Dynamic Lexicon Generation for Natural Scene Images

IIIT Hyderabad

Team No - 28
Mentor -- Sarthak Sharma

Team Members:
Bhavin Kotak - 2018201071
Ishan Tyagi - 2018201017
Jatin Paliwal - 2018202006
Vishal Bidawatka - 2018201004

1 Introduction

Many scene text understanding methods approach the end-to-end recognition problem from a word-spotting perspective and take huge benefit from using small per-image lexicons. Such customized lexicons are normally assumed as given and their source is rarely discussed. In this report we mention a method that generates contextualized lexicons for scene images using only visual information. For this, we exploit the correlation between visual and textual information in a dataset consisting of images and textual content associated with them. Using the topic modeling framework to discover a set of latent topics in such a dataset allows us to re-rank a fixed dictionary in a way that prioritizes the words that are more likely to appear in a given image. Moreover, we train a CNN that is able to reproduce those word rankings but using only the image raw pixels as input. The quality of the automatically obtained custom lexicons is superior to a generic frequency-based baseline.

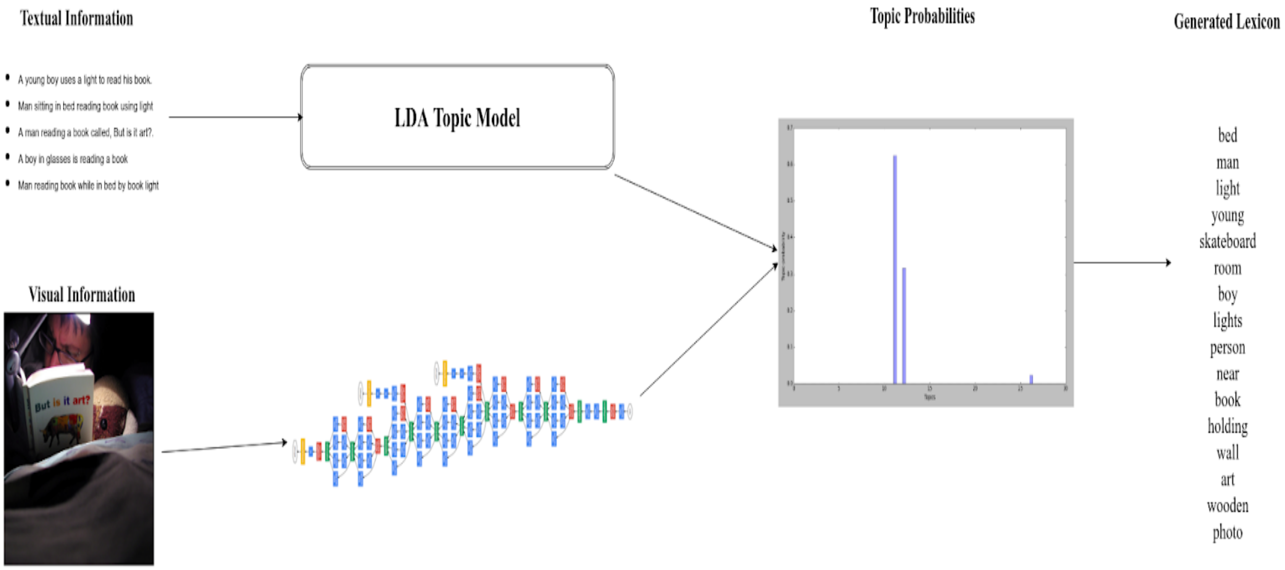
Scene text problem is to understand the scene better with the help of sign board or text which appear in scene.
1.This is related to the problem of Optical Character Recognition (OCR). Scene text is more difficult due to large variability in appearances than OCR methods.

2.One could, for instance, foresee an application to answer questions such as, “What does this sign say?”.
3.Many scene understanding methods recognize objects and regions like roads, trees, sky in the image successfully, but tend to ignore the text on the sign board. Our goal is to fill this gap in understanding the scene.
End-to-end scene text recognition pipelines are usually based in a multi-stage approach, first applying a text detection algorithm to the input image and then recognizing the text present in the cropped bounding boxes provided by the detector.
Scene text recognition from pre-segmented text has been approached in two different conditions: using a small provided lexicon per image (also known as the word spotting task), or performing unconstrained text recognition, i.e. allowing the recognition of out-of-dictionary words.
Many of the existing text recognition methods rely on individual character segmentation and recognition. After that, character candidates are grouped into larger sequences (words and text lines) using spatial and lexicon- based constraints.

2. Method

The underlying idea of our lexicon generation method is that the topic modeling statistical framework can be used to predict a ranking of the most probable words that may appear in a given image. For this we used a three-fold method:
First, we learn a LDA topic model on a text corpus associated with the image dataset.
Second, we train a deep CNN model to generate LDA’s topic-probabilities directly from the image pixels.
Third, we use the generated topic-probabilities, either from the LDA model (using textual information) or from the CNN (using image pixels), along with the word-probabilities from the learned LDA model to rerank the words of a given dictionary.

