Learning Concept Taxonomies from Multi-modal Data

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

Abstract

We study the problem of automatically building hypernym taxonomies from textual and visual data. Previous works in taxonomy induction generally ignore the increasingly prominent visual data, which encode important perceptual semantics. Instead, we propose a probabilistic model for taxonomy induction by jointly leveraging text and images. To avoid hand-crafted feature engineering, we design end-to-end features based on distributed representations of images and words. The model is discriminatively trained given a small set of existing ontologies and is capable of building full taxonomies from scratch for a collection of unseen conceptual label items with associated images. We evaluate our model and features on the WordNet hierarchies, where our system outperforms previous approaches by a large gap.

1 Introduction

Human knowledge is naturally organized as semantic hierarchies. For example, in WordNet (Miller, 1995), specific concepts are categorized and assigned to more general ones, leading to a semantic hierarchical structure (a.k.a taxonomy). A variety of NLP tasks, such as question answering (Harabagiu et al., 2003), document clustering (Hotho et al., 2002) and text generation (Biran and McKeown, 2013) can benefit from the conceptual relationship present in these hierarchies.

Traditional methods of manually constructing taxonomies by experts (e.g. WordNet) and interest communities (e.g. Wikipedia) are either knowledge or time intensive, and the results have limited coverage. Therefore, automatic induction of taxonomies is drawing increasing attention in both NLP and computer vision. On one hand, a number of methods have been developed to build hierarchies based on lexical patterns in text (Yang and Callan, 2009; Snow et al., 2006; Kozareva and Hovy, 2010; Navigli et al., 2011; Fu et al., 2014; Bansal et al., 2014; Tuan et al., 2015). These works generally ignore the rich visual data which encode important perceptual semantics (Bruni et al., 2014) and have proven to be complementary to linguistic information and helpful for many tasks (Silberer and Lapata, 2014; Kiela and Bottou, 2014; Zhang et al., 2015; Chen et al., 2013). On the other hand, researchers have built visual hierarchies by utilizing only visual features (Griffin and Perona, 2008; Yan et al., 2015; Sivic et al., 2008). The resulting hierarchies are limited in interpretability and usability for knowledge transfer.

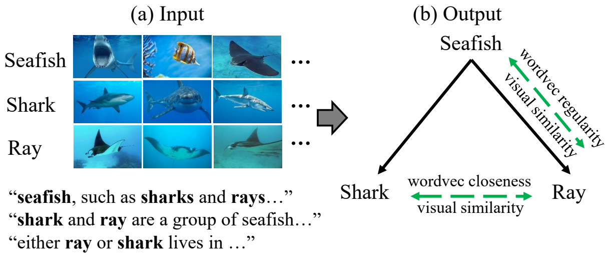
(a) Input
Seafish
Shark
Ray
"**seafish**, such as **sharks** and **rays**…"
"**shark** and **ray** are a group of seafish…"
"either **ray** or **shark** lives in …"
(b) Output
Seafish
wordvec regularity
visual similarity
Shark
Ray
wordvec closeness
visual similarity

Figure 1: An overview of our system. (a) Input: a collection of label items, represented by text and images; (b) Output: we build a taxonomy from scratch by extracting features based on distributed representations of text and images.

Hence, we propose to combine both visual and textual knowledge to automatically build taxonomies. We induce *is-a* taxonomies by supervised learning from existing entity ontologies where each concept category (entity) is associated with images, either from existing dataset (e.g. ImageNet (Deng et al., 2009)) or retrieved from the web using search engines, as illustrated in Fig 1. Such a scenario is realistic and can be extended to a variety of tasks; for example, in knowledge base

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1791–1801,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

construction (Chen et al., 2013), text and image collections are readily available but label relations among categories are to be uncovered. In large-scale object recognition, automatically learning relations between labels can be quite useful (Deng et al., 2014; Zhao et al., 2011).

Both textual and visual information provide important cues for taxonomy induction. Fig 1 illustrates this via an example. The parent category *seafish* and its two child categories *shark* and *ray* are closely related as: (1) there is a hypernym-hyponym (*is-a*) relation between the words "seafish" and "shark"/"ray" through text descriptions like "...seafish, such as shark and ray...", "...shark and ray are a group of seafish..."; (2) images of the close neighbors, e.g., *shark* and *ray* are usually visually similar and images of the child, e.g. *shark/ray* are similar to a subset of images of *seafish*. To effectively capture these patterns, in contrast to previous works that rely on various hand-crafted features (Chen et al., 2013; Bansal et al., 2014), we extract features by leveraging the *distributed representations* that embed images (Simonyan and Zisserman, 2014) and words (Mikolov et al., 2013) as compact vectors, based on which the semantic closeness is directly measured in vector space. Further, we develop a probabilistic framework that integrates the rich multi-modal features to induce "is-a" relations between categories, encouraging *local semantic consistency* that each category should be visually and textually close to its parent and siblings.

In summary, this paper has the following contributions: (1) We propose a novel probabilistic Bayesian model (Section 3) for taxonomy induction by jointly leveraging textual and visual data. The model is discriminatively trained and can be directly applied to build a taxonomy from scratch for a collection of semantic labels. (2) We design novel features (Section 4) based on general-purpose distributed representations of text and images to capture both textual and visual relations between labels. (3) We evaluate our model and features on the ImageNet hierarchies with two different taxonomy induction tasks (Section 5). We achieve superior performance on both tasks and improve the F_1 score by 2x in the *taxonomy construction* task, compared to previous approaches. Extensive comparisons demonstrate the effectiveness of integrating visual features with language features for taxonomy induction. We also provide qualitative analysis on our features, the learned model, and the taxonomies induced to provide further insights (Section 5.3).

