Visual Entity Linking

Neha Tilak, Sunil Gandhi
Dept. Of Computer Science & Engineering,
UMBC,
Baltimore, MD
tilak.neha, sunilga1@umbc.edu

Tim Oates
Dept. Of Computer Science & Engineering,
UMBC,
Baltimore, MD
oates@cs.umbc.edu

***Abstract*—Entity linking is the task of identifying entities like people and places in textual data and linking them to corresponding entities in a knowledge base. In this paper we solve a visual equivalent of this task called *visual entity linking*. The goal is to link regions of images to corresponding entities in knowledge bases. Visual entity linking will enable computers to better understand visual content and thus can be used in tasks like image retrieval and visual question answering.**

More specifically, we propose a novel approach for linking image regions to entities in Dbpedia and Freebase. First, we select candidate entities using an automatic image description generation algorithm. We then extract image regions using object detection methods and compare them to depictions of entities in a knowledge base. We evaluate our approach on the Flickr8k dataset through surveys on Amazon Mechanical Turk, and present an extensive analysis to identify the sources of errors in our system.

I. Introduction

Understanding visual content is an important problem in AI. Consider the image shown in Figure 1. We can recognize several parts of the image, such as a suspension bridge, water and land. We can also identify, based on prior knowledge, that the bridge is the Golden Gate bridge. Having identified the Golden Gate bridge, we can infer that the body of water in the image is the San Francisco Bay and that the picture was taken in California. Thus, understanding images not only requires us to identify objects but also to understand relations between them. Having such deep understanding of images could be useful for several tasks, like image retrieval and visual question answering.

Fig. 1. An image of the Golden Gate Bridge.

Traditionally, entity linking is defined as the task of identifying entity mentions in textual data and linking them to corresponding entities in a knowledge base. For example, entity linking over the sentence “The Golden gate Bridge is over the San Francisco Bay” can be linked to two entities ’Dbpedia:Golden_Gate_Bridge’ and ’Dbpedia:San_Francisco_Bay’ and these two entities are related to each other with relation ’dbp:place’. Entity linking in text is a well known and extensively studied problem in the knowledge management community. However, very little attention has been paid to linking image regions to structured knowledge bases in the computer vision and knowledge base communities. The computer vision community has mostly focused on assigning class labels to image regions. Although image classification approaches are intuitive and powerful, they are restricted to a fixed number of classes and do not allow for the rich reasoning required for understanding images like the one in Figure 1.

To overcome this problem, we introduce the visual equivalent of the entity linking problem called *visual entity linking*. The goal of this task is to associate objects in images to entities in structured knowledge bases like DBpedia and Freebase. This task inherently requires us to disambiguate entities using images associated with them. Unlike classification approaches that can perform one specific task, we can query and retrieve for a large array of questions by linking image regions and knowledge base entities. Note that in this paper we link image regions to entities, we leave prediction of relationship between entities in images for future work. A system for linking entities to knowledge bases can support tasks like image retrieval, visual question answering [29], visual verification [19], and finding object affordances [28].

To summarize, the following are the main contributions of this paper:

1) We present a novel system for linking image regions to corresponding entities in structured knowledge bases like DBpedia and Freebase.
2) We evaluate our approach on images in the Flickr8k dataset through surveys on Amazon Mechanical Turk and compare the results with those of a baseline system.
3) We present an extensive analysis to identify the sources of errors in the entity linking system.

The remainder of this paper is organized as follows. Section

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

II discusses related work. Section III introduces background necessary for understanding our system. Section IV describes our system for linking image regions to corresponding entities in knowledge bases, which is evaluated in Section V. Section VII concludes and discusses future work.

II. Related Work

Entity linking and named entity recognition in text are extensively studied problems in the knowledge management community [15] [12]. But entity linking on visual content as proposed in this paper has received relatively less attention.

In the last few years there have been significant advances in image understanding through image classification and object recognition [20] [23]. Most of these approaches use deep neural networks for classifying images into a fixed number of classes. Although a supervised classification approach is intuitive and powerful, it does not allow for rich reasoning and retrieval over images. Another problem that has received significant attention is that of image captioning [26] [10]. Although image captioning is an interesting problem and goes beyond a fixed set of classes, it does not provide information regarding where in the image the object is present. We use image captioning as way of reducing the search space of candidate entities in our visual entity linking model.

Zero shot learning over objects is another research direction that recognizes unseen objects. They do so by performing visual similarity between unseen objects and previously seen objects. [1] [7] [21]. These methods perform well when there is single object in an image as these methods are not applied over image regions.

