google's search engine (which also assumes text associated with each image of the database), and combine multiple queries generated from the title of the article [50], or from the narrative keywords [34]. Similarly, [38] proposes a story picturing engine, that first processes a story to detect certain key words,and then uses these to select annotated images from a database. Alternatively to image retrieval strategies, text illustration can be carried out from a generative perspective.

There exist early approaches that extracted object arrangements from sentences to then generate computer graphic representations of static [11] and dynamic [64] scenes. [25] proposed a system to animate a human avatar based on the emotions inferred from text. And very recently, advanced sentence parsers have been used to extract objects and their relations from sentences, and then automatically render 2D [88] and 3D [6] scenes. Finally, illustration of short texts, like chat messages or tweets, has also been investigated [36], [78].

Geolocation :

This task considers the problem of geographically referencing news articles based on text and image cues. One of the pioneering works on a related topic proposed an approach for geolocating web documents by detecting and disambiguating location names in the text [15]. Also only using text information, [68] matched a predefined tagged map to the most likely GPS location of tagged Flickr photos. On the other hand, several image-based methods leverage massive datasets of geotagged images for geolocation of generic scenes on the global Earth scale [28], [40], [79], or for place recognition tasks [7].

DESCRIPTION

Breakingnews Dataset :

We have come a long way since the days when a news articleconsistedsolelyofafewparagraphsoftext:online.articles are illustrated by pictures and even videos, and readers can share their comments on the story in the same web-page, complementing the original document. Studying the effects and interactions of these multiple modalities has clear interesting applications, such as easingtheworkofjournalistbyautomaticallysuggesting pictures from a repository, or determining the best way to promote a given article in order to reach the widest readership. Yet, no benchmarks that capture this multimodality are available for scientific research.

For these reasons, we propose a novel dataset of news articles2 with images, captions, geolocation information and comments, which we will use to evaluate CNN and LSTM architectures on a variety of tasks. Our models are based on the most recent state-or-the-art approaches in deep learning, and thus, this dataset is intended to be a touchstone to explore the current limits of these methodologies, shown to be very effective when dealing with images

Using Image Processing to Detect Text:

Detecting text from images is a prototypical modern puzzle that incorporates image processing, computer vision, and machine learning. Many existing applications do a splendid job in performing this function, such as Google Lens and Cam Scanner. Both of these applications take the next step and implement an optical character recognition (OCR) algorithm to interpret images into actual text.

As part of a bigger project, I wanted to implement such an OCR. While I'm sure there are plenty of libraries that can already accomplish this feat, I needed certain control and customization for my purpose. I additionally wanted to take this opportunity to transition to using Open CV for image processing and computer vision tasks.

Up until this point, I have used Matlabs built in image processing toolbox. The toolbox is splendid and makes image processing projects incredibly easy. Yet, with Open CV being a common choice in industry (and my student Matlablicence expiring soon), I thought it to be beneficial to explore this python library.

Conceptually, finding text is fairly simple. Ideally, a threshold filter can be applied that separates the letters from a contrasting background. After this is achieved, each letter is a blob that can be isolated by finding pixel regions. For my application, I was interested in different types of lettering, not just the letters themselves (bold, italics, underline, etc.).

Similarly to most image processing tasks, the first step in this procedure is to perform a threshold. At first I played around with using Otsu's threshold method, a global threshold technique. I initially thought this would be more simple to code, but later realized OpenCV has fairly convenient functions that allow developers to easily change threshold algorithms.

The two methods produced fairly different results. Arguably, both can be used depending on the desired information. If the whole word is desired, then the blended result of the Otsu method is more useful. If individual letters are needed an adaptive threshold is preferred. Even if a whole word is desired, the word can be determined from individual letters, as will be shown in a future episode. An adaptive filter will also be more convenient if an image is not clean or has poor lighting.

With the image threshold applied, one would think that the letters can be determined. Taking this direct approach creates a very patent glitch. Cavities that occur within letters will be considered letters themselves. This bug occurs due to the way pixel regions are determined in OpenCV. Opposed to Matlab which uses the 'regionprops' command to find pixel regions, OpenCV instead detects contours around regions of a certain pixel value. The command used is 'findContours()' and it's output can be tweaked by the parameters it is given.