2 Related Work

Many approaches have been recently developed that build hierarchies purely by identifying either lexical patterns or statistical features in text corpora (Yang and Callan, 2009; Snow et al., 2006; Kozareva and Hovy, 2010; Navigli et al., 2011; Zhu et al., 2013; Fu et al., 2014; Bansal et al., 2014; Tuan et al., 2014; Tuan et al., 2015; Kiela et al., 2015). The approaches in Yang and Callan (2009) and Snow et al. (2006) assume a starting incomplete hierarchy and try to extend it by inserting new terms. Kozareva and Hovy (2010) and Navigli et al. (2011) first find leaf nodes and then use lexical patterns to find intermediate terms and all the attested hypernymy links between them. In (Tuan et al., 2014), syntactic contextual similarity is exploited to construct the taxonomy, while Tuan et al. (2015) go one step further to consider trustiness and collective synonym/contrastive evidence. Different from them, our model is discriminatively trained with multi-modal data. The works of Fu et al. (2014) and Bansal et al. (2014) use similar language-based features as ours. Specifically, in (Fu et al., 2014), linguistic regularities between pretrained word vectors (Mikolov et al., 2013) are modeled as projection mappings. The trained projection matrix is then used to induce pairwise hypernym-hyponym relations between words. Our features are partially motivated by Fu et al. (2014), but we jointly leverage both textual and visual information. In Kiela et al. (2015), both textual and visual evidences are exploited to detect pairwise lexical entailments. Our work is significantly different as our model is optimized over the whole taxonomy space rather than considering only word pairs separately. In (Bansal et al., 2014), a structural learning model is developed to induce a globally optimal hierarchy. Compared with this work, we exploit much richer features from both text and images, and leverage distributed representations instead of hand-crafted features.

Several approaches (Griffin and Perona, 2008; Bart et al., 2008; Marszałek and Schmid, 2008) have also been proposed to construct visual hierarchies from image collections. In (Bart et al., 2008), a nonparametric Bayesian model is developed to group images based on low-level features.

In (Griffin and Perona, 2008) and (Marszałek and Schmid, 2008), a visual taxonomy is built to accelerate image categorization. In (Chen et al., 2013), only binary object-object relations are extracted using co-detection matrices. Our work differs from all of these as we integrate textual with visual information to construct taxonomies.

Also of note are several works that integrate text and images as evidence for knowledge base autocompletion (Bordes et al., 2011) and zero-shot recognition (Gan et al., 2015; Gan et al., ; Socher et al., 2013). Our work is different because our task is to accurately construct multi-level hyponym-hypernym hierarchies from a set of (seen or unseen) categories.

3 Taxonomy Induction Model

Our model is motivated by the key observation that in a semantically meaningful taxonomy, a category tends to be closely related to its children as well as its siblings. For instance, there exists a hypernym-hyponym relation between the name of category *shark* and that of its parent *seafish*. Besides, images of *shark* tend to be visually similar to those of *ray*, both of which are seafishes. Our model is thus designed to encourage such local semantic consistency; and by jointly considering all categories in the inference, a globally optimal structure is achieved. A key advantage of the model is that we incorporate both visual and textual features induced from distributed representations of images and text (Section 4). These features capture the rich underlying semantics and facilitate taxonomy induction. We further distinguish the relative importance of visual and textual features that could vary in different layers of a taxonomy. Intuitively, visual features would be increasingly indicative in the deeper layers, as sub-categories under the same category of specific objects tend to be visually similar. In contrast, textual features would be more important when inducing hierarchical relations between the categories of general concepts (i.e. in the near-root layers) where visual characteristics are not necessarily similar.

3.1 The Problem

Assume a set of N categories $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\}$, where each category x_n consists of a text term t_n as its name, as well as a set of images $\boldsymbol{i}_n = \{i_1, i_2, \ldots\}$. Our goal is to construct a taxonomy tree T over these categories[1], such that categories of specific object types (e.g. shark) are grouped and assigned to those of general concepts (e.g. seafish). As the categories in $\boldsymbol{x}$ may be from multiple disjoint taxonomy trees, we add a *pseudo* category x_0 as the hyper-root so that the optimal taxonomy is ensured to be a single tree. Let $z_n \in \{1, \ldots, N\}$ be the index of the parent of category x_n, i.e. x_{z_n} is the hypernymic category of x_n. Thus the problem of inducing a taxonomy structure is equivalent to inferring the conditional distribution $p(\boldsymbol{z}|\boldsymbol{x})$ over the set of (latent) indices $\boldsymbol{z} = \{z_1, \ldots, z_n\}$, based on the images and text.

3.2 Model

We formulate the distribution $p(\boldsymbol{z}|\boldsymbol{x})$ through a model which leverages rich multi-modal features. Specifically, let $\boldsymbol{c}_n$ be the set of child nodes of category x_n in a taxonomy encoded by $\boldsymbol{z}$. Our model is defined as

$$p_w(\boldsymbol{z}, \boldsymbol{\pi}|\boldsymbol{x}, \boldsymbol{\alpha}) \propto p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{n=1}^{N} \prod_{x_{n'} \in \boldsymbol{c}_n} \pi_n g_w(x_n, x_{n'}, \boldsymbol{c}_n \backslash x_{n'}) \tag{1}$$

where $g_w(x_n, x_{n'}, \boldsymbol{c}_n \backslash x_{n'})$, defined as

$$g_w(x_n, x_{n'}, \boldsymbol{c}_n \backslash x_{n'}) = \exp\{\boldsymbol{w}_{d(x_{n'})}^{\top} \boldsymbol{f}_{n,n',\boldsymbol{c}_n \backslash x_{n'}}\},$$

