**Abstract**

A long-standing goal in the field of artificial intelligence is to develop agents that can perceive and understand the rich visual world around us and who can communicate with us about it in natural language. Significant strides have been made towards this goal over the last few years due to simultaneous advances in computing infrastructure, data gathering and algorithms. The progress has been especially rapid in visual recognition, where computers can now classify images into categories with a performance that rivals that of humans, or even surpasses it in some cases such as classifying breeds of dogs. However, despite much encouraging progress, most of the advances in visual recognition still take place in the context of assigning one or a few discrete levels to an image(e.g. person, boat, keyboard etc.).

In this project we have used a model that can describe various parts of an image in natural language, from this sentences we have extracted action verbs which helps us identifying the event in which an object is busy. In particular, first we introduce a model that embeds both images and sentences into a common multi-model embedding space. This space then allows us to identify images that depict an arbitrary sentence description and conversely, we can identify sentences that describe an image. This model can take an image and directly generate a sentence description without being constrained a finite collection of human-written sentences to choose from. At the end to identify the events, we have used a parser to extract the action verbs.

**Literature Survey**

Advancement of computer science has made a huge progress in the area of information extraction from various documents. This document can be an image or a simple text file. Extracting valuable data and logically connecting them to form a valuable information is not so difficult. Attempt to extract temporal information is a popular and interesting research area of Natural Language Processing (NLP). Generally, events are described in different newspaper texts, stories and other important documents where events happen in time and temporal location and ordering of these events are specified. One of the important tasks of text analysis clearly requires identifying events described in a text and locating these in time. This is also important in a wide range of NLP applications that include temporal question answering, machine translation and document summarisation.

Temporal relation identification based on machine learning approaches an be found in Boguraev et al.(2005), Main et al.(2006), Chambers et al.(2007) and some of the TempEval 2007 participants (Verhagen et al., 2007). Most of these works tried to improve classification accuracies through feature engineering. The performance of any machine learning based system is often limited by the amount of available training data. Mani et al.(2006) introduced a temporal resigning component that greatly expands the available training data. The training set was increased by a factor of 10 by computing the closure of the various temporal relations that exist in the training data. The reported significant improvement of the classification accuracies on event-event and event-time relations. Experimental result showed the accuracies of 62.5%-94.95% and 73.68%-90.16% for event-event and event-time relations, respectively. However, this has two shortcomings, namely feature vector duplication caused by the data normalisation process and the unrealistic evaluation scheme. The solutions to these issues are briefly described in Mani et al. (2007). The main challenges involved in this test were first addressed during TempEval-1 in 2007 (Verhagen et al.,2007). This was an initial evaluation exercise based on three limited tasks that were considered realistic both from the perspective of assembling resources for development and testing and from the perspective of developing systems capable of addressing the tasks. In TempEval 2007, following types of event-time temporal relations were considered: **Task A** (relation between the events and times within the same sentence), **Task B** (relation between events and document creation time) and **Task C** (relation between verb events in adjacent sentences). The data set was based on TimeBank, a hand-built goal standard of annotated texts using the TimeML markup scheme1. The data sets included sentence boundaries, timex3 tags (including the special document creation time tag), and event tags. For tasks A and B, a restricted set of events was used, namely those events that occur more than 5 times in TimeBank. For all three tasks, the relation labels used were before, after, overlap, before-or-overlap, overlap-or-after and vague. Six teams participated in the TempEval tasks. Three of the teams used statistics exclusively, one used a rule-based system and the other two employed a hybrid approach. For task A, the range of F-measure scores were from 0.34 to 0.62 for the *strict scheme* and from 0.41 to 0.63 for the *relaxed scheme.* For task B, the scores ware from 0.66 to 0.80 (*strict*) and 0.71 to 0.81 (*relaxed*). Finally, task C scores range from 0.42 to 0.55 (*strict*) and from 0.56 to 0.66 (*relaxed*)*.*

We come across a hybrid approach for event extraction and event actor identification for new sources of textual information, rich in events, grow significantly, such as social networks, blogs, and wikis. They are added to old sources like informative web sites, emails and forums, which shows importance to manage these data automatically. One of the important tasks of text analysis clearly requires identifying events described in a text and locating these in time. Event extraction has emerged to be very important in improving complex natural language processing (NLP) applications such as automatic summarisation (Danial et al.,2003) and question answering (QA). TimeMl (Pustejovsky et al., 2003) presented a rich specification for annotating events in NL text extending the features of the previous one.

