Graph Based Method Approach to the ImageCLEF2015 Task1 - Image Annotation

Ludovic Dos Santos, Benjamin Piwowarski, and Patrick Gallinari

Sorbonne Universits, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris.
firstname.lastname@lip6.fr

Abstract. We address the tasks of image classification and tagging through a transductive approach that automatically learns to project the different images onto a common latent space. This learned representation is then used to classify the images. We construst a graph between images and concepts using the deep representations given by ImageCLEF and WordNet database, and we exploit the idea that two connected nodes will tend to have a similar latent representation. This assumption allows us to learn correlations between the labels of connected images.

Key words: Representation Learning, Graph based method, Deep Relations

1 Introduction

The ImageCLEF 2015[3] image annotation [2] is composed of two subtasks : the Image Concept detection and localisation, and the Generation of Textual Descriptions of Images. We have participated in the former, whose objective is to predict which concepts are present in an image given in input. Use of labeled data is allowed, and we therefore used the available ImageNET Convolutional Neural Network (CNN).

The system we propose is based on a model [**?**] that projects nodes of a graph within a vector space (typically, of dimension 100-200). This model requires that some nodes be labeled, and uses this information within the embedding process. To leverage ImageNet data, where no relationship links images together, we had to build a graph. We experimented with different ways to construct a simple graph based on the output of the ImageNET CNN representations and the WordNet database[1], and presents results on the CLEF task corresponding to the settings of different hyperparameters of the model.

The article is organized as follows: Section 2 presents an overview of our model and how we construct the graph based on the images and the WordNet database, Section 3 the representation and the classifier learning, Section 4 our algorithm, and Section 5 outlines our results and conclusions.

2 Overview and graph construction

We propose a solution that relies on the exploitation of the correlations that exists between images that have a similar 1000 dimensional activations of the fc8 layer of Oxford VGGs 16-layer Convolutional Neural Network model (OV-CNN) given in the ImageCLEF challenge. It aims at taking directly into account the correlations between the CNN-labels of connected images. The underlying idea of this model is the following:

- We construct a graph mixing images and concepts based on the CNN-representation and the WordNet database.
- Each node is mapped onto a latent representation in a vectorial space $\mathbb{R}^Z$. The latent space is common to all node types (images and concepts).
- This latent representation defines a metric in $\mathbb{R}^Z$ such that two connected nodes tend to have a close representation (*smoothness assumption*).
- All the nodes and relations are not equal w.r.t. their position in the graph, their number of neighbors, etc. We use two functions, ψ_i and $\phi_{i,j}$, to model the importance of nodes and the relationships in the embedding process.
- A classification function is learned for the images. It takes as input a latent image representation and computes associated class labels.

All the notation used in this working note are summarized in Table 1.

Notation	Meaning
x_i	Node of the graph
A_k	1000 dimensional activations of the fc8 layer of an image k
z_i	Latent representation of node x_i
z_i^{copy}	Numerical copy of z_i
$\mathcal{T}$	Set of possible nodes types
T	Number of nodes types
t_i	Type of node x_i
$r(i,j)$	Relation type between node x_i and x_j
$i \xrightarrow{r} j$	Relation of type r from i to j
R	Number of relation's type
N	Number of nodes
$x_1 \cdots x_\ell$	Labeled nodes used for the classification task
N_j^r	Number of neighbors of x_j considering the relation r
E	Total number of edges
E^r	Total number of edges of type r
$\mathcal{Y}^t$	Set of categories associated to the type of node t
$\mathcal{C}^t$	Cardinality of $\mathcal{Y}^t$
y_i	Vector of categories for the node x_i
y_i^c	Value of category c for the node x_i

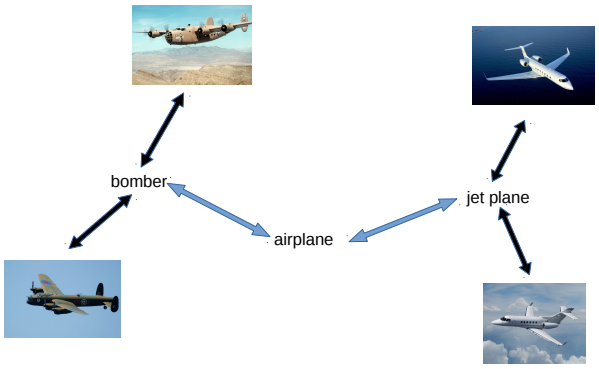
Table 1. Notations

2.1 Gaph construction

To apply our method, we need to construct a graph. The resulting graph is composed of nodes of two types, images and (WordNet) concepts. To construct the graph, we connect images to a subset of the concepts and then enrich this graph using Wordnet relationships between concepts.