Below given image shows diagramatic representation of our model:


Textual Information
A young boy uses a light to read his book.
Man sitting in bed reading book using light
A man reading a book called, But is it art?.
A boy in glasses is reading a book
Man reading book while in bed by book light
LDA Topic Model
Visual Information
But is it art?
Topic Probabilities
Generated Lexicon
bed
man
light
young
skateboard
room
boy
lights
person
near
book
holding
wall
art
wooden
photo

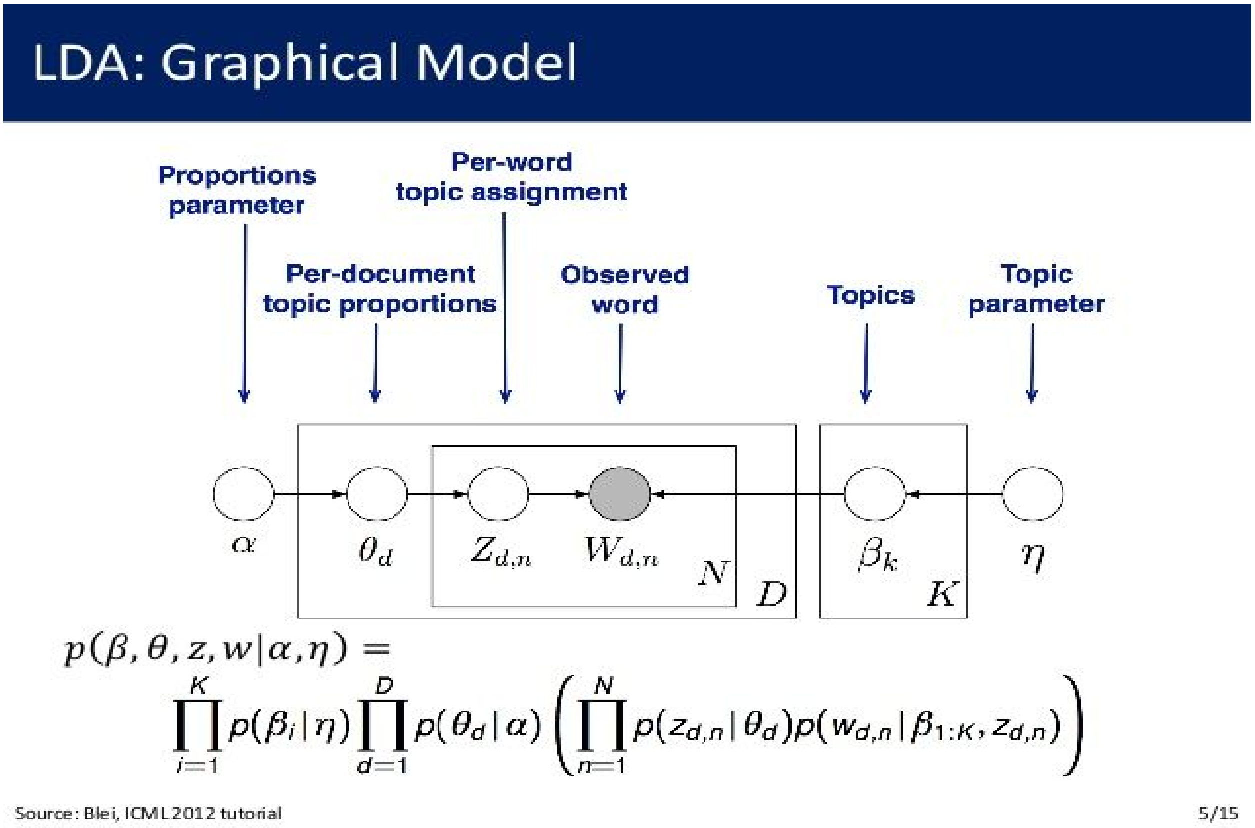
2.1 Learning the LDA topic model using Textual Information

What is a Topic Model?

In machine learning and natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. It is a frequently used text-mining tool for discovery of hidden semantic structures in a text body.

What is LDA ?

In natural language processing, **latent Dirichlet allocation** (**LDA**) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.
LDA topic model to discover latent topics from training data by using only the textual information
For example, if observations are words collected into documents, that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. LDA is an example of a topic model.



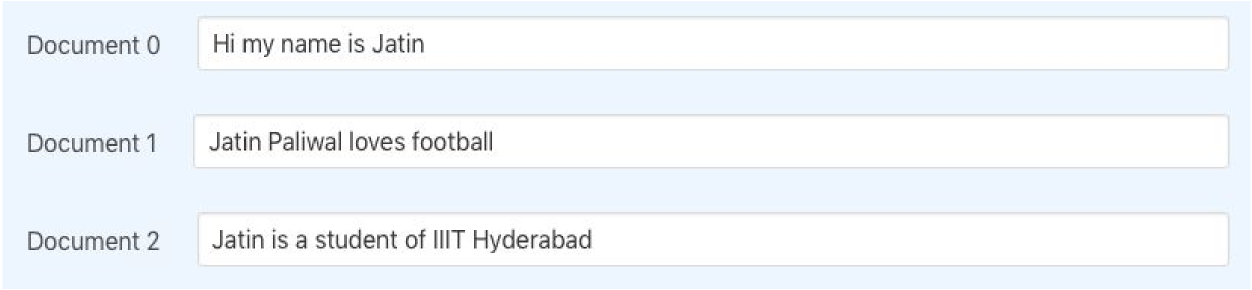
LDA: Graphical Model

$$p(\beta,\theta,z,w|\alpha,\eta) = \prod_{i=1}^{K} p(\beta_i|\eta)\prod_{d=1}^{D} p(\theta_d|\alpha)\left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d)p(w_{d,n}|\beta_{1:K},z_{d,n})\right)$$

