Learning Concept Taxonomies from Multi-modal Data Supplementary Material

Hao Zhang[1], **Zhiting Hu**[1], **Yuntian Deng**[1], **Mrinmaya Sachan**[1], **Zhicheng Yan**[2], **Eric P. Xing**[1]
[1]Carnegie Mellon University, [2]UIUC
{hao,zhitingh,yuntiand,mrinmays,epxing}@cs.cmu.edu

The supplementary material is organized as follows. In section 1, we provide an illustration of the model (section 1.1), the derivation of the Gibbs sampler (section 1.2) and the gradient descent updating rule (section 1.3). Section 2 gives more details on the feature design and extraction. As supplementary to the paper, section 3 discloses more implementation details for reproducibility, including the word embedding training (section 3.1) and discussion on the implementation efficiency (section 3.2).

1 Model Derivation

1.1 Illustration of Our Model

Fig 1 illustrates the intuition of our model. Each parent-children group (the green boxes in Fig 1) corresponds to a consistency term which encodes the semantic closeness of all parent-child pairs and sibling pairs within that group. The model encourages the local semantic consistency by factorizing consistency terms of all parent-children groups present in the taxonomy.

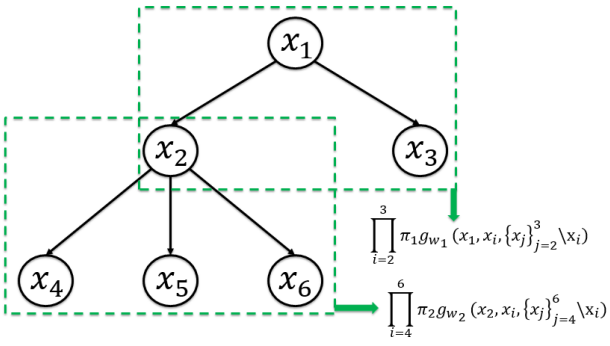

Figure 1: An illustration of our model, which encourages *local semantic consistency.*

1.2 Gibbs Sampling

The probability of a configuration $\boldsymbol{z}$ is defined as

$$p_w(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\alpha}) \propto \prod_n \Gamma(q_n + \alpha_n) \prod_{x_{n'} \in \boldsymbol{c}_n} g_w(x_n, x_{n'}, \boldsymbol{c}_n \backslash x_{n'}) \cdot \mathbf{1}(\boldsymbol{z}). \tag{1}$$

To sample the parent index z_n of category x_n, conditioned on the structure of the rest nodes, we have

$$\begin{aligned} p(z_n = m|\boldsymbol{z}\backslash z_n, \cdot) \propto & \prod_{t \neq m} \Gamma(q_t^{-n} + \alpha_t) \prod_{x_{n'} \in \boldsymbol{c}_t \backslash x_n} g_w(x_t, x_{n'}, \boldsymbol{c}_t \backslash x_n) \\ & \cdot \Gamma(q_m^{-n} + 1 + \alpha_t) \prod_{x_{n'} \in \boldsymbol{c}_m \cup \{x_n\}} g_w(x_m, x_{n'}, \boldsymbol{c}_m \cup \{x_n\}) \\ & \cdot \mathbf{1}(z_n = m, \boldsymbol{z}\backslash z_n), \end{aligned}$$

where q_m is the number of children of category m; the superscript $-n$ denotes the number excluding x_n. To simplify the sampling procedure, we divide $p(z_n = m|\boldsymbol{z}\backslash z_n, \cdot)$ with the "likelihood" of the whole structure excluding x_n, i.e.

$$p(\boldsymbol{z}\backslash z_n|\cdot) \propto \prod_t \Gamma(q_t^{-n} + \alpha_t) \prod_{x_{n'} \in \boldsymbol{c}_t \backslash x_n} g_w(x_t, x_{n'}, \boldsymbol{c}_t \backslash x_n),$$

which is independent of the value of z_n. This leads to our sampling formula as in Eq 3 of the paper

$$p(z_n = m|\boldsymbol{z}\backslash z_n, \cdot) \propto \mathbf{1}(z_n = m, \boldsymbol{z}\backslash z_n) \cdot (q_m^{-n} + \alpha_m) \cdot \frac{\prod_{x_{n'} \in \boldsymbol{c}_m \cup \{x_n\}} g_w(x_m, x_{n'}, \boldsymbol{c}_m \cup \{x_n\})}{\prod_{x_{n'} \in \boldsymbol{c}_m \backslash x_n} g_w(x_m, x_{n'}, \boldsymbol{c}_m \backslash x_n)}.$$