There are several existing knowledge base construction systems. [17] [4] construct knowledge bases from text. NEIL [5] is another system for extracting common sense relationships from web images. The Visual Genome [11] dataset was recently introduced and provides structured representations of images. Given the existence of such large amounts of structured knowledge, it is important to link entities in these knowledge bases.

Several recent papers have used Markov Logic Networks [18] as the underlying knowledge representation. [28] use MLNs and reason over knowledge to assign object affordances. [19] perform visual verification of relation phrases by searching over visual consistencies among subject, object, and action in images. These papers show the usefulness of reasoning over images and entities.

There are few existing approaches linking entities across text and images. [16] and [27] are closest to our system. [27] links annotated image regions to entities. But their model requires that each segment is classified into one of 275 predetermined categories and annotated with a caption. Also, unlike our system, they do not provide an end to end system for visual entity linking. [16] tries to link objects detected in videos to the entities mentioned in video sub-titles,requiring human annotations for identifying entities in videos.

To the best of our knowledge, ours is the only end to end system that links regions of images to entities in a knowledge base without using external data or annotations.

III. Background

Our visual entity linking model uses existing algorithms that are briefly explained in this section.

Selective Search. Selective Search [24] [25] is a class-independent method of object localization. It produces a large number of bounding box proposals with the objective of enclosing objects in the image. We use these proposals for narrowing down the search space of image regions where objects might be present.

Fast Deepbox. Fast Deepbox [13] is a method that measures the "objectness" of the contents of a bounding box. A bottom-up proposal engine like Selective Search produces a large number of unordered bounding-box proposals. Fast Deepbox re-ranks these proposals based on an objectness score, giving image regions with a higher probability of containing objects a higher rank. Fast Deepbox uses a 4-layer convolutional neural network for processing the bounding box proposals. For this work, we used the publicly available code for Selective Search and Fast Deepbox.

VGGNET for Object Recognition. VGGNET [20] was the winner of the 2014 Imagenet Large Scale Visual Recognition Challenge. The model was trained on the Imagenet dataset [6] and classifies images into one of 1000 categories. We use a pre-trained VGGNET model. For the purposes of our experiments, we remove the last two layers of the VGGNET model. When an image is forwarded through this modified network, we get a 4096-dimensional vector representation of the image. We use these 4096-dimensional vectors to represent image content as these image features are widely used and have shown their utility in many tasks, including object detection [8], image captioning [10], and visual question answering[29]. For the remainder of this paper, we call this vector representation the *VGGNET vector* of the image.

DBpedia and Freebase: Structured Knowledge Bases DBpedia [2] and Freebase [3] are knowledge bases that store structured and unstructured information and are examples of RDF (Resource Description Framework) databases that build on the W3C's linked data technology stack. In this work, we use the SPARQL endpoints provided by DBpedia and Freebase for obtaining data from the knowledge base through SPARQL queries. Many DBpedia entities have relations called 'foaf:depiction' that store an image depicting the entity. Freebase has a similar relation '/common/topic/image' that points to related images. Figure 2 shows an example depiction from DBpedia for the *Tiger* entity. In the remainder of this paper, we refer to these images as "depictions" of the entity and use these depictions as a reference for mapping bounding boxes to entities.

IV. Model

Our visual entity linking system extracts bounding boxes around objects in an image and then links them to entities in a knowledge base using a relevance score.

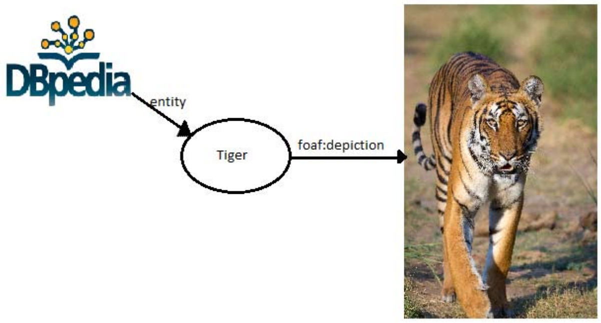


Fig. 2. Entity depictions in DBpedia

One way of linking bounding boxes and entities in a knowledge base is to perform exhaustive search over bounding boxes and entities. However, exhaustive search is computationally expensive because of the large numbers of bounding boxes for an image and entities in a knowledge base. Second, an exhaustive matching approach suffers from errors due to noise in the data as the number of entity candidates increase. We overcome this challenge by reducing the search space to fewer entities by using a caption generation model.

Figure 3 shows the architecture of our system. The key building blocks of our system are the image description generator, entity extractor, knowledge base query client, bounding box generator and entity linker.