Source: Blei, ICML 2012 tutorial

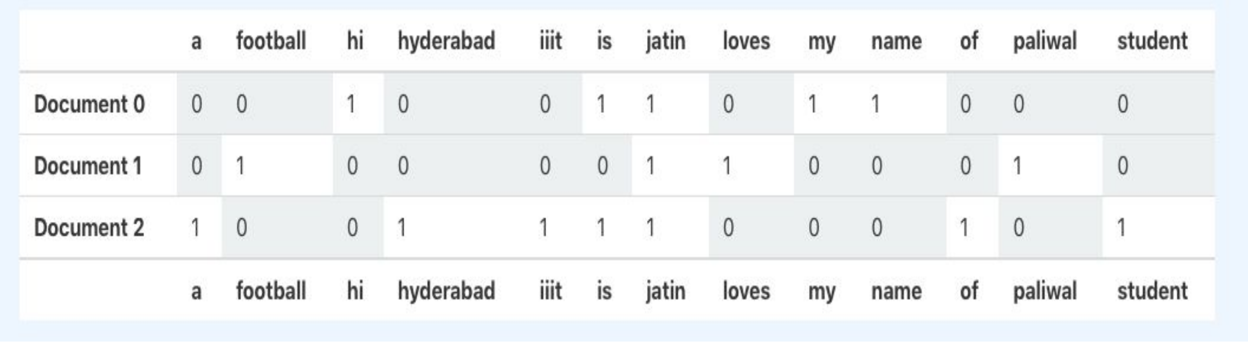
5/15

Step 1: Document as input to LDA model



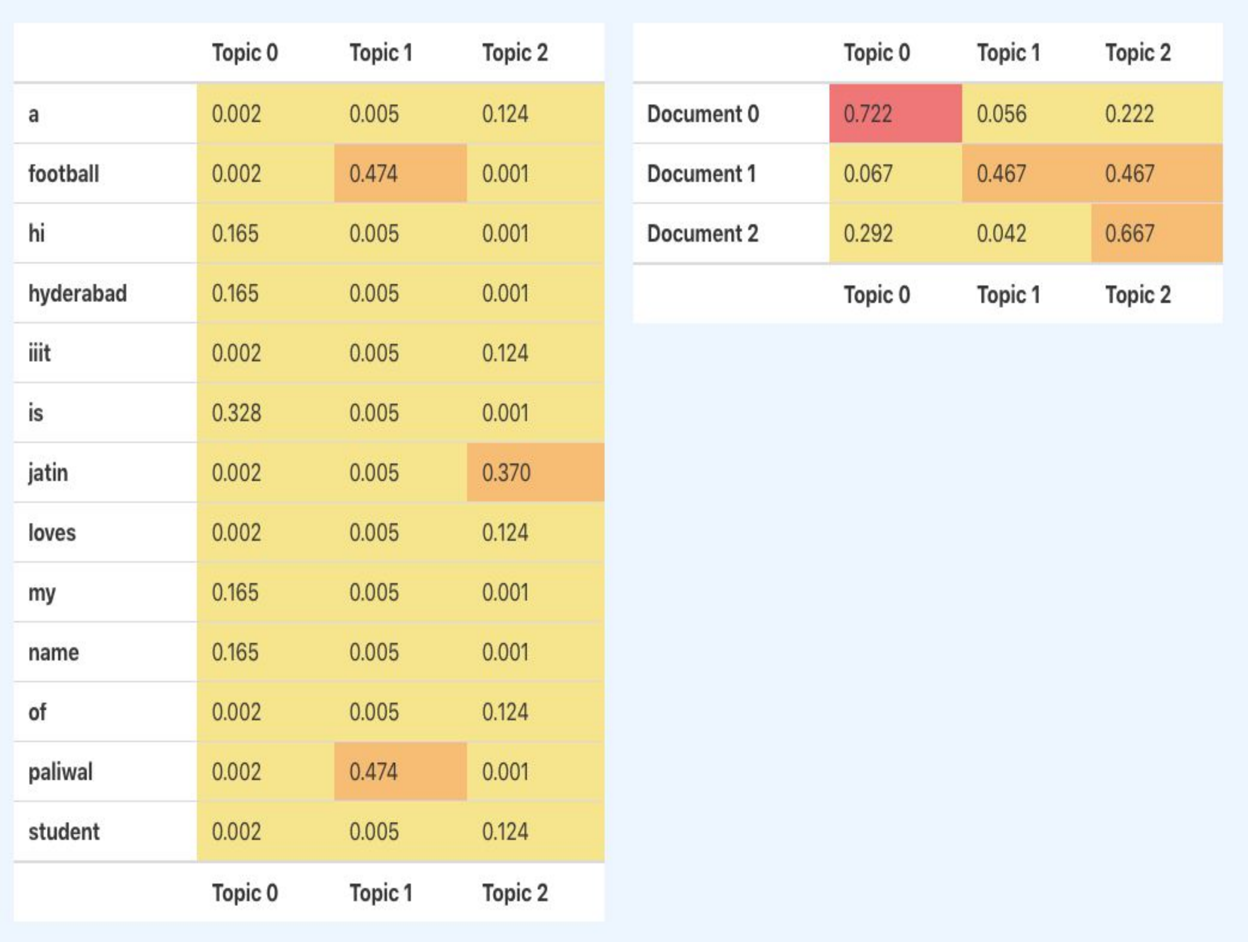
Document 0	Hi my name is Jatin
Document 1	Jatin Paliwal loves football
Document 2	Jatin is a student of IIIT Hyderabad