1.3 Gradient Descent

Our training algorithm updates $\boldsymbol{w}$ through maximum likelihood estimation. As we employ an exponential form for the local consistency function $g_w(\cdot) = \exp(\boldsymbol{w}_{d(\cdot)}^\top \boldsymbol{f})$ (where we have simplified the notations to avoid cluttering the notations), the model defined in Eq 1 can be seen to have a log-linear form with respect to $\boldsymbol{w}_{d(\cdot)}$. For the weights $\boldsymbol{w}_l$ of layer l, all the terms in Eq 1, except $g_{w_l}(\cdot)$

for the nodes in the lth layer, are independent of $\boldsymbol{w}_l$, and we denote them as $C_{\boldsymbol{z}}$. Thus we have

$$\log p_w(\tilde{\boldsymbol{z}}|\boldsymbol{x}, \boldsymbol{\alpha}) = \log \frac{C_{\tilde{\boldsymbol{z}}} \exp\{\boldsymbol{w}_l^\top \boldsymbol{f}_{\tilde{\boldsymbol{z}},l}\}}{\sum_{\boldsymbol{z}} C_{\boldsymbol{z}} \exp\{\boldsymbol{w}_l^\top \boldsymbol{f}_{\boldsymbol{z},l}\}},$$

where $\tilde{\boldsymbol{z}}$ is the gold taxonomy from training data, and $\boldsymbol{f}_{\boldsymbol{z},l} = \sum_{n:\ell(x_n)=l} \boldsymbol{f}(x_{z_n}, x_n, \boldsymbol{c}_n \backslash x_n)$ is the sum over the node feature vectors of layer l in taxonomy $\boldsymbol{z}$. Take derivative with respect to $\boldsymbol{w}_l$ we obtain the gradient

$$\begin{aligned}\delta \boldsymbol{w}_l &= \boldsymbol{f}_{\tilde{\boldsymbol{z}},l} - \sum_{\boldsymbol{z}} \frac{C_{\boldsymbol{z}} \exp\{\boldsymbol{w}_l^\top \boldsymbol{f}_{\boldsymbol{z},l}\}}{\sum_{\boldsymbol{z}'} C_{\boldsymbol{z}'} \exp\{\boldsymbol{w}_l^\top \boldsymbol{f}_{\boldsymbol{z}',l}\}} \boldsymbol{f}_{\boldsymbol{z},l} + \log C_{\tilde{\boldsymbol{z}}} \\ &= \sum_{n:\ell(x_n)=l} \{\boldsymbol{f}(x_{\tilde{z}_n}, x_n, \tilde{\boldsymbol{c}}_n \backslash x_n) - \mathbb{E}_p[\boldsymbol{f}(x_{z_n}, x_n, \boldsymbol{c}_n \backslash x_n)]\} \\ &\quad + \log C_{\tilde{\boldsymbol{z}}}.\end{aligned}$$

The expectation is approximated by collecting a set of samples using the Gibbs sampler as described above, and then averaging over them.

2 Feature Extraction

In this section, we further elaborate the procedures how to extract the features, as complementary to the descriptions in section 4 of our paper.

2.1 Parent-child Word-word Relation Feature (PC-T1)

Following Fu et al. (2014), we first learn C^{tt} word-word projection matrices $\{\boldsymbol{\Phi}_c^{tt}\}_{c=1}^{C^{tt}}$ using all pairwise relations from the training hierarchies, where C^{tt} is the number of clusters chosen by cross validation (Fu et al., 2014), and the superscript "tt" denotes text to text (word to word). Then, we compute the distance $d = \|\boldsymbol{\Phi}_c^{tt} \boldsymbol{v}_{t_{n'}} - \boldsymbol{v}_{t_n}\|_2$, where $\boldsymbol{v}_{t_{n'}}$ and $\boldsymbol{v}_{t_n}$ are the word embedding of the child and parent category, respectively, and $\boldsymbol{\Phi}_c^{tt}$ is the projection matrix for the pair $\{\boldsymbol{v}_{t_{n'}}, \boldsymbol{v}_{t_n}\}$, whose index $c \in \{1, 2, \ldots, C^{tt}\}$ is determined by cluster assignment of the pair (see more details in (Fu et al., 2014)). Then, we quantize d into a histogram lying on $[u, v]$ with k bins, thus produce a k-dimensional vector as the feature vector.

2.2 Parent-child Image-word Relation Feature (PC-V2)

We already elaborate how PC-V2 feature is extracted in the paper. Here we provide more detailed implementation notes.

We firstly $\ell 2$-normalize the mean image vector $\overline{\boldsymbol{v}}_{\boldsymbol{i}_{n'}}$ of the child category $x_{n'}$. We then learn the image-word projection matrices $\{\boldsymbol{\Phi}_c^{it}\}_{c=1}^{C^{it}}$ to project $\overline{\boldsymbol{v}}_{\boldsymbol{i}_{n'}}$ to $\boldsymbol{v}_{t_n}$, where $\boldsymbol{v}_{t_n}$ is the word embedding of the parent node x_n, C^{it} is the number of clusters and "it" denotes image to word. Then we use the same quantization strategy to extract a k-dimensional vector as the parent-child image-word relation feature. It is noticeable that for category without $\overline{\boldsymbol{v}}_{\boldsymbol{i}_{n'}}$ (without images), we produce a k-dimensional zero vector instead. As each part of the feature is independent with the other, when multiplying with the weights $\boldsymbol{w}$, the counterpart in $\boldsymbol{w}$ is automatically cancelled by multiplying zeroes, contributing nothing to the local semantic consistency term.

