Structured Prediction with Output Embeddings for Semantic Image Annotation

Ariadna Quattoni[1], **Arnau Ramisa**[2], **Pranava Swaroop Madhyastha**[3]
Edgar Simo-Serra[4], **Francesc Moreno-Noguer**[2]
[1]Xerox Research Center Europe, ariadna.quattoni@xrce.xerox.com
[2] Institut de Robòtica i Informàtica Industrial (CSIC-UPC), {aramisa,fmoreno}@iri.upc.edu
[3]TALP Research Center, Universitat Politècnica de Catalunya, pranava@cs.upc.edu
[4]Waseda University, esimo@aoni.waseda.jp

Abstract

We address the task of annotating images with semantic tuples. Solving this problem requires an algorithm able to deal with hundreds of classes for each argument of the tuple. In such contexts, data sparsity becomes a key challenge. We propose handling this sparsity by incorporating feature representations of both the inputs (images) and outputs (argument classes) into a factorized log-linear model.

1 Introduction

Many important problems in machine learning can be framed as structured prediction tasks where the goal is to learn functions that map inputs to structured outputs such as sequences, trees or general graphs. A wide range of applications involve learning over large state spaces, e.g., if the output is a labeled graph, each node of the graph may take values over a potentially large set of labels. Data sparsity then becomes a challenge, as there will be many classes with very few training examples.

Within this context, we are interested in the task of predicting semantic tuples for images. That is, given an input image we seek to predict what are the events or actions (referred here as *predicates*), who and what are the participants (referred here as *actors*) of the actions and where is the action taking place (referred here as *locatives*). For example, an image might be annotated with the semantic tuples: $\langle run, dog, park \rangle$ and $\langle play, dog, grass \rangle$. We call each field of a tuple an *argument*.

To handle the data sparsity challenge imposed by the large state space, we will leverage an approach that has proven to be useful in multiclass and multilabel prediction tasks (Weston et al., 2010; Akata et al., 2013). The idea is to represent a value for an argument a using a feature vector representation $\phi \in I\!R^n$. We will integrate this argument representation into the structured prediction model.

In summary, our main contribution is to propose an approach that incorporates feature representations of the outputs into a structured prediction model, and apply it to the problem of annotating images with semantic tuples. We present an experimental study using different output feature representations and analyze how they affect performance for different argument types.

2 Semantic Tuple Image Annotation

Task: We will address the task of predicting semantic tuples for images. Following Farhadi et al. (2010), we will focus on a simple semantic representation that considers three basic arguments: predicate, actors and locatives. For example, in the tuple $\langle play, dog, grass \rangle$, "$play$" is the predicate, "$dog$" is the actor and "$grass$" is the locative.

Given this representation, we can formally define our problem as that of learning a function $\theta : X \times P \times A \times L \rightarrow I\!R$ that scores the compatibility between images and semantic tuples. Here X is the space of images; P, A and L are discrete sets of predicate, actor and locative arguments respectively, and $\langle p\ a\ l \rangle$ is a specific tuple instance. The overall learning process is illustrated in Fig. 1.

Dataset: For our experiments we used a subset of the Flickr8k dataset, proposed in Hodosh et al. (2013). This dataset (subset $\mathcal{B}$ in Fig. 1) consists of 8,000 images from Flickr of people and animals (mostly dogs) performing some action, with five crowd-sourced descriptive captions for each one.

We first manually annotated 1,544 captions, corresponding to 311 images (approximately one third of the development set (subset $\mathcal{C}$ in Fig. 1), producing more than 2,000 semantic tuples of predicate, actor and locative. For the experiments we partitioned the images and annotations into training, validation and test sets of 150, 50 and 100 images respectively.