Step 2: Generate Bag of words:



	a	football	hi	hyderabad	iiit	is	jatin	loves	my	name	of	paliwal	student
Document 0	0	0	1	0	0	1	1	0	1	1	0	0	0
Document 1	0	1	0	0	0	0	1	1	0	0	0	1	0
Document 2	1	0	0	1	1	1	1	0	0	0	1	0	1
	a	football	hi	hyderabad	iiit	is	jatin	loves	my	name	of	paliwal	student

Step 3: Find probabilities of words in topics



	Topic 0	Topic 1	Topic 2
a	0.002	0.005	0.124
football	0.002	0.474	0.001
hi	0.165	0.005	0.001
hyderabad	0.165	0.005	0.001
iiit	0.002	0.005	0.124
is	0.328	0.005	0.001
jatin	0.002	0.005	0.370
loves	0.002	0.005	0.124
my	0.165	0.005	0.001
name	0.165	0.005	0.001
of	0.002	0.005	0.124
paliwal	0.002	0.474	0.001
student	0.002	0.005	0.124
	Topic 0	Topic 1	Topic 2

	Topic 0	Topic 1	Topic 2
Document 0	0.722	0.056	0.222
Document 1	0.067	0.467	0.467
Document 2	0.292	0.042	0.667
	Topic 0	Topic 1	Topic 2

Going deeper to reduce threads

We can solve this problem, by introducing a latent (i.e. hidden) layer. Say we know 10 topics/themes that occur throughout the documents. But these topics are not observed, we only observe words and documents, thus topics are latent. And we want to utilise this information to cut down on the number of threads. Then what we do is, connect the words to the topics

depending on how well that word fall in that topic and then connect the topics to the documents based on what topics each document touch upon.
Now say you got each document having around 5 topics and each topic relating to 500 words. That is we need 1000*5 threads to connect documents to topics and 10*500 threads to connect topics to words, adding up to 10000.

2.2 Training a CNN to predict probability distributions over LDA's Topics

Once we have the LDA topic model, we want to train a deep CNN model to predict the same probability distributions over topics as the LDA model does for textual information, but using only the raw pixels of new unseen images. For this we can generate a set of training (and validation) samples as follows:
Given an image from the training set we represent its corresponding textual information (captions) as probability values over the LDA's topics this way we obtain a set of M training (and validation) examples of the form {(x_1,y_1), ..., (x_m , y_m)} such that x_i is an image and y is the probability distribution over topics obtained by projecting its associated textual information into the LDA topic space. Using this training set we train a deep CNN to predict the probability distribution y is for unseen images, directly from the image pixels. In fact, we use a transfer learning approach here in order to shortcut the training process by fine-tuning the well known Inception deep CNN model.

2.3 Using topic models for generating word ranks

Once the LDA topic model is learned as explained in section-3.1, we can represent the textual information corresponding to an unobserved image as probability distribution over the topics of LDA model P (topic | text), which is done by projecting the textual information to the topic-space. Since the contribution of each word to each topic, P (word | topic) was pre-computed when we learned the LDA model, we can calculate the probability of occurrence for each word in the dictionary P (word | text) as follows:

$$P(word \mid text) = \sum_{i=1:K} (P(word \mid topic_i) P(topic_i \mid text))$$

Similarly, once the deep CNN is trained as explained in section-3.2, we can obtain the probability distribution over topics for an unseen image P (topic | image) as the output of the CNN when feeding the image pixels on its input.
Again, since the word-probability for each topic which P (word | topic) is known from the corresponding LDA model, which we used to supervise the training of deep CNN's training, we can calculate the probability of occurrence of each word in the dictionary P (word | image) as follows :

$$P(word \mid image) = \sum_{i=1:K} (P(word \mid topic_i)P(topic_i \mid image))$$

Using the obtained probability distributions over words (i.e. P (word | text)), or P (word | image)) we are able to rank a given dictionary in order to prioritize the words that have more chances to appear in a given image. In the following section we show how the the word rankings obtained from both approaches are very similar, which demonstrates the capability of the deep CNN to generate topic probabilities directly from the image pixels. Moreover, the rankings generated this way prove to be better that a frequency-based word ranking in predicting which are the expected scene text instances (words) to be found in a given image.

3. Implementation Details

In this section we present the experimental evaluation of the proposed method on its ability to generate lexicons that can be used to improve the performance of systems for reading text in natural scene images. First, we present the datasets used for training and evaluation in section 3.1. Then, in section 3.2, we provide the data pre-processing step, in Section 3.5 we explain the implementation details of our experiments. In section 3.3, we analyze the performance of the word rankings obtained by representing image captions as a mixture of LDA topics as detailed in section 2.3. Finally in section 3.4, we show the performance of the word ranking obtained with our CNN network trained for predicting topic probabilities.