TimeML view of events as situations that *happen or occur,* or elements describing *states or circumstances* in which something obtains or holds the truth. These events are generally expressed by tensed or un-tensed verbs, normalisation, adjectives, predictive clauses or prepositional phrases. The 2007 TempEval challenge attempted to address this question (Boguraev et al, 2005). In 2010, TempEval-2, event extraction task was introduced as task B. Event extraction system is bases on a supervised machine learning, Support Vector Machine (SVM). It makes use of various features extracted from the TimeML corpus. In order to improve the performance of the system, we incorporate the knowledge of semantic role labelling, WordNet and several heuristics.

At this stage it can be understood that all events are involved with the actors, either active or passive. Actually, event actions are done by someone or somebody is doing this kind of action. Event actions involve with person, organisation and sometimes with location also. In the present attempt, the approaches that were conducted for identifying emotion holders (Das and Bandyopadhyay, 2010). Thereafter, came up the following heuristics for actor identification, there are various type of sentences, (i) non-event sentences, i.e. those sentences that don’t contain any event entity. (ii) If multiple events exist in any sentence, then all the events will have the same actors. Once an actor is identified for any event, it is assigned to the other event as well. (iii) If there are multiple actors and events, then <event, actor> pairs are formed by considering an event and its closest possible actor in the sentence. All the events my not have active actor. The actor my be passive also.

Another type of approach for temporal expression identification in by using Conditional Random Field (CRF)1 machine learning classifier. CRF++ templates can capture the relation between the different features in a sequence to identify temporal expressions. Temporal expressions mostly appear as multi-word entities such as “the next three days”. Therefore the use of CRF classifier that uses context information of a token seemed most appropriate. Initially, all the sentences have been changed to a vertical token-by-token level sequential structure for temporal expressions representation by a B-I-O encoding of temporal expression, “B” indicates the ‘beginning of sequence’, “I” indicates a token inside a sequence and “O” indicates an outside word. Carefully choosing the features list based on the several entities that denote month names, year, weekdays, various digit expressions (day, time, AM, PM etc.). In certain temporal expression patterns (*several months, last evening*) some words (*several, last*) act as modifiers to the following words that represent the time expression. Temporal expression include time expression modifiers, relative days, periodic temporal set, year-eve day, month name with their short pattern forms, session of year, time of day, decade list and so on. POS information of each token is a feature. Careful observation of a simple intuition revelation that most temporal expressions contain some tokens conveying the *“time”* information while others possibly conveying the *“quantity”* of time. For example, in the expression *“next three days”, “three”* quantifies *“days”*. Following are the different temporal expressions lists that have been utilised:

* + A list of time expression modifiers: *this, mid, recent, earlier, beginning, late* etc.
  + A list of relative days: *yesterday, tomorrow* etc.
  + A list of periodic temporal set: *hourly, nightly* etc.
  + A list of year eve day: *Christmas Day, Valentine Day* etc.
  + A list of month names with their short pattern forms: *April, Apr.* etc.
  + A list of season of year: *spring, winter* etc.
  + A list of time of day: *morning, afternoon, evening* etc.
  + A list of decades list: *twenties, thirties* etc.

Stanford CoreNLP tool is used to tokenise, lemmatise, named-entity annotate and part-of-speech tag the text portions of the input files. For event extraction, the features have been considered at word level, where each word has its own set of features. The general features used to train CRF model are:

**Morphological Features:** Event words are represented mostly as verbs and nouns. The major problem is detecting the events having non-verbal PoS levels. Linguistically, non-verbal word forms are derived from verbal word forms. Various inflectional and derivational morphological rules are involved in the process of evolving from verbal to non-verbal word forms. A set of handcrafted rules to identify the suffixes such as (*‘-cion’, ‘-tion’, ‘-ion’*)*,* i.e., the morphological markers of word token, where Person, Location and Organisation words are not considered. The POS and lemma, in a 5-window (-2,+2), has been used for event extraction.

**Syntactic Features:** Different event words notions are contained in the sentences such as: verb-noun combinations structure, the complements of aspectual propositional phrases (PPs) headed by prepositions and a particular type of complex prepositions. These notions are captured to be used as syntactic features for event extraction.

**WordNet Features:** The RiTa WordNet2 package has been effectively used to extract different properties of words, such as Synonyms, Antonyms, Hypernyms, & Hyponyms, Holonyms, Meeronyms, Coordinates, & Similars, Nominalizations, Verb-Groups, & Derived-terms. These Wordnet properties are used in the training file for the CRF in the form of binary features for verbs and nouns indicating if the words like “act”, “activity”, “phenomenon” etc. occur in different relations of the Wordnet ontology.