In order to avoid this mistake, we need to determine with contours are cavities. Contours need to be pulled along with their hierarchies. We want to distinguish contours by whether they are external or internal. This is done, by setting the second argument in the 'findContours()' function to 'RETR_CCOMP'. The function will now return an array of lists that describe the relationship of each contour with regards to other contours. One of the indices provide the number of the parent of the internal contour or -1 if there is no parent.

With this data, we simply need to ignore any contours that have a parent. Looping through each contour, record their maximum left, right, top, and bottom pixel locations if they passed the mentioned conditional. These coordinates will be used to make the bounding box for each letter.

By applying the above procedure, we get impressively clean results. Letters are isolated and bounded appropriately. The algorithm is actually "too good" in places, which can be seen in locations where the dots above the letter 'i' are interpreted as a single object. Yet the results are not without fault either. This specific font and text editor appears to place the letter 't' and 'y' too close to each other. In the word 'liberty', we can see that 'rty' is considered to be a single letter.

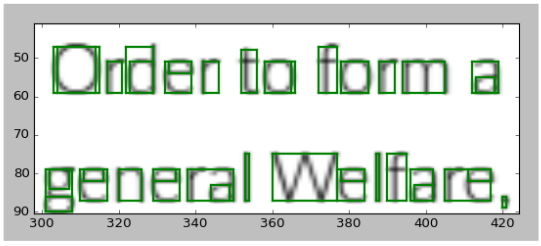
I ultimately want to use this script on handwritten text. The script was tested on two different people's handwriting. One set of handwriting was meant to be very clean, while the other, not so much (I'll leave it to the reader to guess which one is which). This test displayed some areas of trouble that will need to be overcome in the future.

For one, any non-continuous letter will be detected as multiple separate objects. This is most clearly visible in the letter 'T' in the image below. The top of the 'T' is considered to be a separate object from the bottom. The opposite occurs as well. Any script style letters that combine multiple letters together will be interpreted as a single symbol. This occurs with 'Fo' in the word 'For' below. Both of these issues will require some ingenuity to overcome.

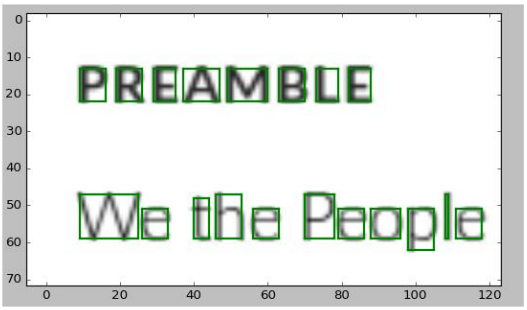
A couple of immediate ideas come to mind. For letters that have been combined, we can look for letters that have a larger bounding box as compared to other letters in its cluster, paragraph, or sentence. If the bounding box is too large, it can suggest that the letter was accidentally combined. From here arises the issue of performing a correct split.

To combine broken letters, we might have to first determine that a bounded region does not resemble an existing letter in our alphabet. This will most likely be determined from some convoluted neural network. After determining an object is not a letter, combination of adjacent bounding boxes will be tested to see if a more clear letter is possible.

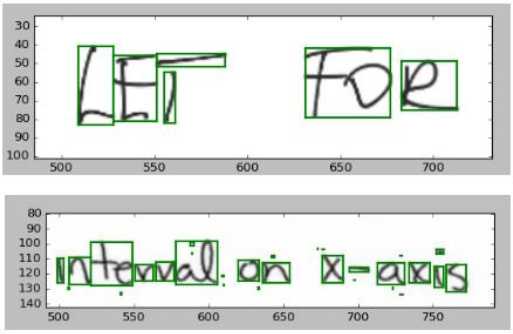
With the script working suitably on normal text, I will be moving on to the next steps of the project in later articles. Some other tasks I want to accomplish is letter recognition, paragraph estimation, and font and style determination.