As shown in Figure 3, the input image is processed by the Image Description Generator to produce a textual description of the image. The Entity Extractor finds the candidate entities in the generated description. The Knowledge Base Query Client retrieves the depictions of each of the entities from DBpedia or Freebase. The image is also presented to the Bounding Box Generator which produces a number of bounding box proposals. The depictions retrieved from the knowledge base and the bounding box proposals are processed by the Entity Linker that matches the entities with the most suitable bounding box, thereby generating entity-bounding box pairs. All these building blocks are explained in detail below.

A. *Image Description Generation*

The first building block of our model is the Image Description Generator. We used the VGGNET vectors of images and LSTM (Long-Short Term Memory) for generating text captions for the input image. Figure 4 shows the architecture of the image captioning model. This model is based on [26]. It maps the images and sentences into the same vector space and uses LSTM to generate descriptions of images.

To generate image captions, we first generate a VGGNET vector for the image. The resulting 4096-dimensional vector is input to an LSTM. The output is passed through the LSTM and fully connected layer with a number of output nodes equal to the size of the vocabulary. The output layer has a logistic softmax layer that gives the probabilities for each word in the vocabulary. The word with the maximum probability is chosen to be the first word in the sentence. We then provide the index of the first word as input to the network with the goal of predicting the second word of the sentence at the output layer. We keep on doing this until we encounter the end of sentence. The sequence of all output words from the language model is the description of the image. For training the model we follow a similar procedure, but after predicting each word we calculate the negative log likelihood loss and back-propagate the error through the LSTM network.

B. *Entity Extraction*

The purpose of the Entity Extractor is to identify the entities in the image caption and convert them into a format suitable for the knowledge base query engine. The intuition is that the salient entities appearing in the image will also be mentioned in the description. We restrict the entity mentions to nouns and identify them using the Natural Language Toolkit library (NLTK) for Python [14].

The generated description is first processed by the tokenizer, which divides the sentence into a sequence of tokens or words. The part-of-speech tagger classifies the words into lexical categories like nouns, verbs, adjectives, prepositions, and phrases. We select only the noun tokens from this list. The Lemmatizer finds the lemma, or base form, of each word. The last stage of entity extraction is the Formater. It converts the lemmatized words into a camel-cased format which is the standard used in naming entities in DBpedia and Freebase. Thus, if the caption of an image is “A man is standing in grass with a dog”, the entities generated would be “Man”, “Dog”, and “Grass”.

C. *Knowledge Base Query Client*

Once the candidate entities are extracted from the image description, we download the depictions of these entities from Dbpedia or Freebase. Both DBpedia and Freebase are RDF knowledge bases that support SPARQL queries. We use a publicly available SPARQLWrapper python package to fetch a single depiction per entity from DBpedia. If the entity is absent in Dbpedia or does not have a depiction, we fetch its depiction from Freebase. If there are multiple entities with same name, we download depictions of all entities from Dbpedia and Freebase.

D. *Bounding Box Generation*

The second part of our entity linking model is locating the image regions or bounding boxes that enclose objects within them. We generate multiple bounding box proposals using the Selective Search approach. These bounding box proposals are either directly fed to the Entity Linker module or the proposals are ranked and filtered using their “objectness” score before being forwarded to the Entity Linker model.

We use Selective Search as the top-down method for generating bounding box proposals. With an objective of selecting only those boxes that have a high likelihood of containing an object, we use Fast Deepbox to re-rank bounding box proposals. Fast Deepbox measures the “objectness” score