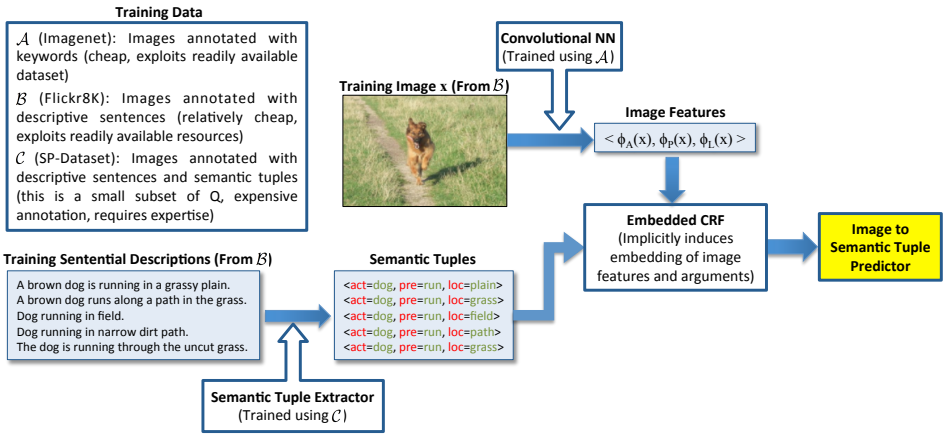

Figure 1: Overview of our approach. First, images $x \in \mathcal{A}$ are represented using image features $\phi_s(x)$, and semantic tuples are obtained applying our semantic tuple extractor (learned from the subset $\mathcal{C}$) to their corresponding captions. The resulting enlarged training set, where each image is paired with a bag of semantic tuples, is used to train our embedded CRF model that maps images to semantic tuples.

Data augmentation: To enlarge the manually annotated dataset we trained a model able to predict semantic tuples from captions using standard shallow and deep linguistic features (e.g., POS tags, dependency parsing, semantic role labeling). We extract the predicates by looking at the words tagged as verbs by the POS tagger. Then, the extraction of arguments for each predicate is resolved as a classification problem.

More specifically, for each detected predicate in a sentence we regard each noun as a positive or negative training example of a given relation depending on whether the candidate noun is or is not an argument of the predicate. We use these examples to train a SVM classifier that predicts if a candidate noun is an argument of a given predicate based on several linguistic features computed over the syntactic path of the dependency tree that connects them. We run the learned tuple predictor model on all the captions of the Fickr8k dataset to obtain a larger dataset of 8,000 images paired with semantic tuples.

3 Bilinear Models with Output Features

In this section we explain how we incorporate output feature representations into a factorized linear model. For simplicity, we will consider factorized sequence models over sequences of fixed length. However, it should not be hard to see that all the ideas presented here can be easily generalized to other structured prediction settings.

Let $y = [y_1 \dots y_T]$ be a set of labels and $S = [S_1, \dots, S_T]$ be the set of possible label values, where $y_i \in S_i$. We are interested in learning a model that computes $P(y|x)$, i.e., the conditional probability of a sequence y given some input x. We will consider factorized log-linear models that take the form:

$$P(y|x) = \frac{exp^{\theta(x,y)}}{\sum_y exp^{\theta(x,y)}} \quad (1)$$

The scoring function $\theta(x, y)$ is modeled as a sum of unary and binary bilinear potentials and is defined as:

$$\theta(x,y) = \sum_{t=1}^{T} v_{y_t}^\top W_t \phi(x,t) + \sum_{t=1}^{T} v_{y_t}^\top Z_t v_{y_{t+1}} \quad (2)$$

where $v_{y_t} \in I\!R^{n_t}$ is a n_t-dimensional feature representation of label arguments $y_t \in S_t$ and $\phi(x,t) \in I\!R^{d_t}$ is a d_t-dimensional feature representation of the t^{th} input factor of x.

The first set of terms in the above equation are usually referred as unary potentials and measure the compatibility between a single state at t and the feature representation of input factor t. The second set of terms are the binary potentials and measure the compatibility between pairs of states at adjacent factors. The scoring $\theta(x, y)$ function is fully parameterized by the unary parameter matrices $W_t \in I\!R^{n_t \times d_t}$ and the binary parameter matrices $Z_t \in I\!R^{n_t \times n_t}$.