2.3 Parent-child Image-image Relation Feature (PC-V1)

As described in section 4.1 of the paper, for image-image relation, we compute the *vissim* which is defined as

$$vissim(x_n, x_m) = \frac{\mathcal{N}(\overline{\boldsymbol{v}}_{\boldsymbol{i}_m}; \overline{\boldsymbol{v}}_{\boldsymbol{i}_n}, \Sigma_n) + \mathcal{N}(\overline{\boldsymbol{v}}_{\boldsymbol{i}_n}; \overline{\boldsymbol{v}}_{\boldsymbol{i}_m}, \Sigma_m)}{2}$$

as the visual similarity between two categories, where the Gaussian of the child category is estimated using all images in that category, yet the Gaussian of the parent category is fit using only the top K images with highest probabilities under the distribution of the child category. This usually results in a relatively small value which is not in the same scale with other features. Hence, we further transform the *vissim* into log scale and rescale it using:

$$d = \frac{s}{\log(vissim(x_n, x_m))}$$

so that a smaller value of d indicates stronger similarity. Then, the visual distance d is quantized into a histogram and a d-dimensional vector is produced as the feature. For nodes without images, a zero vector will be used instead.

2.4 Siblings Image-image Relation Feature (S-V1)

Similar to the parent-child image-image relation feature, we first compute pairwise visual similarity between each pair of siblings (but using all images in that category when fitting the Gaussian). Then, their mean value is quantized as a feature vector.

2.5 Siblings Word-word Relation Feature (S-T1)

Based on the observation that word vectors with a smaller distance are usually semantically closer, we first compute the *cosine distance* between the

word embedding of each pair of siblings, then quantize the mean distance into a histogram to form the feature vector, where the histogram range $[u, v]$ is determined empirically for each feature.

For all features mentioned above, We set the number of bins k to 20 by cross validation.

2.6 Surface Features

For two categories x_n and $x_{n'}$ with category name t_n and $t_{n'}$ respectively, we list the surface features we used as below (Bansal et al., 2014).

- **Ends with**, i.e. whether $t_{n'}$ ends with t_n (e.g. $catshark$ is a sub-category of $shark$).
- **Contains**, i.e. whether $t_{n'}$ contains t_n.
- **Capitalization**, whether $t_{n'}$ and t_n are capitalized. Intuitively, if $t_{n'}$ is capitalized while t_n is not, the probability of x_n being a parent of $x_{n'}$ tends to be low.
- **Suffix match**, whether $t_{n'}$ and t_n share a common suffix with length k, where k is ranged as $k = 1, \ldots, 7$.
- **LCS**, the *longest continuous common substring* of $t_{n'}$ and t_n. The value $2\frac{|LCS|}{|t_{n'}|+|t_n|}$ is quantized into a histogram as a feature vector.
- **Length difference**, i.e. the indicator features for rounded-off and binned values of $2\frac{|t_{n'}|-|t_n|}{|t_{n'}|+|t_n|}$.

3 Implementation Details

3.1 Word Embedding Training

Preprocessing. We first download the entire structure of ImageNet2011 Release. Every category in ImageNet is denoted as a *synset*, and every synset is jointly described by multiple terms [1]. To match every synset against the corpus for word embedding training, we match every descriptive term of the synset, and discard synsets that are not found or those which rarely exist in the corpus.

We implement a tri-tree in order to detect synset terms in the large corpus more efficiently. The time complexity for phrase matching is $O(dn)$, where d is the depth of the tri-tree and n is the number of tokens in the corpus.

[1] A term could be a single word (e.g. apple), or a phrase represented by multiple words (e.g. thresher shark).

Training. Once we determined the mappings between synsets and words in the training corpus, we re-scan the whole corpus and replace the matched words (or phrases) with a unique string *__Synset_id*, where *id* is the ID of the query synset. We use the hierarchical softmax training algorithm (Mikolov et al., 2013) to train 15 iterations for 200-dimensional word embedding. Synsets occurring fewer than 5 times in the corpus are removed.

3.2 Efficiency

Features described above can be classified as *pairwise features* or *group-wise features*. Specifically, all pairwise features, including the *parent-child word-word relation feature (PC-T1)*, *parent-child image-word relation feature (PC-V2)*, *parent-child image-image relation feature (PC-V1)* and *surface features*, can be obtained in $O(1)$ time by precomputation. While, the group-wise features, including the *sibling image-image relation feature (S-V1)* and *siblings word-word relation feature (S-T1)*, can be obtained in linear time.

References

Mohit Bansal, David Burkett, Gerard de Melo, and Dan Klein. 2014. Structured learning for taxonomy induction with belief propagation.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *ACL*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.