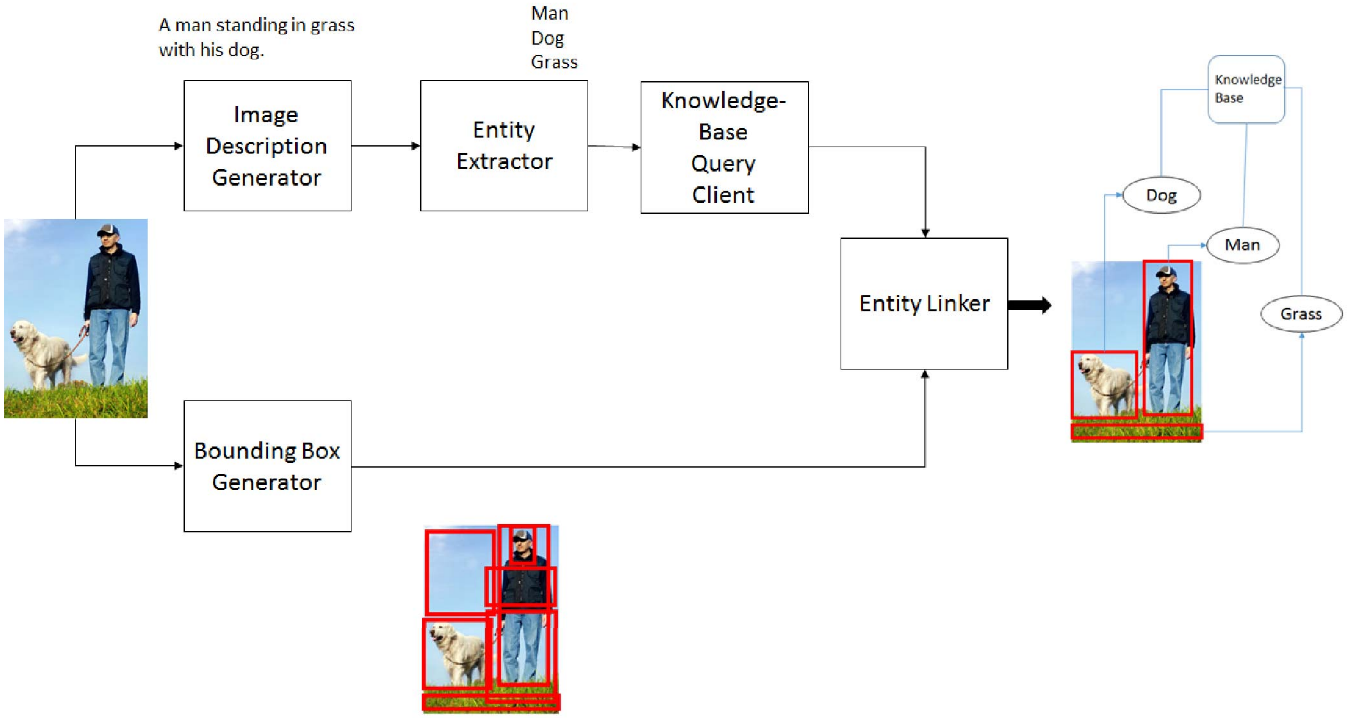

A man standing in grass with his dog.
Man
Dog
Grass
Image Description Generator
Entity Extractor
Knowledge-Base Query Client
Entity Linker
Bounding Box Generator
Knowledge Base
Dog
Man
Grass

Fig. 3. Architecture of our visual entity linking model

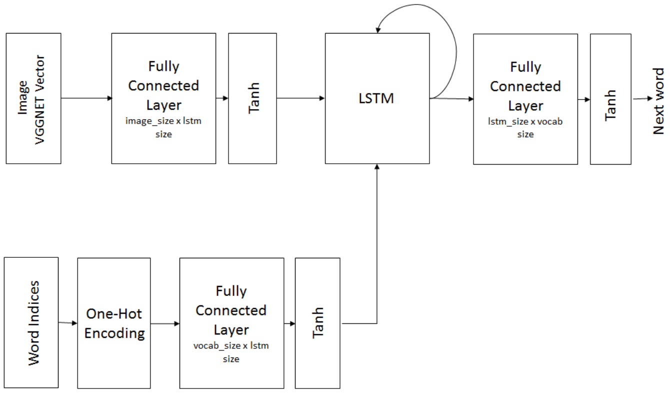


Fig. 4. Architecture diagram of the Image Description Generator

of each bounding box. We select only the top n bounding boxes as candidates for the Entity Linker where n is a user-defined parameter. In this paper, we present our results both with and without Fast Deepbox for bounding box selection. The number of bounding boxes generated by Selective Search is much higher than the combination of Selective Search and Fast Deepbox.

E. Linking Image Regions and Entities

The last stage of the visual entity linking model generates a number of entity-bounding box pairs such that each entity is paired with the bounding box that is most “similar” to its depiction in the knowledge base. In this section we describe the similarity metric used for comparing entity depictions and bounding boxes in images.

The depictions obtained from knowledge bases are generally RGB images. We crop rectangular parts of the test images as per the bounding box dimensions generated from the Bounding Box Generator to obtain an RGB image for each of the bounding box proposals. We then resize both the depiction image and the bounding box proposal image and compute the dot product between their VGGNET vectors.

For each entity, we pick the bounding box for which the dot product was highest and link that bounding box to the entity if the value of the dot product is over a certain threshold. Here, the threshold is a hyper-parameter, and we experiment with different values of dot product thresholds and present our results.