The main idea is to define a feature space where semantically similar labels will be close. Like in the multilabel scenario (Weston et al., 2010; Akata et al., 2013), having full feature representations for arguments will allow us to share information across different classes and generalize better. With

a good output feature representation, our model should be able to make sensible predictions about pairs of arguments that it has not observed at training. This is easy to see: consider a case were we have a pair of arguments represented with feature vectors a_1 and a_2 and suppose that we have not observed the factor a_1, a_2 in our training data but we have observed the factor b_1, b_2. Then if a_1 is close in the feature space to argument b_1 and a_2 is close to b_2 our model will predict that a_1 and a_2 are compatible. That is it will assign probability to the factor a_1, a_2 which seems a natural generalization from the observed training data.

Now we show that the rank of W and Z have useful interpretations. Let $W = U\Sigma V$ be the singular value decomposition of W. We can then write unary potentials: $v_y^\top W\phi(x,t)$ as: $v_y^\top U\ \Sigma\ [V\phi(x,t)]$ Thus we can regard the bilinear form as a function computing a weighted inner product over some real embedding $v_y^\top U$ representing state y and some real embedding $[V\phi(x,t)]$ representing input factor t. The rank of W gives us the intrinsic dimensionality of the embedding. Thus if we want to induce shared low-dimensional embeddings across different states it seems reasonable to impose a low rank penalty on W. Similarly, let $Z = U\Sigma V$ be now the singular value decomposition of Z. We can write the binary potentials $v_y^\top Z v_{y'}$ as: $v_y^\top U\ \Sigma\ V v_{y'}$ and thus the binary potentials compute a weighted inner product between a real embedding of state y and a real embedding of state y'. As before, the rank of Z gives us the intrinsic dimensionality of the embedding and, to induce a low dimensional embedding for binary potentials, we will impose a low rank penalty on Z.

After having described the type of scoring functions we are interested in, we now turn our attention to the learning problem. That is, given a training set $D = \{\langle x\ y\rangle\}$ of pairs of inputs x and output sequences y we need to learn the parameters $\{W\}$ and $\{Z\}$. For this purpose we will do standard max-likelihood estimation and find the parameters that minimize the conditional negative log-likelihood of the data in D. That is, we will find the $\{W\}$ and $\{Z\}$ that minimize the following loss function $\mathcal{L}(D,\{W\},\{Z\})$: $-\sum_{\langle x\ y\rangle\in D} logP(y|x;\{W\},\{Z\})$ Recall that we are interested in learning low-rank unary and binary potentials. To achieve this we take a common approach which is to use as the nuclear norm $|W|_*$ and $|Z|_*$ as a convex approximation of the rank function, the final optimization problem becomes:

$$min_{\{W\}}\mathcal{L}(D,\{W\}) + \sum_t \alpha|W_t|_* + \beta|Z_t|_* \quad (3)$$

where $\mathcal{L}(D,\{W\}) = \sum_{d\in D} loss(d,\{W\})$ is the negative log likelihood function and α and β are two constants that control the trade off between minimizing the loss and the implicit dimensionality of the embeddings. We use a simple optimization scheme known as Forward Backward Splitting, or FOBOS (Duchi and Singer, 2009).

For our task we will consider a simple factorized scoring function: $\theta(x,\langle p\ a\ l\rangle)$ that has one factor associated with the *locative-predicate* pair and one factor associated with the *predicate-actor* pair. Since this corresponds to a chain structure, $argmax_{t\in T}\theta(x;\langle p\ a\ l\rangle)$ can be efficiently computed using Viterbi decoding in time $\mathcal{O}(N^2)$, where $N = max(|P|,|A|,|L|)$. Similarly, we can also find the top k predictions in $\mathcal{O}(kN^2)$. Thus for this application the scoring function of the bilinear CRF will take the form of:

$$\begin{aligned}\theta(x,\langle p\ a\ l\rangle) \quad = \quad & \lambda_{loc}(l)^\top W_{loc}\phi_{loc}(l) \\ & +\lambda_{pre}(p)^\top W_{pre}\phi_{pre}(p) \\ & +\lambda_{act}(a)^\top W_{act}\phi_{act}(a) \\ & +\phi_{loc}(l)^\top W^{loc}_{pre}\phi_{pre}(p) \\ & +\phi_{pre}(p)^\top W^{pre}_{act}\phi_{act}(a) \quad (4)\end{aligned}$$