We first directly use the representation A_k of an image, i.e. the 1000 dimensional activations of the OV-CNN. Each dimension of this vector corresponds to a label, and we suppose that an image is labeled by the i^{th} label if its corresponding component is above a threshold $\mu \in \mathbb{R}$. In that case, each label being a precise WordNet concept, we connect the image to the corresponding concept in the graph.

We then use WordNet to connect concepts together. We selected two relationships that may be important when trying to label images: the hypernym-hyponym relationship ("is-a") and the meronymy relationship ("part of"). In order to keep the graph size manageable, we started from the concepts of the CNN representation and added all the concepts that would be necessary to link the different concepts together. For example, in the figure below, we have four images. Two are classified "bomber" by the VO-CNN two "jet plane". Both concepts are hyponyms of airplane. We thus have the corresponding graph :



The final graph has 25,730,835 image-concept edges and 4,309 concept-concept edges.

2.2 Training set

The initial training set for subtask 1 contains 1,979 labeled images (there are 500,000 images in the entire dataset) for 250 concepts. That give us 7.9 images per concept on average, which is a relatively small number.
To increase the number of training examples, we again used the A_k representation and the corresponding OV-CNN concepts. We used another threshold $\nu \in \mathbb{R}$ to determine when an image should be labeled with a given OV-CNN concept.

In practice, given an image k, if the i-th component of A_k is bigger than ν we check if the corresponding OV-CNN concept is an hypernym of an ImageCLEF concept C_{clef}. If then, we add the image k to the training set labeled with C_{clef}.

When $\nu = 20$, the labeled images in our training set contains 43.970 images (175.9 images per concept), and when $\nu = 10$, 446.130 images (1784.5 images per concept).

3 Learning latent node representations

3.1 Classifier

The mapping onto the latent space is learned so that the labels for each type of node can be predicted from the latent representations. For that, we consider a linear classification function for each type of node k denoted by f_θ^k. This function takes as input a node representation and outputs the predicted label(s).

The f-functions can be learned by minimizing a loss on labeled data as follows:

$$\sum_{i=1}^{\ell} \psi_i \Delta(f_\theta^{t_i}(z_i), y_i) \tag{1}$$

where $\Delta(f_\theta^{t_i}(z_i^x), y_i)$ is the loss of predicting labels $f_\theta^{t_i}(z_i)$ instead of observed labels y_i. Here ψ_i represents the relative importance of node i in the graph. For example, $\psi_i = 1/N$ would define a model where all nodes have the same importance.

In our case, in order to set the value of ψ_i, we think our model should consider that all nodes have not the same importance, so we should be able to emphasis on more central nodes with regard to different relations. To do so, we use what makes a node more important is it has many neighbors of different types : $\psi_i = \frac{1}{R} \sum_{r=1}^{R} \frac{N_i^r}{E^r}$

In our implementation, we use a hinge-loss function for Δ:

$$\Delta(f_\theta^t(z), y) = \sum_{k=1}^{C^t} max(0; 1 - y^k f_\theta^{t,k}(z)) \tag{2}$$

where y^k is the desired score of category k for node x (-1 or $+1$) and $f_\theta^{t,k}(z)$ is the predicted score of category k by the model.

3.2 Transductive classification model

Let us denote by $z_i \in \mathbb{R}^Z$ the hidden representation of node x_i which is a vector of size Z. When updating z_i, in order to capture the graph metric in the

latent space, we use the following loss which embodies the metric smoothness assumption and forces linked nodes to have similar representations:

$$\sum_j \phi_{i,j} ||z_i - z'_j||^2 \tag{3}$$

We are using an L_2 norm in the latent space, but other metrics could be used as well. Furthermore, $\phi_{i,j}$ aims at considering the importance of an edge w.r.t. the hole graph and the shape of it around this specific edge.

We assume that all the edges of the graph should have the same impact regardless of their predominance ($\phi_{i,j} \propto \frac{1}{E^{r(i,j)}}$). Then, as in the PageRank algorithm, we assume that the information coming from a neighbor j of i should be divided equally through his neighbors of the same type as i ($\phi_{i,j} \propto \frac{1}{N_j^{r(i,j)}}$). Thus we set

$$\phi_{i,j} = \frac{1}{R N_j^{r(i,j)} E^{r(i,j)}}$$

3.3 Loss Function

The final expected objective loss of our model combines the classification and regularization losses 1 and 3:

$$L(z, \theta) = \sum_{i=1}^{\ell} \psi_i \Delta(f_\theta^{t_i}(z_i), y_i) + \lambda \sum_{i,j} \phi_{i,j} ||z_i - z'_j||^2 \tag{4}$$

The minimization of this function aims at finding a trade-off between the smoothness over the latent representations of correlated nodes $\mathcal{Z}$ and the predicted observed labels in $\mathcal{Y}_k$. Optimizing this loss allows us to learn:

- The projection z_i of each node x_i in the latent space.
- The classification functions f_θ^k for each nodes type k which transform the latent space to categories scores.