measures the semantic consistency between category $x_{n'}$, its parent x_n as well as its siblings indexed by $\boldsymbol{c}_n \backslash x_{n'}$. The function $g_w(\cdot)$ is loglinear with respect to $\boldsymbol{f}_{n,n',\boldsymbol{c}_n \backslash x_{n'}}$, which is the feature vector defined over the set of relevant categories $(x_n, x_{n'}, \boldsymbol{c}_n \backslash x_{n'})$, with $\boldsymbol{c}_n \backslash x_{n'}$ being the set of child categories excluding $x_{n'}$ (Section 4). The simple exponential formulation can effectively encourage close relations among nearby categories in the induced taxonomy. The function has combination weights $\boldsymbol{w} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_L\}$, where L is the maximum depth of the taxonomy, to capture the importance of different features, and the function $d(x_{n'})$ to return the depth of $x_{n'}$ in the current taxonomy. Each layer l $(1 \leq l \leq L)$ of the taxonomy has a specific $\boldsymbol{w}_l$ thereby allowing varying weights of the same features in different layers. The parameters are learned in a *supervised* manner. In eq 1, we also introduce a weight π_n for each node x_n, in order to capture the varying popularity of different categories (in terms of being a parent category). For example, some categories like

[1] We assume T to be a tree. Most existing taxonomies are modeled as trees (Bansal et al., 2014), since a tree helps simplify the construction and ensures that the learned taxonomy is interpretable. With minor modifications, our model also works on non-tree structures.

plant can have a large number of sub-categories, while others such as *stone* have less. We model $\boldsymbol{\pi}$ as a multinomial distribution with Dirichlet prior $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ to encode any prior knowledge of the category popularity[2]; and the conjugacy allows us to marginalize out $\boldsymbol{\pi}$ analytically to get

$$\begin{aligned} p_w(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\alpha}) &\propto \int p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{n=1}^{N} \prod_{x_{n'}\in \boldsymbol{c}_n} \pi_n g_w(x_n, x_{n'}, \boldsymbol{c}_n \backslash x_{n'}) d\boldsymbol{\pi} \\ &\propto \prod_n \Gamma(q_n + \alpha_n) \prod_{x_{n'}\in \boldsymbol{c}_n} g_w(x_n, x_{n'}, \boldsymbol{c}_n \backslash x_{n'}) \end{aligned} \tag{2}$$

where q_n is the number of children of category x_n.

Next, we describe our approach to infer the expectation for each z_n, and based on that select a particular taxonomy structure for the category nodes $\boldsymbol{x}$. As $\boldsymbol{z}$ is constrained to be a tree (i.e. cycle without loops), we include with eq 2, an indicator factor $\mathbf{1}(\boldsymbol{z})$ that takes 1 if $\boldsymbol{z}$ corresponds a tree and 0 otherwise. We modify the inference algorithm appropriately to incorporate this constraint.

Inference. Exact inference is computationally intractable due to the normalization constant of eq 2. We therefore use Gibbs Sampling, a procedure for approximate inference. Here we present the sampling formula for each z_n directly, and defer the details to the supplementary material. The sampling procedure is highly efficient because the normalization term and the factors that are irrelevant to z_n are cancelled out. The formula is

$$\begin{aligned} p(z_n = m|\boldsymbol{z}\backslash z_n, \cdot) &\propto \mathbf{1}(z_n = m, \boldsymbol{z}\backslash z_n) \cdot \left(q_m^{-n} + \alpha_m\right) \cdot \\ &\frac{\prod_{x_{n'}\in \boldsymbol{c}_m \cup \{x_n\}} g_w(x_m, x_{n'}, \boldsymbol{c}_m \cup \{x_n\})}{\prod_{x_{n'}\in \boldsymbol{c}_m \backslash x_n} g_w(x_m, x_{n'}, \boldsymbol{c}_m \backslash x_n)}, \end{aligned} \tag{3}$$

where q_m is the number of children of category m; the superscript $-n$ denotes the number excluding x_n. Examining the validity of the taxonomy structure (i.e. the tree indicator) in each sampling step can be computationally prohibitive. To handle this, we restrict the candidate value of z_n in eq 3, ensuring that the new z_n is always a tree. Specifically, given a tree T, we define a *structure operation* as the procedure of detaching one node x_n in T from its parent and appending it to another node x_m which is not a descendant of x_n.

Proposition 1. *(1) Applying a structure operation on a tree T will result in a structure that is still a tree. (2) Any tree structure over the node set $\boldsymbol{x}$ that has the same root node with tree T can be achieved by applying structure operation on T a finite number of times.*

The proof is straightforward and we omit it due to space limitations. We also add a pseudo node x_0 as the fixed root of the taxonomy. Hence by initializing a tree-structured state rooted at x_0 and restricting each updating step as a structure operation, our sampling procedure is able to explore the whole valid tree space.

Output taxonomy selection. To apply the model to discover the underlying taxonomy from a given set of categories, we first obtain the marginals of $\boldsymbol{z}$ by averaging over the samples generated through eq 3, then output the optimal taxonomy $\boldsymbol{z}^*$ by finding the maximum spanning tree (MST) using the Chu-Liu-Edmonds algorithm (Chu and Liu, 1965; Bansal et al., 2014).