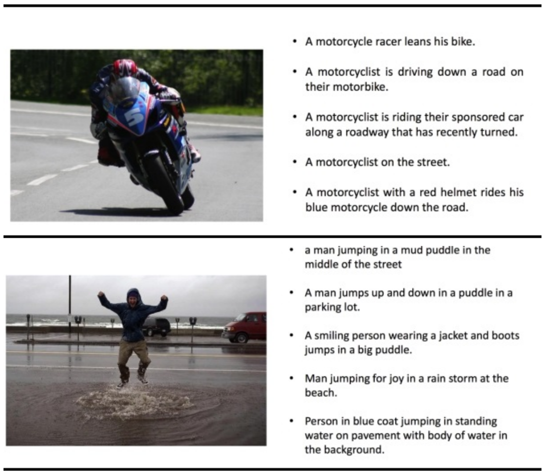
To summarize, we use cues from the image description model to derive candidate entities. We use bounding box proposals from Selective Search and Fast DeepBox as candidates for image regions. Lastly, we use VGGNET, a deep convolutional network, to match entities to bounding boxes using dot product similarity.

V. Experimental Setup

In this section we describe our experimental setup and then provide detailed evaluation of our results.

A. Dataset

We used the Flickr8k dataset [9] for training our image description generation model. Flickr8k is a collection of 8000 images from the Flickr website. Each of the images is paired with five captions that describe the image. The annotations were generated by people on Amazon Mechanical Turk (AMT). Figure 5 shows examples of two images and annotations from the dataset.



- A motorcycle racer leans his bike.
- A motorcyclist is driving down a road on their motorbike.
- A motorcyclist is riding their sponsored car along a roadway that has recently turned.
- A motorcyclist on the street.
- A motorcyclist with a red helmet rides his blue motorcycle down the road.

- a man jumping in a mud puddle in the middle of the street
- A man jumps up and down in a puddle in a parking lot.
- A smiling person wearing a jacket and boots jumps in a big puddle.
- Man jumping for joy in a rain storm at the beach.
- Person in blue coat jumping in standing water on pavement with body of water in the background.

Fig. 5. Example of images and annotations from Flickr8k dataset

We used 6000 images from the Flickr8k dataset for training the image description generation module. We trained the model for 20 iterations over the entire training set with a learning rate of 0.02, a learning-rate decay of 0.95, and 50% dropout [22]. The trained model is used for generating image descriptions. We used 200 images from the test set for evaluation of our model.

B. Evaluation using Amazon Mechanical Turk

Our system uses cues from the description generation model to produce a list of candidate entities. It uses Selective Search and Fast Deepbox for obtaining bounding box proposals. Lastly, it generates pairs of entities and bounding boxes using dot product similarity.

We want our system to produce as many entity mappings as possible while ensuring that each box is linked to the right entity and that each bounding box contains only the relevant object with as few surrounding objects as possible. We not only want to understand how good the entity mappings are but also the sources of error in the entity mapping system. We evaluate our model in terms of the following:

1) The factual correctness of the descriptions obtained from the image description generator. Image captions generated need not, and cannot, describe every detail in the image. But what they do describe should be correct with reference to the images.
2) The accuracy of linking a bounding box to the right entity. Of all the pairs of bounding boxes and entities generated by our model, we are interested in knowing the number of pairs that are accurate. Thus, we define the percentage of entity-bounding box pairs that are correct as *mapping accuracy* (MA).
3) The tightness of the bounding box around the object of interest. A good bounding box will fit the labelled object only and have a minimum of extraneous space around the object. This is important for two reasons. First, bad bounding boxes may result in lower accuracy for entity-bounding box pairing. Second, an algorithm can generate bounding boxes containing complete images every time, thus ensuring that any given object is inside the bounding box and achieving high mapping accuracy. But such a bounding box is not useful.

Since we do not have ground truth data for these parameters in the Flickr8k dataset, we evaluate the entity linking results by asking human annotators to rate image descriptions and entity mappings. We use Amazon Mechanical Turk (AMT) for crowd sourcing the evaluation. We evaluate each entity-bounding box pair individually. For example, if we obtain three bounding boxes linked to three different entities for an image, we present one pair to the annotator at a time and ask them to rate the correctness of the entity mapping. Thus, for three mappings, we present three questionnaires. Along with the entity mapping, we also ask the AMT workers to rate the image captions generated by the image description generator. Each of these questionnaires (AMT HITs) is given to four different AMT workers to ensure quality of the responses. We average the results of AMT workers for all experiments.

Each HIT contains an image with one bounding box drawn over it. The box is labelled with the name of a DBpedia entity. Below the image, we display the description generated by the image captioning module. We ask three questions of the AMT workers about the image and box label. The questions and their choices of answers are as follows:

1) Is the description correct with respect to the image ? Ignore grammatical errors.
(a) Yes (b) No (c) Partially right
2) Is the label associated with the box closely related to an object enclosed in the box?
(a) Yes (b) No
3) Roughly speaking, what percentage of the box is occupied by the object denoted by the label, if any?
(a) less than 25% (b) 25% to 50%
(c) 51% to 75% (d) more than 75%

In the remainder of this section, we present the results obtained from the AMT survey.