3.1 Dataset

We make use of a standard dataset, namely the MS-COCO dataset. The MS-COCO is a large scale dataset providing task-specific annotations for object detection, segmentation, and image captioning.
The download link for the dataset is given below:
http://images.cododataset.org/annotations/annotations_trainval2014.zip

3.2 Data Pre-processing

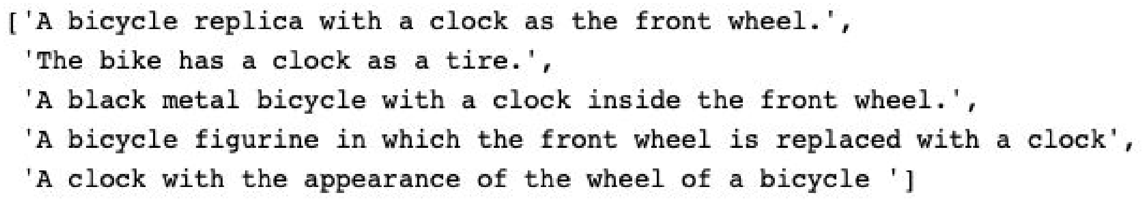
The data available is to us is in the form of json file containing captions as long sentences. We need to preprocess and convert it to a form that can be given as an input to the LDA model.

We have performed the following steps:

1.Tokenization: Split the text into sentences and the sentences into words. Lowercase the words and remove punctuation.
2.Words that have fewer than 3 characters are removed.
3.All stopwords are removed.
4.Words are lemmatized — words in third person are changed to first person and verbs in past and future tenses are changed into present.
5.Words are stemmed — words are reduced to their root form.

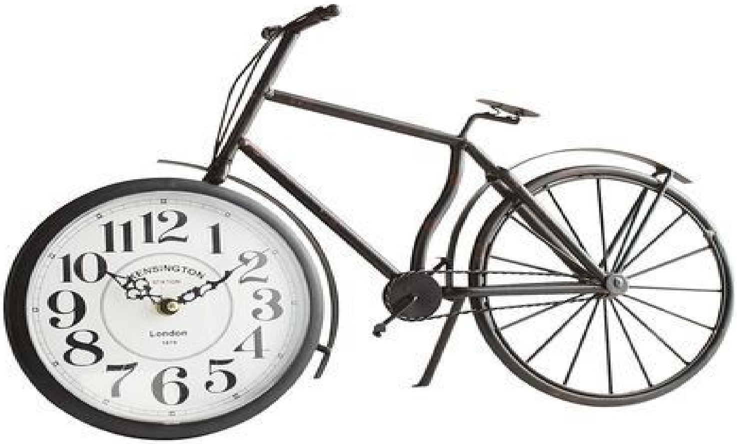
3.3 Model

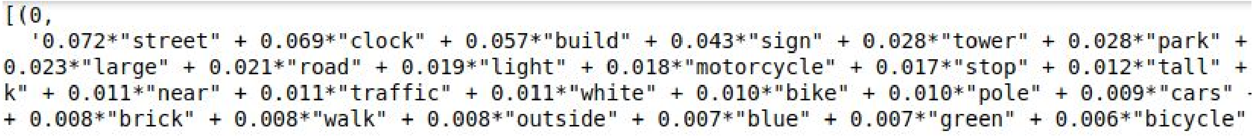
In our experiments involving topic modeling we have used the gensim Python library for learning and inferring the LDA model. We have learned multiple LDA models with a varying number of topics.



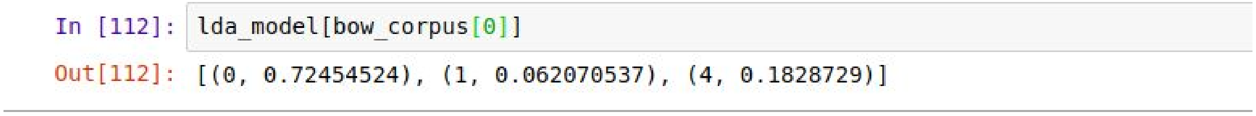
```
['A bicycle replica with a clock as the front wheel.',
 'The bike has a clock as a tire.',
 'A black metal bicycle with a clock inside the front wheel.',
 'A bicycle figurine in which the front wheel is replaced with a clock',
 'A clock with the appearance of the wheel of a bicycle ']
```

An example of captions for the image given below as input to the LDA
This is the text corpus associated with the given image(below) images.