Cavities in the letters 'g', 'o', 'e', and 'a' are selected as separate objects



Successful letter detection



Different hand writing styles tested

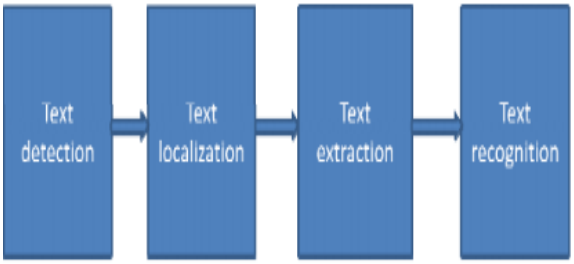


Figure 1 basic block of texture ana lysis

Many of the techniques of digital image processing, or digital picture processing as it often was called, were developed in the 1960s, at Bell Laboratories, the Jet Propulsion Laboratory, Massachusetts Institute of Technology, University of Maryland, and a few other research facilities, with application to satellite imagery, wire-photo standards conversion, medical imaging, videophone, character recognition, and photograph enhancement.[3] The purpose of early image processing was to improve the quality of the image. It was aimed for human beings to improve the visual effect of people. In image processing, the input is a low-quality image, and the output is an image with improved quality. Common image processing include image enhancement, restoration, encoding, and compression. The first successful application was the American Jet Propulsion Laboratory (JPL).

They used image processing techniques such as geometric correction, gradation transformation, noise removal, etc. on the thousands of lunar photos sent back by the Space Detector Ranger 7 in 1964, taking into account the position of the sun and the environment of the moon. The impact of the successful mapping of the moon's surface map by the computer has been a huge success. Later, more complex image processing was performed on the nearly 100,000 photos sent back by the spacecraft, so that the topographic map, color map and panoramic mosaic of the moon were obtained, which achieved extraordinary results and laid a solid foundation for human landing on the moon.

The cost of processing was fairly high, however, with the computing equipment of that era. That changed in the 1970s, when digital image processing proliferated as cheaper computers and dedicated hardware became available. This led to images being processed in real-time, for some dedicated problems such as television standards conversion. As general-purpose computers became faster, they started to take over the role of dedicated hardware for all but the most specialized and computer-intensive operations. With the fast computers and signal processors available in the 2000s, digital image processing has become the most common form of image processing, and is generally used because it is not only the most versatile method, but also the cheapest.

RESULTS AND CONCLUSION

In this section, we present experimental results on the four tasks we considered. We first discuss CNN results for source detection, geolocation prediction and article illustration, followed by a discussion on LSTM and a mixed LSTM/CNNs model for caption generation.Implementation Details Before describing the experimental results, we discuss the technical aspects related to the preparation of the dataset and to the implementation details, including the hyper-parameters set-up for representation and learning. Dataset considerations for the experimental setup: The BreakingNews dataset is split into train, validation and test sets with 60%, 20% and 20% articles respectively. To ensurefairnessintheexperiments,wealsocheckedthere was almost no overlap of images between the sets using the VGG19 features and a cosine distance.

This ratio of near-identical image pairs was in the order of 10−6. Textual representations: In total there are 47,677 unique tokens with frequency 5 or more in the training set (out of 115,427), so the BoW representation is Db = 47,677 dimensional. Regarding the captions, the BoW sizeis13,507(outof89,262uniqueterms).ForWord2Vec, we used the BreakingNews training set with the skipgram method to learn the embedding space. The size of the embedding space was Dw = 500, the window size was 30, the sampling rate was set to 1e-5, and the number of negative samples was set to 5. These hyperparameters were chosen based on the article illustration task for a small subset. We also experimented with a publicly available Word2Vec model trained with the Google 100 billion words dataset, but the performance was worse.Furthermore, we considered usingGlove [63] and Doc2Vec [49] instead of Word2Vec. However, we have found these embeddings to perform worse in our dataset.