C. Creating a Baseline

To the best of our knowledge, ours is the only work that creates an end-to-end system to link entities from images to knowledge bases. Thus we created a simple baseline system.

The baseline system produces a list of candidate entities using the same method as described in sections IV-A and IV-B then generates the same number of bounding box proposals as the entities using a combination of Selective Search and Fast Deepbox. Finally, it randomly assigns each of the *n* candidate entities to one of the top *n* bounding boxes obtained from Deepbox. Here, it does not use any similarity measure or entity

Similarity Threshold	Number of Entity-Box Pairs Extracted
0	519
1400	465
2000	261
Baseline	519

TABLE I
NUMBER OF ENTITY-BOUNDING BOX PAIRS EXTRACTED FROM 200 TEST IMAGES

depictions for linking an entity to a bounding box. We present these mappings to AMT workers and use the results as a comparison.

VI. RESULTS

We performed two main categories of experiments. First, we observed the effect of the dot product similarity threshold value on the resultant entity mappings. Second, we fixed the threshold value and observed the effect of using Fast Deepbox on the results.

Since candidate entities are extracted from image captions, we also evaluated the quality of image descriptions generated by the caption generation module. For all test images, 55.6% of the descriptions were factually right or partially right. We believe that with more training data, our captioning will produce correct descriptions more often. Here, we present results for bounding box quality and mapping accuracy for images with correct as well as incorrect captions. This gives us some insight regarding the source of errors in our entity linking system.

A. Threshold for dot product similarity

We link a bounding box to an entity only if the dot product similarity between the bounding box vector and the depiction vector is more than a threshold value. We experimented with different values of threshold and observed the effect on the results.

Table I shows the number of entities extracted from the test images for different levels of threshold. It can be seen that as the threshold value is increased, some pairs of entities and bounding boxes are rejected by the model.

Table II shows the mapping accuracy of the model. The values are given for different levels of threshold. The second column shows mapping accuracy across all test images, while the third column show mapping accuracy given that the generated image caption is correct. The last row of the table shows the mapping accuracy for our baseline system. It can be seen that our model performs better than the baseline for all three values of dot product threshold. It is also interesting to note that our model gives a higher mapping accuracy with a higher value of threshold as compared to zero threshold. This is because raising the threshold allows our system to reject abstract or non visual entities.

Table III shows the quality of bounding boxes obtained when the box is linked to the correct entity. As explained in previous section, annotators were asked to determined what percentage of the bounding box is occupied by the object. The values shown in the table are the percentage of boxes that fall into one of the four response categories: 0-25%, 26-50%, 51-75% and 76-100%. The values are noted for different dot product thresholds. It can be seen that when our model is used with a dot product threshold of 2000, around 56% of the bounding boxes were linked to entities that occupied at least half of the box by the labelled object. This percentage is greater than the ones obtained with a threshold of 1400 and 0. It can also be seen that our model gives better quality boxes than the baseline.

Similarity Threshold	All Test Images	Test Images with Correct Captions
0	48.65%	68.46%
1400	54.78%	75.24%
2000	55.84%	75.51%
Baseline	46.38%	57.67%

TABLE II
PERCENTAGE OF BOUNDING BOXES MAPPED TO CORRECT ENTITIES

Similarity Threshold	0-25%	26-50%	51-75%	76-100%
0	24.26%	21.11%	19.54%	35.09%
1400	34.64%	20.41%	15.21%	29.74%
2000	21.61%	22.30%	24.19%	31.90%
Baseline	34.69%	27.41%	20.04%	17.86%

TABLE III
QUALITY OF BOUNDING BOXES

By looking at the table II, it can be said that although the baseline performs poorly compared to our model in terms of mapping accuracy, it still gives a mapping accuracy of 57.67% when the captions are correct. This happens primarily because of two reasons. First, there are several images that contain only one entity and randomly assigning the entity to ”most object like” bounding box does not decrease accuracy. Second, baseline prefers boxes that occupy larger areas of image. This is evidenced by fact that 62% of the bounding boxes linked to entities in the baseline were such that the labelled object occupied less than half of the box area. Thus the baseline suffers more in terms of bounding box quality than in mapping accuracy.

B. Deepbox for sorting bounding boxes based on “objectness” score

Fast Deepbox sorts the bounding box proposals in the order of the “objectness” measure of the boxes. Consequently, it removes some of the bounding box candidates that were otherwise linked correctly or incorrectly to an entity when Selective Search alone was used. In this section, we explore the effect of using Fast Deepbox in combination with Selective Search on the performance of our model. All the results listed below are obtained by setting the threshold value of dot product similarity to 2000.