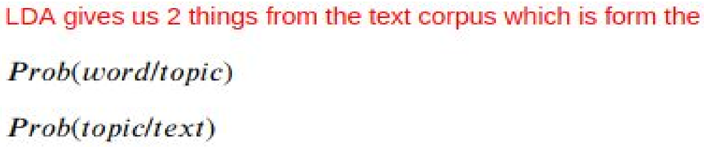
```
[(0,
  '0.072*"street" + 0.069*"clock" + 0.057*"build" + 0.043*"sign" + 0.028*"tower" + 0.028*"park" +
0.023*"large" + 0.021*"road" + 0.019*"light" + 0.018*"motorcycle" + 0.017*"stop" + 0.012*"tall" +
k" + 0.011*"near" + 0.011*"traffic" + 0.011*"white" + 0.010*"bike" + 0.010*"pole" + 0.009*"cars" -
+ 0.008*"brick" + 0.008*"walk" + 0.008*"outside" + 0.007*"blue" + 0.007*"green" + 0.006*"bicycle"
```



```
In [112]: lda_model[bow_corpus[0]]
Out[112]: [(0, 0.72454524), (1, 0.062070537), (4, 0.1828729)]
```

The result of LDA for the given image and captions

In the above images document 0 is a mixture of multiple topics, It contains a mix of
Topic 0 - 72%, Topic 1 - 6%, and Topic 4 - 18%.
As we can clearly see in the word distribution of the given topic, that topic 0 contains words like -> clock, cycle or bike, that's why it has more weightage for topic 0.



LDA gives us 2 things from the text corpus which is form the

$Prob(word/topic)$

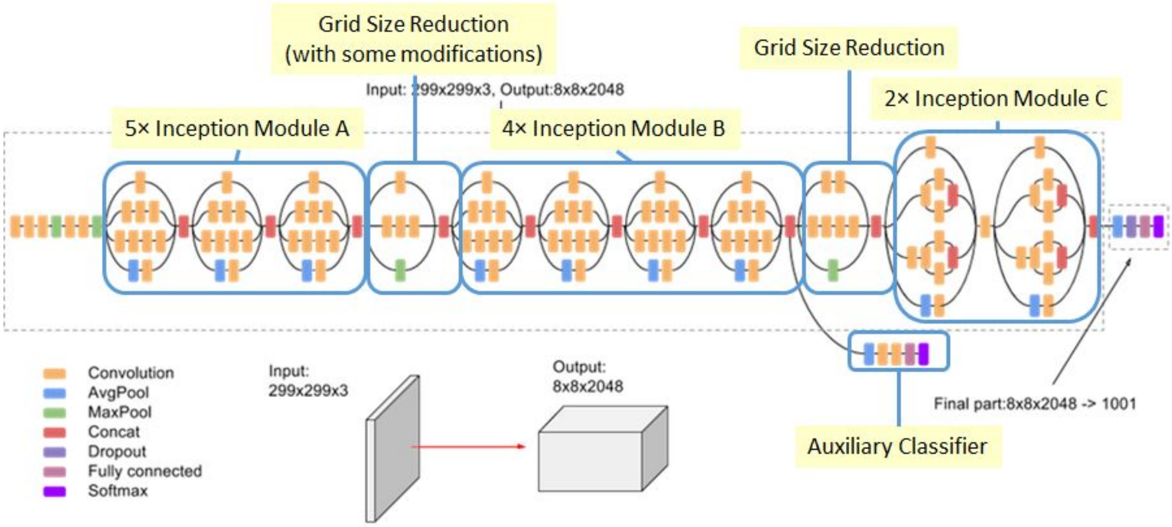
$Prob(topic/text)$

On the other hand, we have used the Keras framework for fine-tuning of the Inception v3 model.

Inception V3 model -

The Inception deep convolutional architecture was introduced as GoogLeNet in (Szegedy et al. 2015a), here named Inception-v1. Later the Inception architecture was refined in various ways, first by the introduction of batch normalization (Ioffe and Szegedy 2015) (Inception-v2). Later by additional factorization ideas in the third iteration (Szegedy et al. 2015b) which will be referred to as Inception-v3 in this report.”

We have trained three final layers of the net from scratch, where one is contains 256 neurons and one contains 128 and the last accommodating it to the size of our topic modeling task, and leaving the rest of the net untouched. The third and second last layer contains relu activation function, and the last layer contains softmax. We used the cross entropy loss function and Adam optimizer with a fixed learning rate of 0.001 and a batch size of 500 for 500 epochs.



An image showing the architecture of Inception V3

Prob(topic/text) will be used as class labels for training of our CNN Models.
Using the Training set we train a deep CNN to predict the probability distribution for unseen images directly from the image pixels.

3.4 Code repository

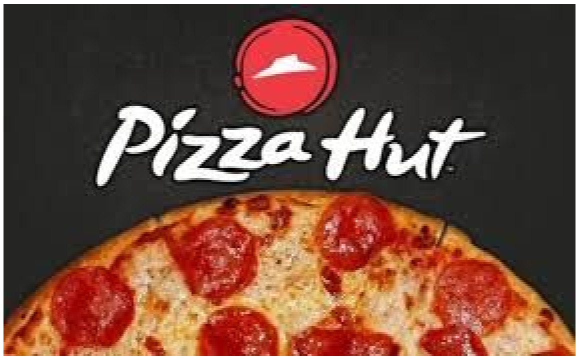
We have used Github for code repository and collaboration. The repository contains all the trained models, jupyter notebook, report and presentation. Below is the link for it:

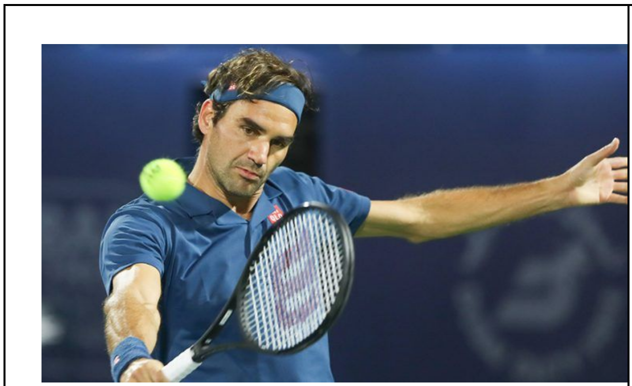
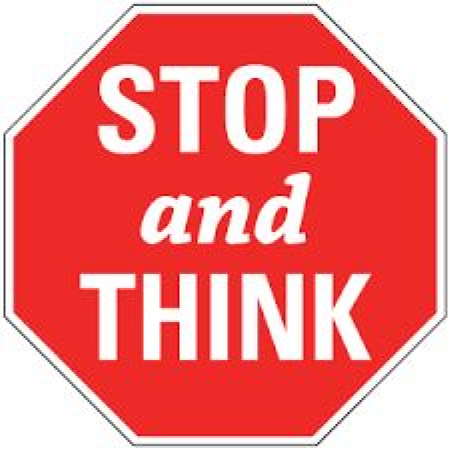
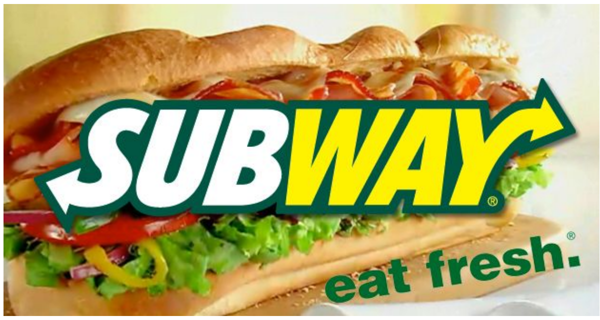
Link:
https://github.com/vishalbidawatka/Dynamic-Lexicon-Generation-for-Natural-Scene-Images.git

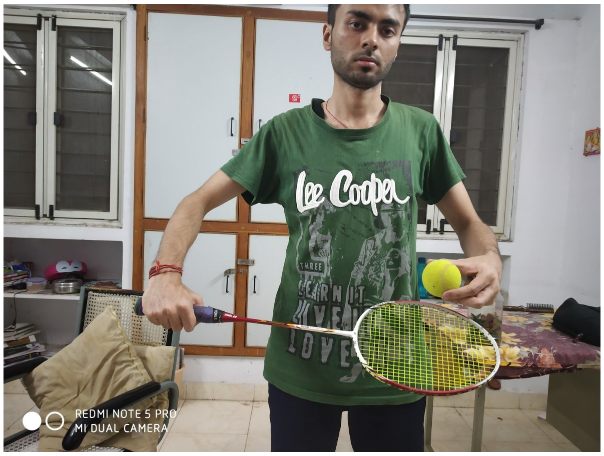
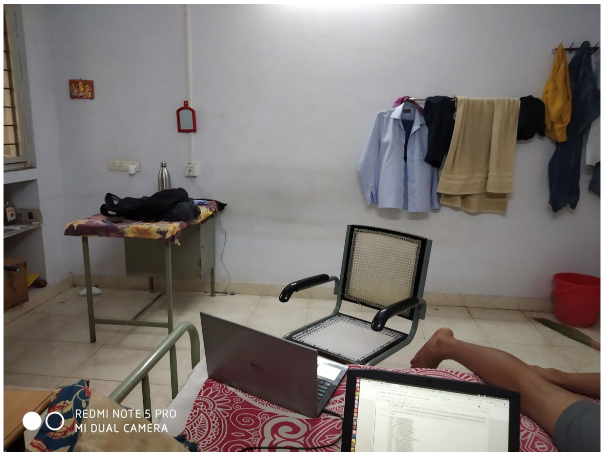
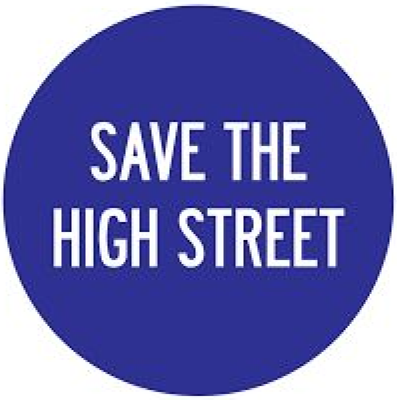
4. Observations