4 Algorithm

Learning consists in minimizing the loss function defined in Equation 4. Different optimization methods can be considered. We have used a Stochastic Gradient Descent Method to learn the latent representations.

Furthermore, we can remark that the $\phi_{i,j}$ and ψ_i can also be understood as a specific way of sampling during the SGD. Actually, if we firstly choose uniformly the type of edge then choose uniformly an edge given that type and pick the nodes from the chosen edge we update a given representation z_i in expectation $\frac{1}{R}\sum_{r=1}^{R} \frac{N_i^r}{E^r}$ times for the classification term and $\frac{1}{R}\sum_{r=1}^{R} \frac{N_i^r}{E^r}$ times for the graph regularization one. Adding the PageRank division in the graph regularization, we find out our previous assumptions. The algorithm is detailed below.

For a fixed number of iterations
  Choose r in $\mathcal{R}$ at random.
  Pick randomly an edge $i \xrightarrow{r} j$.
  If $(i \leq \ell)$
    $\theta \leftarrow \theta + \epsilon \nabla_\theta \Delta(f_\theta^{t_i}(z_i), y_i)$
    $z_i \leftarrow z_i + \epsilon \nabla_{z_i} \Delta(f_\theta^{t_i}(z_i), y_i)$
  If $(j \leq \ell)$
    $\theta \leftarrow \theta + \epsilon \nabla \Delta(f_\theta^{t_j}(z_j), y_j)$
    $z_j \leftarrow z_j + \epsilon \nabla_{z_j} \Delta(f_\theta^{t_j}(z_j), y_j)$
  $z_i \leftarrow z_i + \epsilon \lambda \nabla_{z_i} ||z_i - z_j||^2$
  $z_j \leftarrow z_j + \epsilon \lambda \nabla_{z_j} ||z_i - z_j||^2$

The algorithm chooses iteratively, with the sampling method explained above, a pair of connected nodes and then make a gradient update over the parameters of the model. If one of the chosen nodes is part of the first labeled training set, the algorithm first performs an update according to the first term of Equation 1. This update – lines 5-6 and 9-10 – consists in successively modifying the parameters of the classification function θ and of the latent representations z_i and z_j so as to minimize the classification loss term.

Here, ϵ is the gradient step, and λ is the trade-off between the classification and smoothness terms.

5 Results

As our model is not able to localize a concept in a given image we only present the results of the mean average precision with zero overlap (MAP-0).

Model	MAP 0 overlap
$N = 250, \nu = 10, \lambda = 50$	0.057422
$N = 100, \nu = 10, \lambda = 50$	0.056822
$N = 250, \nu = 10, \lambda = 150$	0.05625
$N = 100, \nu = 10, \lambda = 150$	0.056141
$N = 250, \nu = 20, \lambda = 150$	0.052665
$N = 250, \nu = 20, \lambda = 50$	0.052585

As we can see, the bigger the training set is (ν is small) the better our model performs. As the graph has many edges our model needs a bigger dimension (N) in order to adapt to the graph topology. Furthermore we can see that the more we constrain (λ is big) the representations of two nodes to be close, the worst the performances get.

The results are quite low compared to other opproaches, but we did not exploit (at least directly) any image related feature. Compared to a random approach, we are still able to extract some information from a graph constructed

using some basic information about the classification of an image and the relationship between the labels/concepts.

Our methodology would benefit from a real graph linking the images (e.g. a social network), and results would be much improved if using image-based features.

In future work, we will combine this approach with image features to check whether they can bring some further information.

Acknowledgments. This work was supported in part by a grant from the Research Agency ANR (Agence Nationale de la Recherche) under the MLVIS project.

References

1. Christiane Fellbaum. WordNet: An Electronic Lexical Database. Bradford Books. 1998.
2. Andrew Gilbert, Luca Piras, Josiah Wang, Fei Yan, Emmanuel Dellandrea, Robert Gaizauskas, Mauricio Villegas, and Krystian Mikolajczyk. Overview of the ImageCLEF 2015 Scalable Image Annotation, Localization and Sentence Generation task. In *CLEF2015 Working Notes*, CEUR Workshop Proceedings, Toulouse, France, September 8-11 2015. CEUR-WS.org.
3. Mauricio Villegas, Henning Müller, Andrew Gilbert, Luca Piras, Josiah Wang, Krystian Mikolajczyk, Alba García Seco de Herrera, Stefano Bromuri, M. Ashraful Amin, Mahmood Kazi Mohammed, Burak Acar, Suzan Uskudarli, Neda B. Marvasti, José F. Aldana, and María del Mar Roldán García. General Overview of ImageCLEF at the CLEF 2015 Labs. Lecture Notes in Computer Science. Springer International Publishing, 2015.