Training. We need to learn the model parameters $\boldsymbol{w}_l$ of each layer l, which capture the relative importance of different features. The model is trained using the EM algorithm. Let $\ell(x_n)$ be the depth (layer) of category x_n; and $\tilde{\boldsymbol{z}}$ (siblings $\tilde{\boldsymbol{c}}_n$) denote the gold structure in training data. Our training algorithm updates $\boldsymbol{w}$ through maximum likelihood estimation, wherein the gradient of $\boldsymbol{w}_l$ is (see the supplementary materials for details):

$$\delta \boldsymbol{w}_l = \sum_{n:\ell(x_n)=l} \{\boldsymbol{f}(x_{\tilde{z}_n}, x_n, \tilde{\boldsymbol{c}}_n \backslash x_n) - \mathbb{E}_p[\boldsymbol{f}(x_{z_n}, x_n, \boldsymbol{c}_n \backslash x_n)]\},$$

which is the net difference between gold feature vectors and expected feature vectors as per the model. The expectation is approximated by collecting samples using the sampler described above and averaging them.

4 Features

In this section, we describe the feature vector $\boldsymbol{f}$ used in our model, and defer more details in the supplementary material. Compared to previous taxonomy induction works which rely purely on linguistic information, we exploit both perceptual and textual features to capture the rich spectrum of semantics encoded in images and text. Moreover, we leverage the *distributed representations* of images and words to construct compact and effective features. Specifically, each image i is represented as an embedding vector $\boldsymbol{v}_i \in \mathbb{R}^a$ extracted by deep convolutional neural networks. Such image representation has been successfully applied in various vision tasks. On the other hand, the category name t is represented by its word embedding $\boldsymbol{v}_t \in \mathbb{R}^b$, a low-dimensional dense vector induced by the Skip-gram model (Mikolov et

[2] $\boldsymbol{\alpha}$ could be estimated using training data.

al., 2013) which is widely used in diverse NLP applications too. Then we design $\boldsymbol{f}(x_n, x_{n'}, \boldsymbol{c}_n \backslash x_{n'})$ based on the above image and text representations. The feature vector $\boldsymbol{f}$ is used to measure the local semantic consistency between category $x_{n'}$ and its parent category x_n as well as its siblings $\boldsymbol{c}_n \backslash x_{n'}$.

4.1 Image Features

Sibling similarity. As mentioned above, close neighbors in a taxonomy tend to be visually similar, indicating that the embedding of images of sibling categories should be close to each other in the vector space $\mathbb{R}^a$. For a category x_n and its image set $\boldsymbol{i}_n$, we fit a Gaussian distribution $\mathcal{N}(\overline{\boldsymbol{v}}_{\boldsymbol{i}_n}, \Sigma_n)$ to the image vectors, where $\overline{\boldsymbol{v}}_{\boldsymbol{i}_n} \in \mathbb{R}^a$ is the mean vector and $\Sigma_n \in \mathbb{R}^{a \times a}$ is the covariance matrix. For a sibling category x_m of x_n, we define the visual similarity between x_n and x_m as

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

which is the average probability of the mean image vector of one category under the Gaussian distribution of the other. This takes into account not only the distance between the mean images, but also the closeness of the images of each category. Accordingly, we compute the visual similarity between $x_{n'}$ and the set $\boldsymbol{c}_n \backslash x_{n'}$ by averaging:

$$vissim(x_{n'}, \boldsymbol{c}_n \backslash x_{n'}) = \frac{\sum_{x_m \in \boldsymbol{c}_n \backslash x_{n'}} vissim(x_{n'}, x_m)}{|\boldsymbol{c}_n| - 1}.$$

We then bin the values of $vissim(x_{n'}, \boldsymbol{c}_n \backslash x_{n'})$ and represent it as an one-hot vector, which constitutes $\boldsymbol{f}$ as a component named as *siblings image-image relation feature* (denoted as *S-V1*[3]).

Parent prediction. Similar to feature S-V1, we also create the similarity feature between the image vectors of the parent and child, to measure their visual similarity. However, the parent node is usually a more general concept than the child, and it usually consists of images that are not necessarily similar to its child. Intuitively, by narrowing the set of images to those that are most similar to its child improves the feature. Therefore, different from S-V1, when estimating the Gaussian distribution of the parent node, we only use the top K images with highest probabilities under the Gaussian distribution of the child node. We empirically show in section 5.3 that choosing an appropriate K consistently boosts the performance. We name this feature as *parent-child image-image relation feature* (denoted as *PC-V1*).

Further, inspired by the linguistic regularities of word embedding, i.e. the hypernym-hyponym relationship between words can be approximated by a linear projection operator between word vectors (Mikolov et al., 2013; Fu et al., 2014), we design a similar strategy to (Fu et al., 2014) between images and words so that the parent can be "predicted" given the image embedding of its child category and the projection matrix. Specifically, let $(x_n, x_{n'})$ be a parent-child pair in the training data, we learn a projection matrix $\boldsymbol{\Phi}$ which minimizes the distance between $\boldsymbol{\Phi}\overline{\boldsymbol{v}}_{\boldsymbol{i}_{n'}}$ (i.e. the projected mean image vector $\overline{\boldsymbol{v}}_{\boldsymbol{i}_{n'}}$ of the child) and $\boldsymbol{v}_{t_n}$ (i.e. the word embedding of the parent):

$$\boldsymbol{\Phi}^* = \underset{\boldsymbol{\Phi}}{\operatorname{argmin}} \frac{1}{N} \sum_n \|\boldsymbol{\Phi}\overline{\boldsymbol{v}}_{\boldsymbol{i}_{n'}} - \boldsymbol{v}_{t_n}\|_2^2 + \lambda \|\boldsymbol{\Phi}\|_1,$$

where N is the number of parent-child pairs in the training data. Once the projection matrix has been learned, the similarity between a child node $x_{n'}$ and its parent x_n is computed as $\|\boldsymbol{\Phi}\overline{\boldsymbol{v}}_{\boldsymbol{i}_{n'}} - \boldsymbol{v}_{t_n}\|$, and we also create an one-hot vector by binning the feature value. We call this feature as *parent-child image-word relation feature* (*PC-V2*).