We extracted 261 entity and bounding box pairs using just selective search and 202 using a combination of Selective

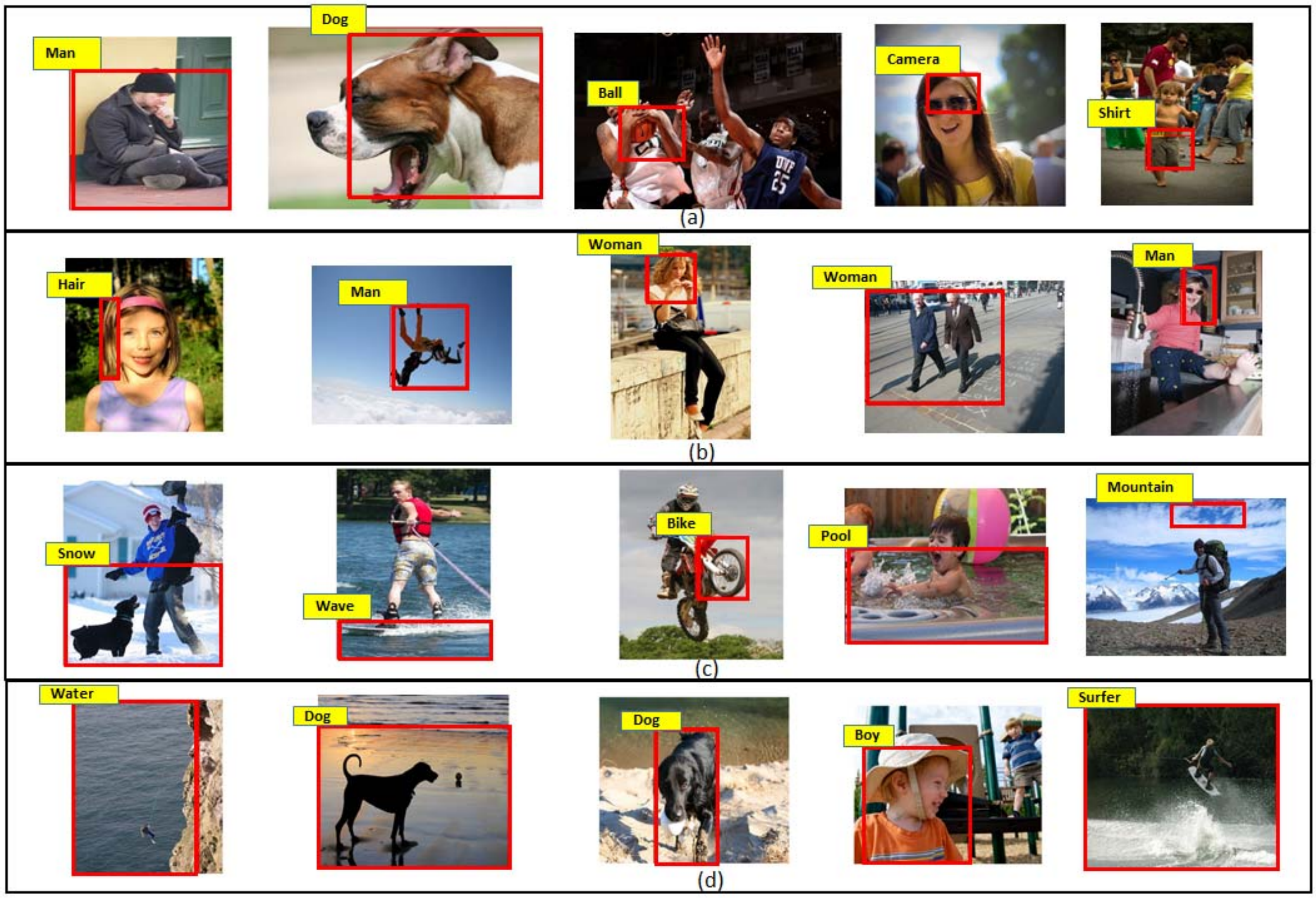


Fig. 6. Example outputs of the model for following configurations: (a) Selective Search is used with a zero threshold value for the dot product similarity. (b) Selective Search is used with a threshold value of 1400 for the dot product similarity. (c) Selective Search is used with a threshold value of 2000 for dot product similarity. (d) A combination of Selective Search and Fast Deepbox is used with a threshold value of 2000 for the dot product similarity.

	Selective Search	**Selective Search** **+** **Fast Deepbox**
All test images	55.84%	66.21%
Test images with correct caption	75.51%	80.31%

TABLE IV
Mapping Accuracy with Fast Deepbox and without it.