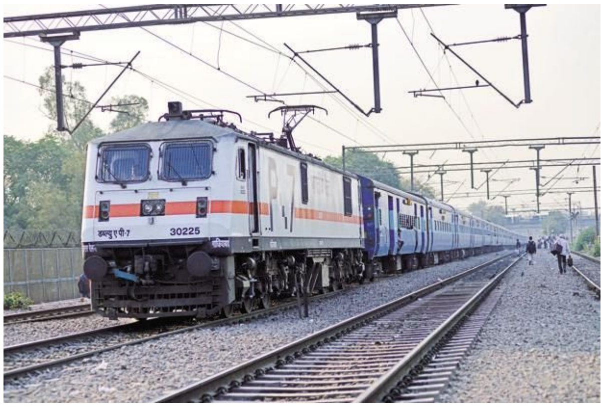
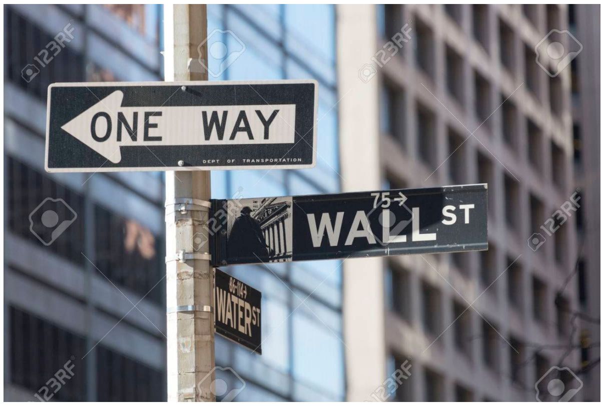
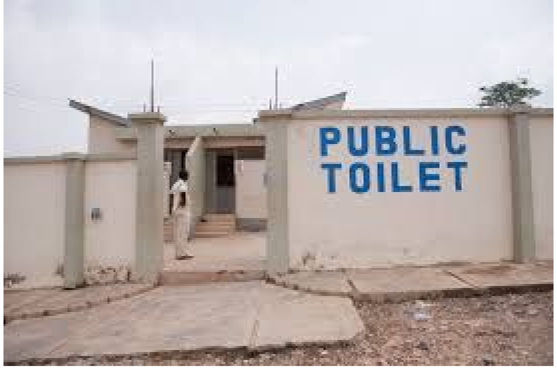
The CNN takes only the image raw pixels as input and generates the appropriate topics. The topics are based on the probabilities of it occurring in the image with highest probability given the minimum rank. Below are some of our observations on different images.

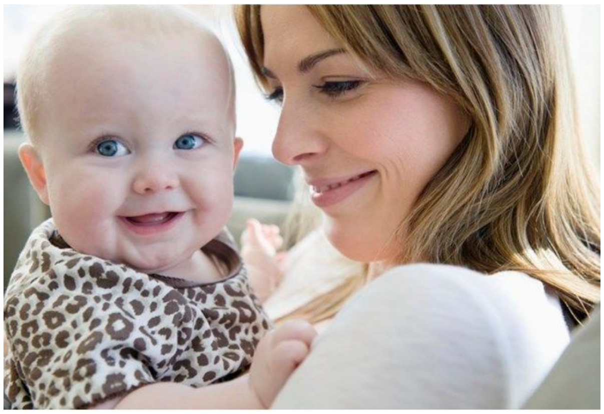
4.1 Output lexicons

	Rank: 1 PIZZA Rank: 11 SLICE

	Rank: 1 TENNIS Rank: 2 BALL Rank: 3 COURT Rank: 4 PLAYER Rank: 5 RACKET Rank: 6 PLAY Rank: 7 HOLD Rank: 9 RACQUET Rank: 10 SWING Rank: 12 SERVE Rank: 14 GAME Rank: 15 MATCH Rank: 18 MALE
	Rank: 2 SIGN Rank: 6 STOP
	Rank: 161 SUBWAY Rank: 3 FOOD Rank: 7 SANDWICH Rank: 18 VEGETABLES Rank: 16 ENGINE

	Rank: 2 HOLD Rank: 4 TENNIS Rank: 9 PERSON Rank: 11 TABLE Rank: 17 BALL Rank: 25 PLAYER Rank: 33 SERVE Rank: 46 RACQUET
	Rank: 1 LAPTOP Rank: 2 DESK Rank: 3 PEOPLE Rank: 4 WINDOW Rank: 7 TABLE Rank: 14 SCREEN Rank: 51 CHAIR Rank: 68 GLASS Rank: 171 SHIRT
	Rank: 39 STREET

	Rank: 1 TRAIN Rank: 2 TRACK Rank: 13 RAIL Rank: 9 PLATFORM
 ONE WAY DEPT OF TRANSPORTATION 75→ WALL ST WATER ST 123RF	Rank: 1 STREET Rank: 2 SIGN Rank: 28 CROSS
	Rank : 30 TOILET

	Rank: 21 SMILE Rank: 22 BABY Rank: 8 CHILD Rank: 2 WOMAN Rank: 27 LADY Rank: 72 FACE Rank: 14 HAND

Note: It produces particularly interesting results that can be potentially leveraged by end-to-end reading systems.
“TENNIS”, a word instance that is included in the image no. 2 and whose recognition would be very difficult without the context provided by the scene.

5 Conclusion

Topic Modeling statistical framework can be used to leverage the correlation between visual and textual information in order to predict the words that are more likely to appear in the image as scene text instances. Moreover, we have shown that is possible to train a deep CNN model to reproduce those topic model based word rankings but using only an image as input.

Our Implementation shows that the quality of the automatically obtained custom lexicons is superior to a generic frequency based baseline, and thus can be used to improve scene text recognition methods.

6 References

1. Dynamic Lexicon Generation for Natural Scene Images
   https://link.springer.com/chapter/10.1007/978-3-319-46604-0_29
2. https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

7 Task Assignment

1. Ishan Tyagi - LDA Implementation
2. Vishal Bidawatka - CNN

3. Bhavin Kotak - Mobilenet model implementation and generating ranking of lexicons from input images.
4. Jatin Paliwal - LDA and data pre-processing