4.2 Word Features

We briefly introduce the text features employed. More details about the text feature extraction could be found in the supplementary material.

Word embedding features.d PC-V1, We induce features using word vectors to measure both sibling-sibling and parent-child closeness in text domain (Fu et al., 2014). One exception is that, as each category has only one word, the sibling similarity is computed as the cosine distance between two word vectors (instead of mean vectors). This will produce another two parts of features, *parent-child word-word relation feature* (*PC-T1*) and *siblings word-word relation feature* (*S-T1*).

Word surface features. In addition to the embedding-based features, we further leverage lexical features based on the surface forms of child/parent category names. Specifically, we employ the *Capitalization*, *Ends with*, *Contains*, *Suffix match*, *LCS* and *Length different* features, which are commonly used in previous works in taxonomy induction (Yang and Callan, 2009; Bansal et al., 2014).

5 Experiments

We first disclose our implementation details in section 5.1 and the supplementary material for bet-

[3] S: sibling, PC: parent-child, V: visual, T: textual.

ter reproducibility. We then compare our model with previous state-of-the-art methods (Fu et al., 2014; Bansal et al., 2014) with two taxonomy induction tasks. Finally, we provide analysis on the weights and taxonomies induced.

5.1 Implementation Details

Dataset. We conduct our experiments on the ImageNet2011 dataset (Deng et al., 2009), which provides a large collection of category items (synsets), with associated images and a label hierarchy (sampled from WordNet) over them. The original ImageNet taxonomy is preprocessed, resulting in a tree structure with 28231 nodes.

Word embedding training. We train word embedding for synsets by replacing each word/phrase in a synset with a unique token and then using Google's word2vec tool (Mikolov et al., 2013). We combine three public available corpora together, including the latest Wikipedia dump (Wikipedia, 2014), the One Billion Word Language Modeling Benchmark (Chelba et al., 2013) and the UMBC webbase corpus (Han et al., 2013), resulting in a corpus with total 6 billion tokens. The dimension of the embedding is set to 200.

Image processing. we employ the ILSVRC12 pre-trained convolutional neural networks (Simonyan and Zisserman, 2014) to embed each image into the vector space. Then, for each category x_n with images, we estimate a multivariate Gaussian parameterized by $\mathcal{N}_{x_n} = (\mu_{x_n}, \Sigma_{x_n})$, and constrain Σ_{x_n} to be diagonal to prevent overfitting. For categories with very few images, we only estimate a mean vector μ_{x_n}. For nodes that do not have images, we ignore the visual feature.

Training configuration. The feature vector is a concatenation of 6 parts, as detailed in section 4. All pairwise distances are precomputed and stored in memory to accelerate Gibbs sampling. The initial learning rate for gradient descent in the M step is set to 0.1, and is decreased by a fraction of 10 every 100 EM iterations.

5.2 Evaluation

5.2.1 Experimental Settings

We evaluate our model on three subtrees sampled from the ImageNet taxonomy. To collect the subtrees, we start from a given root (e.g. consumer goods) and traverse the full taxonomy using BFS, and collect all descendant nodes within a depth h (number of nodes in the longest path). We vary h to get a series of subtrees with increasing heights $h \in \{4, 5, 6, 7\}$ and various scales (maximally 1326 nodes) in different domains. The statistics of the evaluation sets are provided in Table 1. To avoid ambiguity, all nodes used in ILSVRC 2012 are removed as the CNN feature extractor is trained on them.

Trees	Tree A	Tree B	Tree C
Synset ID	12638	19919	23733
Name	consumer goods	animal	food, nutrient
$h = 4$	187	207	572
$h = 5$	362	415	890
$h = 6$	493	800	1166
$h = 7$	524	1386	1326

Table 1: Statistics of our evaluation set. The bottom 4 rows give the number of nodes within each height $h \in \{4, 5, 6, 7\}$. The scale of the threes range from small to large, and there is no overlapping among them.

We design two different tasks to evaluate our model. (1) In the *hierarchy completion* task, we randomly remove some nodes from a tree and use the remaining hierarchy for training. In the test phase, we infer the parent of each removed node and compare it with groundtruth. This task is designed to figure out whether our model can successfully induce hierarchical relations after learning from within-domain parent-child pairs. (2) Different from the previous one, the *hierarchy construction* task is designed to test the generalization ability of our model, i.e. whether our model can learn statistical patterns from one hierarchy and transfer the knowledge to build a taxonomy for another collection of out-of-domain labels. Specifically, we select two trees as the training set to learn $\boldsymbol{w}$. In the test phase, the model is required to build the full taxonomy from scratch for the third tree.

We use *Ancestor* F_1 as our evaluation metric (Kozareva and Hovy, 2010; Navigli et al., 2011; Bansal et al., 2014). Specifically, we measure $F_1 = 2PR/(P+R)$ values of predicted "is-a" relations where the precision (P) and recall (R) are:

$$P = \frac{|\text{isa}_{\text{predicted}} \cap \text{isa}_{\text{gold}}|}{|\text{isa}_{\text{predicted}}|}, R = \frac{|\text{isa}_{\text{predicted}} \cap \text{isa}_{\text{gold}}|}{|\text{isa}_{\text{gold}}|}.$$

We compare our method to two previously state-of-the-art models by Fu et al. (2014) and Bansal et al. (2014), which are closest to ours.