The unary potentials measure the compatibility between an image and a semantic argument, the first binary potential measures the compatibility between a locative and a predicate, and the second binary potential measures the compatibility between a predicate and an actor. The scoring function is fully parameterized by the unary parameter matrices $W_{loc} \in I\!R^{d_l\times n_l}$, $W_{pre} \in I\!R^{d_p\times n_p}$ and $W_a \in I\!R^{d_a\times n_a}$ and the binary parameter matrices $W^{loc}_{pre} \in I\!R^{n_l\times n_p}$ and $W^{pre}_{act} \in I\!R^{n_p\times n_a}$. Where, n_l, n_p and n_a are the dimensionality of the locatives, predicates and actors feature representations, respectively and d_l, d_p and d_a are the dimensionality of the image representations. Notice that if we let the argument representation $\phi_t(r) \in I\!R^{|S_t|}$ be an indicator vector for label argument t, we obtain the usual parametrization of a standard factorized linear model, while having a dense feature representations for arguments instead of indicator vectors will allow us to share information across different classes.

4 Representing Semantic Arguments

We will conduct experiments with two different feature representations: 1) Fully unsupervised *Skip-Gram based Continuous Word Representations* (SCWR) representation (Mikolov et al., 2013) and 2) A feature representation computed using the $\langle caption, semantic\text{-}tuples \rangle$ pairs, that we call *Semantic Equivalence Representation* (SER).

We decided to exploit the dataset of captions paired with semantic tuples to induce a useful feature representation for arguments. The idea is quite simple: we wish to leverage the fact that any pair of semantic tuples associated with the same image will be likely describing the same event. Thus, they are in essence different ways of lexicalizing the same underlying concept. Let's look at a concrete example. Imagine that we have an image annotated with the tuples: $\langle play,\ dog,\ water \rangle$ and $\langle play,\ dog,\ river \rangle$. Since both tuples describe the same image, it is quite likely that both "$river$" and "$water$" refer to the same real world entity, i.e, "$river$" and "$water$" are 'semantically equivalent' for this image. Using this idea we can build a representation $\phi_{loc}(i) \in I\!R^{|L|}$ where the j-th dimension corresponds to the number of times the argument j has been semantically equivalent to argument i. More precisely, we compute the probability that argument j can be exchanged with argument i as: $\frac{[i,j]_{sr}}{\sum_j [i,j]_{sr}}$ Where $[i,j]_{sr}$ is the number of times that i and j have appeared as annotations of the same image and with the same other arguments. For example, for the actor arguments $[i,j]_{sr}$ represents the number of time that actor i and actor j have appeared with the same locative and predicate as descriptions of the same image.

5 Related Work

In recent years, some works have tackled the problem of generating rich textual descriptions of images. One of the pioneers is (Kulkarni et al., 2011), where a CRF model combines the output of several vision systems to produce input for a language generation method. In Farhadi et al. (2010), the authors find the similarity between sentences and images in a "meaning" space, represented by semantic tuples which are very similar to our triplets. Other works focus on a simplified problem: ranking of human-generated captions for images. Hodosh et al. (2013) propose to use Kernel Canonical Correlation Analysis to project images and their captions into a joint representation space, in which images and captions can be related and ranked to perform illustration and annotation tasks. Socher et al. (2014) also address the ranking of images given a sentence and vice-versa using a common subspace learned via Recursive Neural Networks. Other recent works also exploit deep networks to address the problem (Vinyals et al., 2015; Karpathy and Fei-Fei, 2015). Using label embeddings combined with bilinear forms has been previously proposed in the context of multiclass and multilabel image classification (Weston et al., 2010; Akata et al., 2013).