Method	**0-25%**	**26-50%**	**51-75%**	**76-100%**
Selective Search	21.61%	22.30%	24.19%	31.90%
Selective Search + Deepbox	38.13%	23.92%	23.92%	14.03%

TABLE V
Quality of Bounding-boxes with Fast DeepBox

Search and Fast Deepbox. It can be seen that fewer pairs of entities and bounding boxes were extracted from the images by using Fast Deepbox as compared to those obtained by using Selective Search alone. It is important to evaluate how the mapping accuracy was affected due to Fast Deepbox. Table IV shows the mapping accuracy of the model. In case of correctly captioned images, fewer entity-bounding box pairs were extracted with Fast Deepbox, but the mapping accuracy was 80.31% as compared to 75.51% with just Selective Search. However, both techniques suffer when the generated captions are incorrect.

Lastly, we evaluate the quality of bounding-boxes when a combination of Fast Deepbox and Selective Search was used. Table V shows the distribution of the bounding boxes over the four response categories of bounding box occupancy. It can be seen that our model suffers slightly in terms of bounding box quality when Fast Deepbox is used as compared to when only Selective Search is used. This is because we observe that Fast Deepbox prefers larger bounding boxes.

C. Example Results

Figure 6 shows some example results of our model with different configurations. When we used selective search for producing bounding box proposals with a zero threshold for entity linking, we observed that some of the bounding boxes were linked to incorrect entities as seen in figure 12a. When we performed this experiment with a higher dot-product threshold of 1400, the number of incorrectly linked bounding boxes went down. This can be seen in figure 12b. Thus, dot-product threshold helped in reducing the mis-linking of bounding boxes. The entity linking errors further reduced with a threshold of 2000 as seen in figure 12c.

When a combination of Fast Deepbox and Selective Search was used with a similarity threshold of 2000, we observed that the model produces comparable results to the previous method. However, it is observed that this model prefers larger bounding boxes for entity linking as can be seen in figure 12d. This reduces the bounding box quality for this method, which explains the results in table V.

VII. Conclusion

In this paper, we presented a novel approach of linking bounding boxes in images and entities in a knowledge-base. We made use of deep neural networks for deriving candidate entities from images and Selective Search and Fast Deepbox for obtaining bounding-box proposals. Finally, we used convolutional neural networks for linking entities to bounding-boxes based on dot-product similarity. We also developed a baseline system for comparing our results. We evaluated our results by getting human annotations from Amazon Mechanical Turk and found that our method performs significantly better than the baseline in terms of entity-linking accuracy and tightness of bounding-boxes.

In the future, we would like to train our model over a bigger image dataset. We believe that with more training data, better image captions would be generated, ultimately improving our entity-linking accuracy. We would also like to experiment with different bounding-box proposal methods like Edge-Box. In future we intend to write a query engine over images using our method for visual entity linking.

An interesting application of our method could be in augmented reality devices like Google Glass and Hololens, where our model can display additional information about objects that the user is seeing. We would like to explore that application in future.

References

[1] E. Bart and S. Ullman. Single-example learning of novel classes using representation by similarity. In *BMVC*, volume 1, page 2, 2005.

[2] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, Sept. 2009.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. ACM.

[4] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3, 2010.

[5] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1409–1416, 2013.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[7] M. Fink. Object classification from a single example utilizing class relevance metrics. *Advances in Neural Information Processing Systems*, 17:449–456, 2005.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 38(1):142–158, 2016.

[9] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Int. Res.*, 47(1):853–899, May 2013.

[10] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*, 2014.

[11] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016.

[12] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466. ACM, 2009.

[13] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. *CoRR*, abs/1505.02146, 2015.

[14] E. Loper and S. Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[15] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.

[16] I. Naim, Y. C. Song, Q. Liu, H. A. Kautz, J. Luo, and D. Gildea. Unsupervised alignment of natural language instructions with video segments. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 1558–1564, 2014.

[17] F. Niu, C. Zhang, C. Ré, and J. Shavlik. Elementary: Large-scale knowledge-base construction via machine learning and statistical inference. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(3):42–73, 2012.

[18] M. Richardson and P. Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[19] F. Sadeghi, S. K. Divvala, and A. Farhadi. Viske: Visual knowledge extraction and question answering by visual verification of relation phrases. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 1456–1464. IEEE, 2015.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[21] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, Jan. 2014.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[24] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[25] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *IEEE International Conference on Computer Vision*, 2011.

[26] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*, 2014.

[27] R. Weegar, K. Astrom, and P. Nugues. Linking entities across images and text. 2015.

[28] Y. Zhu, A. Fathi, and L. Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *Computer Vision–ECCV 2014*, pages 408–424. Springer, 2014.

[29] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. *arXiv preprint arXiv:1511.03416*, 2015.