Method	$h = 4$	$h = 5$	$h = 6$	$h = 7$
Hierarchy Completion				
Fu2014	0.66	0.42	0.26	0.21
Ours (L)	0.70	0.49	0.45	0.37
Ours (LV)	**0.73**	**0.51**	**0.50**	**0.42**
Hierarchy Construction				
Fu2014	0.53	0.33	0.28	0.18
Bansal2014	0.67	0.53	0.43	0.37
Ours (L)	0.58	0.41	0.36	0.30
Ours (LB)	0.68	0.55	0.45	0.40
Ours (LV)	0.66	0.52	0.42	0.34
Ours (LVB - E)	0.68	0.55	0.44	0.39
Ours (LVB)	**0.70**	**0.57**	**0.49**	**0.43**

Table 2: Comparisons among different variants of our model, Fu et al. (2014) and Bansal et al. (2014) on two tasks. The ancestor-F_1 scores are reported.

5.2.2 Results

Hierarchy completion. In the *hierarchy completion* task, we split each tree into 70% nodes for training and 30% for test, and experiment with different h. We compare the following three systems: (1) *Fu2014*[4] (Fu et al., 2014); (2) *Ours (L)*: Our model with only language features enabled (i.e. surface features, parent-child word-word relation feature and siblings word-word relation feature); (3) *Ours (LV)*: Our model with both language features and visual features [5]. The average performance on three trees are reported at Table 2. We observe that the performance gradually drops when h increases, as more nodes are inserted when the tree grows higher, leading to a more complex and difficult taxonomy to be accurately constructed. Overall, our model outperforms Fu2014 in terms of the F_1 score, even without visual features. In the most difficult case with $h = 7$, our model still holds an F_1 score of 0.42 (2× of Fu2014), demonstrating the superiority of our model.

Hierarchy construction. The hierarchy construction task is much more difficult than hierarchy completion task because we need to build a taxonomy from scratch given only a hyper-root. For this task, we use a leave-one-out strategy, i.e. we train our model on every two trees and test on the third, and report the average performance in Table 2. We compare the following methods: (1) *Fu2014*, (2) *Ours (L)*, and (3) *Ours (LV)*, as described above; (4) *Bansal2014*: The model by Bansal et al. (2014) retrained using our dataset; (5) *Ours (LB)*: By excluding visual features, but including other language features from Bansal et al. (2014); (6) *Ours (LVB)*: Our full model further enhanced with all semantic features from Bansal et al. (2014); (7) *Ours (LVB - E)*: By excluding word embedding-based language features from *Ours (LVB)*.

As shown, on the hierarchy construction task, our model with only language features still outperforms Fu2014 with a large gap (0.30 compared to 0.18 when $h = 7$), which uses similar embedding-based features. The potential reasons are two-fold. First, we take into account not only parent-child relations but also siblings. Second, their method is designed to induce only pairwise relations. To build the full taxonomy, they first identify all possible pairwise relations using a simple thresholding strategy and then eliminate conflicted relations to obtain a legitimate tree hierarchy. In contrast, our model is optimized over the full space of all legitimate taxonomies by taking the *structure operation* in account during Gibbs sampling.

When comparing to Bansal2014, our model with only word embedding-based features underperforms theirs. However, when introducing visual features, our performance is comparable (p-value = 0.058).Furthermore, if we discard visual features but add semantic features from Bansal et al. (2014), we achieve a slight improvement of 0.02 over Bansal2014 (p-value = 0.016), which is largely attributed to the incorporation of word embedding-based features that encode high-level linguistic regularity. Finally, if we enhance our full model with all semantic features from Bansal et al. (2014), our model outperforms theirs by a gap of 0.04 (p-value < 0.01), which justifies our intuition that perceptual semantics underneath visual contents are quite helpful.

5.3 Qualitative Analysis

In this section, we conduct qualitative studies to investigate *how* and *when* the visual information helps the taxonomy induction task.

Contributions of visual features. To evaluate the contribution of each part of the visual features to the final performance, we train our model jointly with textual features and different combinations of visual features, and report the ancestor-F_1 scores. As shown in Table 3. When incorporating the feature S-V1, the performance is substantially boosted by a large gap at all heights, show-

[4] We tried different parameter settings for the number of clusters C and the identification threshold δ, and reported the best performance we achieved.

[5] In the comparisons to (Fu et al., 2014) and (Bansal et al., 2014), we simply set $K = \infty$, *i.e.* we use all available images of the parent category to estimate the PC-V1 feature.

S-V1	PC-V1	PC-V2	h = 4	h = 5	h = 6	h = 7
			0.58	0.41	0.36	0.30
✓			0.63	0.48	0.40	0.32
	✓		0.61	0.44	0.38	0.31
		✓	0.60	0.42	0.37	0.31
✓	✓		0.65	**0.52**	0.41	0.33
✓	✓	✓	**0.66**	**0.52**	**0.42**	**0.34**

Table 3: The performance when different combinations of visual features are enabled.

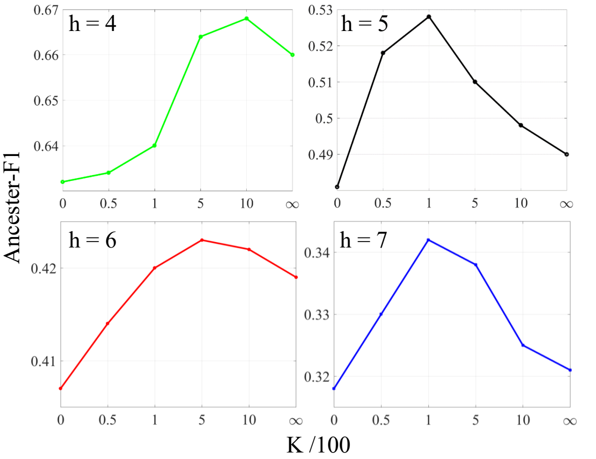

Figure 2: The Ancestor-F_1 scores changes over K (number of images used in the PC-V1 feature) at different heights. The values in the x-axis are $K/100$; $K = \infty$ means all images are used.