6 Experiments

For image features we use the 4,096-dimensional second to last layer of BVLC implementation of 'AlexNet' ImageNet model, a Convolutional Neural Network (CNN) as described in Jia et al. (2014). To test our method we used the 100 test images that were annotated with ground-truth semantic tuples. To measure performance we first predict the top tuple for each image and then measure accuracy for each argument type (i.e. the number of correct predictions among the top 1 triplets). The regularization parameters of each model were set using the validation set. We compare the performance of the following models: 1) Baseline Separate Predictors (S-Pred): We consider a baseline made of independent predictors for each argument type. More specifically we train one-vs-all SVMs (we also tried multi-class SVMs but they did not improve performance) to independently predict locatives, predicates and actors. For each argument type and candidate label we have a score computed by the corresponding SVM. Given an image we generate the top tuples that maximize the sum of scores for each argument type; 2) Baseline KCCA: This model implements the Kernel Canonical Correlation Analysis approach of Hodosh et al. (2013). We first note that this approach is able to rank a list of candidate captions but cannot directly generate tuples. To generate tuples for test images, we first find the caption in the training set that has the highest ranking score for that image and then extract the corresponding semantic tuples from that caption; 3) Indicator Features (IND), this is a standard factorized log-linear model that does not use any feature representation for the outputs; 4) A model that uses the

Figure 2: Samples of predicted tuples. **Top-left:** Examples of visually correct predictions. **Bottom:** Typical errors on one or several arguments. **Top-right:** Sample image and its top predicted tuples. The tuples in blue were not observed neither in the SP-Dataset nor in the automatically enlarged dataset. Note that all of them are descriptive of what is occurring in the scene.

skip-gram continuous word representation of outputs (SCWR); 5) A model that uses that semantic equivalence representation of outputs (SER); 6) A combined model that makes predictions using the best feature representation for each argument type (COMBO).

	S-Pred	KCCA	IND	SCWR	SER	COMBO
LOC	15	23	32	28	**33**	
PRED	11	20	24	**33**	25	
ACT	30	25	**52**	51	50	
MEAN	18.6	22.6	36	37.3	36	**39.3**

Table 1: Comparison of Output Feature Representation.

Table 1 shows the results. We observe that our proposed method performs significantly better than the baselines. The second observation is that the best performing output feature representation is different for different argument types, for the locatives the best representation is SER, for the predicates is the SCWR and for the actors using an output feature representation actually hurts performance. The biggest improvement we get is on the predicate arguments, where we improve almost by 10% in average precision over the baseline using the skip-gram word representation. Overall, the model that uses the best representation performs better than the indicator baseline.

Regarding the rank of the parameter matrices, we observed that the learned models can work well even if we drop the rank to 10% of its maximum rank. This shows that the learned models are efficient in the sense that they can work well with low-dimensional projections of the features.

7 Conclusion

In this paper we have presented a framework for exploiting input and output embeddings in the context of structured prediction. We have applied this framework to the problem of predicting compositional semantic descriptions of images. Our results show the advantages of using output embeddings and inducing low-dimensional embeddings for handling large state spaces in structured prediction problems. The framework we propose is general enough to consider additional sources of information.

8 Acknowledgments

This work was partly funded by the Spanish MINECO project RobInstruct TIN2014-58178-R and by the ERA-net CHISTERA projects VISEN PCIN- 2013-047 and I-DRESS PCIN-2015-147. The authors are grateful to the Nvidia donation program for its support with GPU cards.

References

[Akata et al.2013] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. 2013. Label-embedding for attribute-based classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Duchi and Singer2009] John Duchi and Yoram Singer. 2009. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research (JMLR)*, 10:2899–2934.

[Farhadi et al.2010] Ali Farhadi, Mohsen Hejrati, MohammadAmin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. 2010. Every picture tells a story: Generating sentences from images. In *Proc. European Conference on Computer Vision (ECCV)*.

[Hodosh et al.2013] Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research (JAIR)*, 47:853–899.

[Jia et al.2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

[Karpathy and Fei-Fei2015] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Kulkarni et al.2011] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. 2011. Baby talk: Understanding and generating image descriptions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Mikolov et al.2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proc. International Conference on Learning Representations (ICLR)*.

[Socher et al.2014] Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association of Computational Linguistics (TACL)*, 2:207–218.

[Vinyals et al.2015] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Weston et al.2010] Jason Weston, Samy Bengio, and Nicolas Usunier. 2010. Large scale image annotation: Learning to rank with joint word-image embeddings. In *Proc. European Conference on Computer Vision (ECCV)*.