For instance, for the illustration task, when trained on a subset with 5,000 pairs and evaluated on a subset with 1,000 pairs, the Word2Vec features have a median rank (MR) of 60 (lower is better), while Glove featureshave100andDoc2Vecfeatures200.Wetherefore focused on Word2Vec features in other tasks. Article analysis: The proposed CNN architecture including our novel GCD and CCA losses were implemented using the Caffe framework [35]. Unless specified otherwise, in the experiments we used a base learning rate of

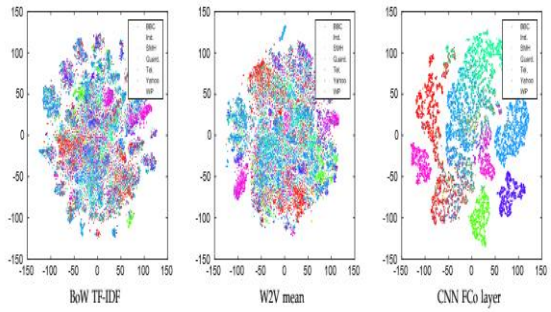


Fig. 4: Source detection: t-SNE embedding of shallow and deep features for the articles in the test set.

0.05, which dropped by a factor of 0.1 after every 1,000 iterations. Stochastic gradient descent was employed as the solver, for which the momentum was set to 0.9. The regularisation parameter weight decay was set to 0.0005. The results reported in this paper were obtained with a batch size of m = 64, which was cross-validated on the validationset.Fortheillustrationtaskthedimensionality of the projection space was also cross-validated, and the cosine distance was used to measure the distance in the projection space. While exhaustive search was not realistic, we experimented as much as possible with network architectures and hyper-parameters of the various layers.

For instance, for the convolutional layer we experimented with size and stride, and found that a kernel width of 5 and a stride of 1 produced the best performance. For the source prediction task, with textual features, using a kernel size of 500×7 instead of 500×5 reduced the accuracy by ∼5 points, and a kernel size of 500×3 instead of 500×5 reduced it by ∼2 points. On the other hand, a stride of 2 and 5 instead of 1 decreased the accuracy by ∼1 and ∼5 points, respectively. All our experiments were performed on an NVIDIA Tesla K40C GPU with 12G memory. It takes approximately 10 hours for training to converge. Depending on the task, the total number of parameters learnt

in the CNN ranges from approximately 0.7M to 1.2M. LSTMs for caption generation: The two architectures in Fig. 3-b1,b2 were implemented using Neuraltalk24.

REFERENCES

[1] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In ICML, 2013.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In International Conference on Computer Vision (ICCV), 2015.

[3] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. Blei, and M. Jordan. Matching words and pictures. The Journal of Machine Learning Research, 3:1107–1135, 2003.

[4] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In ICCV, volume 2, pages 408–415. IEEE, 2001.

[5] L. Cao, J. Yu, J. Luo, and T. Huang. Enhancing semantic and geographic annotation of web images via logistic canonical correlation regression. In ACM International Conference on Multimedia, pages 125–134. ACM, 2009.

[6] A. Chang, M. Savva, and C. Manning. Interactive learning of spatial knowledge for text to 3d scene generation. Sponsor: Idibon, page 14, 2014.

[7] D. Chen, G. Baatz, K. Köser, S. Tsai, R. Vedantham, T. Pylvä, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al. City-scale landmark identification on mobile devices. In CVPR, pages 737– 744. IEEE, 2011.

[8] X. Chen and C. Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In CVPR, 2015.

[9] F. Coelho and C. Ribeiro. Image abstraction in crossmedia retrieval for text illustration. Lecture Notes in Computer Science, 7224 LNCS:329–339, 2012.

[10] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. JMLR, 12(08):2493–2537, 2011.

[11] B. Coyne and R. Sproat. Wordseye: an automatic text-to-scene conversion system. In Conference on Computer Graphics and Interactive Techniques, pages 487–496. ACM, 2001. [12] D. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In International Conference on World Wide Web, pages 761–770. ACM, 2009.

[13] J. Deng, W. Dong, R. Socher, K. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, pages 248– 255, 2009.

[18] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303–338, 2010. [19] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. Platt, C. Zitnick, and G. Zweig. From captions to visual concepts and back. In CVPR, 2015. [20] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, and P. Dollár others. From captions to visual concepts and back. In CVPR, pages 1473–1482, 2015.

[21] A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In ECCV, pages 15–29. Springer, 2010.

[22] Y. Feng and M. Lapata. Topic Models for Image Annotation and Text Illustration. Conference of the North American Chapter of the ACL: Human Language Technologies, (June):831–839, 2010.