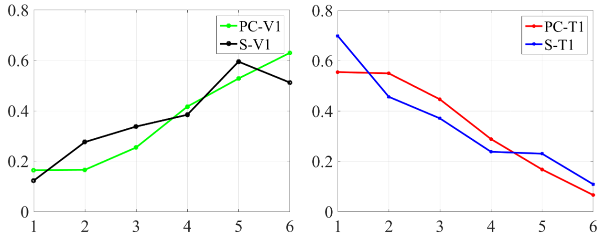

Figure 3: Normalized weights of each feature v.s. the layer depth.

ing that visual similarity between sibling nodes is a strong evidence for taxonomy induction. It is intuitively plausible, as it is highly likely that two specific categories share a common (and more general) parent category if similar visual contents are observed between them. Further, adding the PC-V1 feature gains us a better improvement than adding PC-V2, but both minor than S-V1.

Compared to that of siblings, the visual similarity between parents and children does not strongly holds all the time. For example, images of *Terrestrial animal* are only partially similar to those of *Feline*, because the former one contains the later one as a subset. Our feature captures this type of "contain" relation between parents and children by considering only the top-K images from the parent category that have highest probabilities under the Gaussian distribution of the child category. To see this, we vary K while keep all other settings, and plot the F_1 scores in Fig 2. We observe a trend that when we gradually increase K, the performance goes up until reaching some maximal; It then slightly drops (or oscillates) even when more images are available, which confirms with our feature design that only top images should be considered in parent-child visual similarity.

Overall, the three visual features complement each other, and achieve the highest performance when combined.

Visual representations. To investigate how the image representations affect the final performance, we compare the ancestor-F1 score when different pre-trained CNNs are used for visual feature extraction. Specifically, we employ both the CNN-128 model (128 dimensional feature with 15.6% top-5 error on ILSVRC12) and the VGG-16 model (4096 dimensional feature with 7.5% top-5 error) by Simonyan and Zisserman (2014), but only observe a slight improvement of 0.01 on the ancestor-F1 score for the later one.

Relevance of textual and visual features v.s. depth of tree. Compared to Bansal et al. (2014), a major difference of our model is that different layers of the taxonomy correspond to different weights $\boldsymbol{w}_l$, while in (Bansal et al., 2014) all layers share the same weights. Intuitively, introducing layer-wise $\boldsymbol{w}$ not only extends the model capacity, but also differentiates the importance of each feature at different layers. For example, the images of two specific categories, such as *shark* and *ray*, are very likely to be visually similar. However, when the taxonomy goes from bottom to up (specific to general), the visual similarity is gradually undermined — images of *fish* and *terrestrial animal* are not necessarily similar any more. Hence, it is necessary to privatize the weights $\boldsymbol{w}$ for different layers to capture such variations, i.e. the visual features become more and more evident from shallow to deep layers, while the textual counterparts, which capture more abstract concepts, relatively grow more indicative oppositely from specific to general.

To visualize the variations across layers, for each feature component, we fetch its correspond-

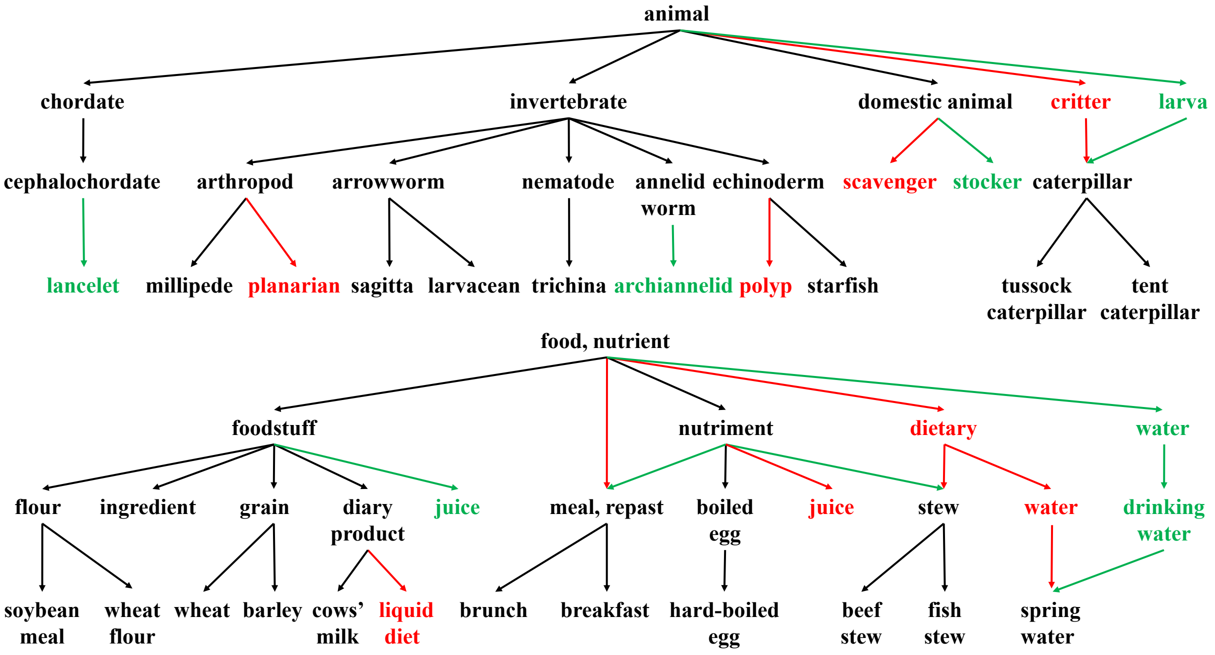

Figure 4: Excerpts of the prediction taxonomies, compared to the groundturth. Edges marked as red and green are false predictions and unpredicted groundtruth links, respectively.