Till now we have talked about event identification from text and document which are available in huge amount in social media and news sites. But images are also an important source of information, specially we can collect lots of information from a single image. Every image has a story behind it. We have used the Dense Captioning task, which requires a computer vision system to both localise and describe salient regions in images in natural language. The dense captioning task generalises object detection when the descriptions consists of a single word, and Image Captioning when one predicted region covers the full image. To address the localisation and description task jointly we propose a Fully Convolutional Localisation Network (FCLN) architecture that processes an image with a single, efficient forward pass, requires no external regions proposals, and can be trained end-to-end with a single round of optimisation. The architecture is composed of Convolutional Network, a novel dense localisation layer, and Recurrent Neural Network language model that generates the label sequences. We evaluate our network on the Visual Genome dataset, which comprises 94,000 images and 4,100,000 region-grounded captions. We observe both speed and accuracy improvements over baselines based on current state of the art approaches in both generation and retrieval settings.

Our ability to effortlessly point out and describe all aspects of an image relies on a strong semantic understanding of a visual scene and all of its elements. However, despite numerous potential applications, this ability remains a challenge for our state of the art visual recognition systems. In the last years there has been significant progress in image classification, where the task is to assign one level to a image. Further work has pushed these advances along two orthogonal directions: First, rapid progress in object detection has identified models that efficiently identify and label multiple salient regions of an image. Second, recent advances in image captioning have expanded the complexity of the label space from a fixed set of categories to sequence of words able to express significantly richer concepts. However, despite encouraging progress along the label density and label complexity axes, these two directions have remained separate. In this work we take a step towards unifying these two inter-connected tasks into one joint framework. First, we introduce the dense captioning task, which requires a model to predict a set of descriptions across regions of an image. Object detection is hence recovered as a special case when the target labels consist of one word, and image captioning is recovered when all images consist of one region that spans the full image. Additionally, we have used a Fully Convolutional Localisation Network (FCLN) for the dense captioning task. Our model is inspired by recent work in image captioning in that it is composed of a Convolutional Neural Network and a Recurrent Neutral Network language model.

**Related Work**

Our work draws on recent work in object detection, image captioning, and soft spatial attention that allows downstream processing of particular regions in the image.

**Object Detection:** Our core visual processing module is Convolutional Neural Network (CNN) which has emerged as a powerful model for visual recognition tasks. The first application of these models to dense prediction tasks was introduced in R-CNN, where each region of interest was processed independently. Further work has focused on processing all regions with only single forward pass of the CNN, and on eliminating explicit region proposal methods by directly predicting the bounding boxes either in the image coordinate system, or in a fully convolutional and hence position-invariant settings. Most related to our approach in the work of Ren et al. who develop a region proposal network (RPN) that regresses from anchors to region of interest. However, they adopt a 4-step optimisation process, while our approach does not require training pipelines. Additionally, we replace their RoI pooling mechanism with a differentiable, spatial soft attention mechanism. In particular, this change allows us the back-propagate through the region proposal network and train the whole model jointly.

**Image Captioning:** Several pioneering approaches have explored the task of describing images with natural language. More recent approaches based on neural networks have adopted Recurrent Neural Networks (RNNs) as the core architectural element for generating captions. These models have previously Benn used in language modelling, where they are known to learn powerful long-time interactions. Several recent approaches to Image Captioning rely on combination of RNN language model conditioned on image information, possibly with soft attention mechanisms. Similar to our work, Karpathy and Fei-Fei run an image captioning model on regions but they do not tackle the joint task of detection of description in one model. This model is end-to-end and designed in such way that the prediction for each region is a function of the global image context, which can show ultimately stronger performance. Finally, the metrics developed for dense captioning task are inspired by metrics develop for image captioning.

**Proposed Work:**

The goal is to design an architecture that jointly localises region of interest and then describes each with natural language. The primary challenge is to develop a model that supports end-to-end training with a single step of optimisation, and both efficient and effective inference. Proposed architecture draws on architectural elements present in recent work on object detection, image captioning and soft spatial attention to simultaneously address these design constraints. Next we first describe the components of our model. Then we address the loss function and the details of training and interface.

**Model Architecture:**

**Convolutional Network:** We use the VGG-16 architecture for its state-of-the-art performance. It Consists of 13 layers of 3 cross 3 convolutions interspersed with 5 layers of 2 cross 2 max pooling. We remove the final pooling layer, so an input image of shape 3 cross W cross H gives rise to a tensor of features of shape C cross W’ cross H’ where C = 512, W’ = [W/16], and H’ = [H/16]. The output of this network encodes the appearance of the image at a set of uniformly sampled image locations, and forms the input to the localisation layer.

**Fully Convolutional Localisation Layer:** The localisation layer receives an input tensor of activations, identifies spatial regions of interest and smoothly extracts a fixed-sized representation from each region. Out approach is based on that of Faster R-CNN, but we replace their RoI pooling mechanism with bilinear interpolation, allowing our model to propagate gradients backward through the coordinates of predicted regions. This modification opens up the possibility of predicting affine or morphed region proposals instead of bounding boxes, but we leave these extensions to future work.