ing block in $\boldsymbol{w}$ as V. Then, we average $|V|$ and observe how its values change with the layer depth h. For example, for the parent-child word-word relation feature, we first fetch its corresponding weights V from $\boldsymbol{w}$ as a 20×6 matrix, where 20 is the feature dimension and 6 is the number of layers. We then average its absolute values[6] in column and get a vector v with length 6. After ℓ_2 normalization, the magnitude of each entry in v directly reflects the relative importance of the feature as an evidence for taxonomy induction. Fig 3(b) plots how their magnitudes change with h for every feature component averaged on three train/test splits. It is noticeable that for both word-word relations (S-T1, PC-T1), their corresponding weights slightly decrease as h increases. On the contrary, the image-image relation features (S-V1, PC-V1) grows relatively more prominent. The results verify our conjecture that when the category hierarchy goes deeper into more specific classes, the visual similarity becomes relatively more indicative as an evidence for taxonomy induction.

Visualizing results. Finally, we visualize some excerpts of our predicted taxonomies, as compared to the groundtruth in Fig 4.

6 Conclusion

In this paper, we study the problem of automatically inducing semantically meaningful concept taxonomies from multi-modal data. We propose a probabilistic Bayesian model which leverages distributed representations for images and words. We compare our model and features to previous ones on two different tasks using the ImageNet hierarchies, and demonstrate superior performance of our model, and the effectiveness of exploiting visual contents for taxonomy induction. We further conduct qualitative studies and distinguish the relative importance of visual and textual features in constructing various parts of a taxonomy.

Acknowledgements

We would like to thank anonymous reviewers for their valuable feedback. We would also like to thank Mohit Bansal for helpful suggestions. We thank NVIDIA for GPU donations. The work is supported by NSF Big Data IIS1447676.

References

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

Evgeniy Bart, Ian Porteous, Pietro Perona, and Max

[6] We take the absolute value because we only care about the relevance of the feature as an evidence for taxonomy induction, but note that the weight can either encourage (positive) or discourage (negative) connections of two nodes.

Welling. 2008. Unsupervised learning of visual taxonomies. In *CVPR*.

Or Biran and Kathleen McKeown. 2013. Classifying taxonomic relations between pairs of wikipedia articles.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Conference on Artificial Intelligence*, number EPFL-CONF-192344.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. 2013. Neil: Extracting visual knowledge from web data. In *CVPR*.

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On shortest arborescence of a directed graph. *Scientia Sinica*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. 2014. Large-scale object classification using label relation graphs. In *ECCV*.

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

Chuang Gan, Yi Yang, Linchao Zhu, Deli Zhao, and Yueting Zhuang. Recognizing an action using its name: A knowledge-based approach. *International Journal of Computer Vision*, pages 1–17.

Chuang Gan, Ming Lin, Yi Yang, Yueting Zhuang, and Alexander G Hauptmann. 2015. Exploring semantic inter-class relationships (SIR) for zero-shot action recognition. In *AAAI*.

Gregory Griffin and Pietro Perona. 2008. Learning and using taxonomies for fast visual categorization. In *CVPR*.

Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. Umbc ebiquity-core: Semantic textual similarity systems. *Atlanta, Georgia, USA*.

Sanda M Harabagiu, Steven J Maiorano, and Marius A Pasca. 2003. Open-domain textual question answering techniques. *Natural Language Engineering*.

Andreas Hotho, Alexander Maedche, and Steffen Staab. 2002. Ontology-based text document clustering.

Douwe Kiela and Léon Bottou. 2014. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *EMNLP*.

Douwe Kiela, Laura Rimell, Ivan Vulic, and Stephen Clark. 2015. Exploiting image generality for lexical entailment detection. In *ACL*.

Zornitsa Kozareva and Eduard Hovy. 2010. A semi-supervised method to learn and construct taxonomies using the web. In *EMNLP*.

Marcin Marszałek and Cordelia Schmid. 2008. Constructing category hierarchies for visual recognition. In *ECCV*.

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

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*.

Roberto Navigli, Paola Velardi, and Stefano Faralli. 2011. A graph-based algorithm for inducing lexical taxonomies from scratch. In *IJCAI*.

Carina Silberer and Mirella Lapata. 2014. Learning grounded meaning representations with autoencoders. In *ACL*.

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

Josef Sivic, Bryan C Russell, Andrew Zisserman, William T Freeman, and Alexei A Efros. 2008. Unsupervised discovery of visual object class hierarchies. In *CVPR*.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *ACL*.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943.

Luu Anh Tuan, Jung-jae Kim, and Ng See Kiong. 2014. Taxonomy construction using syntactic contextual evidence. In *EMNLP*.

Luu Anh Tuan, Jung-jae Kim, and Ng See Kiong. 2015. Incorporating trustiness and collective synonym/contrastive evidence into taxonomy construction.

Wikipedia. 2014. `https://dumps.wikimedia.org/enwiki/20141208/`.

Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. 2015. Hd-cnn: Hierarchical deep convolutional neural networks for large scale visual recognition. In *ICCV*.

Hui Yang and Jamie Callan. 2009. A metric-based framework for automatic taxonomy induction. In *ACL-IJCNLP*.

Hao Zhang, Gunhee Kim, and Eric P. Xing. 2015. Dynamic topic modeling for monitoring market competition from online text and image data. In *KDD*.

Bin Zhao, Fei Li, and Eric P Xing. 2011. Large-scale category structure aware image categorization. In *NIPS*.

Xingwei Zhu, Zhao-Yan Ming, Xiaoyan Zhu, and Tat-Seng Chua. 2013. Topic hierarchy construction for the organization of multi-source user generated contents. In *SIGIR*.