**Inputs/Outputs:** The localisation layer accepts a tensor of activations of size C cross W’ cross H’. It then internally selects B regions of interest and returns three output tensors giving information about these regions:

* 1. **Region Coordinates:** A matrix of shape B cross 4 giving bounding box coordinates for each output region.
  2. **Region Scores:** A vector of length B giving a confidence score for each output region. Regions with high confidence scores are more likely to correspond to ground-truth regions of interest.
  3. **Region Features:** A tensor of shape B cross C cross X cross Y giving features for output regions; is represented by an X cross Y grid of C-dimensional features.

**Convolutional Anchors:** Similar to Faster R-CCN, localisation layer predicts region proposals by regressing offsets from a set of translation-invariant anchors. In particular, we project each point in the W’ cross H’ grid of input features back into the W cross H image plane, and consider k anchor boxes of different aspect ratios entered at this projected point. For each of these k anchor boxes, the localisation layer predicts a confidence score and four scalars regressing from the anchor to predicted box coordinates. These are computed by passing the input feature map through a 3 cross e convolution with 256 filters, a rectified linear nonlinearity, and a 1 cross 1 convolution with 5k filters. This results in a tensor of shape 5k cross W’ cross H’ containing scores and offsets for all anchors.

**Box Regression:** By adopting the parameterisation to regress from anchors to region proposals. Given an anchor box with centre (*xa, ya*), width *wa,* and height *ha,* model predicts scalars (*tx,ty, tw, th*)givingnormalised offsets and log-space scaling transforms, so that the output region has centre (*x, y*) and shape (*w, h*)given by

*x = xa + txwa y = ya +tyha (1)*

*w = wa exp(tw) h= ha exp(hw) (2)*

Note that the boxes are discouraged from drifting too far from their anchors due to L2 regularisation.

**Box Sampling:** Processing a typical image of size W = 720, H = 540 with k = 12 anchor boxes gives rise to 17,280 region proposals. Since running the recognition network and the language model for all proposals would be prohibitively expensive, it is necessary to subsample them.

At training time, we sample a mini batch containing B = 256 boxes with at most B/2 positive regions and the rest negatives. A region is positive if it has an intersection over union (IoU) of at least 0.7 with some ground-truth region; in addition, the predicted region of maximal IoU with each ground-truth region is positive. A region is negative if it has IoU < 0.3 with all ground-truth regions. Out sampled mini batch contains Bp <= B/2 positive regions and Bn = B - Bp negative regions, sampled uniformly without replacement from the set of all positive and all negative regions respectively.

At test time we subsample using greedy non-maximum suppression (NMS) based on the predicted proposal confidences to select the B = 300 most confident proposals. The coordinates and confidences of the sampled proposals are collected into tensors for shape B cross 4 and B respectively, and are output forms the localisation layer.

**Bilinear Interpolation:** After sampling, we are left with region proposals of varying sizes and aspect ratios. In order to interface with the full-connected recognition network and the RNN language model, we must extract a fixed-size feature representation for each variably sized region proposal.

Fast R-CNN proposes an RoI pooling layer where each region proposal is projected onto the W’ cross H’ grid of convolutional features and divided into a coarse X cross Y grid aligned to pixel boundaries by rounding. Features are max-pooled within each grid cell, resulting in an X cross Y grid of output features.

The RoI pooling layer is a function of two inputs: convolutional features and region proposal coordinates. Gradients can be propagated backward from the output features to the input features, but not to the input proposal coordinates. To overcome this limitation, replacement of RoI pooling layer with bilinear interpolation can be effective.

Concretely, given an input feature map U to shape C cross W’ cross H’ and a region proposal, we interpolated the features of U to produce an output feature map V of shape C cross X cross Y. After projecting the region proposal onto U and compute a sampling grid G of shape X cross Y cross 2 associating each element of V with real-valued coordinates into U. If Gi,j = (xi,j , yi,j) then Vc,i,j should be equal to U at (c, xi, j , yi,j ); however since (xi,j , yi,j) are real-valued, we convolve with a sampling kernel k and set

Vc,i,j =Σwi’=1 ΣHj’=1 Uc,i’,j’ k(i’ - xi,j)k(j’ - yi,j) (3)

Bilinear sampling, corresponding to the kernel k(d) = max (0,1 - |d|) is used. The sampling grid is a linear function of the proposal coordinates, so gradients can be propagated backward into predicted region proposal coordinates. Running bilinear interpolation to extract features for all sampled regions gives a tensor of shape B cross C cross X cross Y, forming the final output from the localisation layer.