Sense Discovery via Co-Clustering on Images and Text

Xinlei Chen[†] Alan Ritter[‡] Abhinav Gupta[†] Tom Mitchell[†]
[†]Carnegie Mellon University [‡]Ohio State University
{xinleic,abhinavg,tom.mitchell}@cs.cmu.edu ritter.1492@osu.edu

Abstract

We present a co-clustering framework that can be used to discover multiple semantic and visual senses of a given Noun Phrase (NP). Unlike traditional clustering approaches which assume a one-to-one mapping between the clusters in the text-based feature space and the visual space, we adopt a one-to-many mapping between the two spaces. This is primarily because each semantic sense (concept) can correspond to different visual senses due to viewpoint and appearance variations. Our structure-EM style optimization not only extracts the multiple senses in both semantic and visual feature space, but also discovers the mapping between the senses. We introduce a challenging dataset (CMU Polysemy-30) for this problem consisting of 30 NPs (∼5600 labeled instances out of ∼22K total instances). We have also conducted a large-scale experiment that performs sense disambiguation for ∼2000 NPs.

1. Introduction

Knowledge representation is a classical problem in AI. There have been mammoth efforts such as CYC[1] to build these knowledge bases using human intelligence. Unfortunately, effective construction of broad-coverage knowledge bases still remains an unsolved problem, as manual labeling lacks the richness and scalability required for this task. In recent years, the focus has shifted to building knowledge bases automatically by learning knowledge from free text [8, 1] and images on the internet [9, 17]. While these systems have shown much promise, one issue that limits their performance is the problem of semantic and "visual" polysemy. Polysemy is the capacity for a word or Noun Phrase (NP) to have multiple semantic meanings and visual meanings as well. For example, the noun phrase *Apple* can refer to both the *company* and the *fruit*. Similarly, in the visual world, *Apple* can refer to images of the fruit, the company logo, and even iPhones and iPads. Therefore, handling polysemy by extracting multiple senses of a word/NP is an important problem that needs to be addressed.

One obvious way to handle semantic polysemy is to fall back to human developed knowledge bases such as WordNet [31], Freebase [25] and even Wikipedia [11]. These broad-coverage knowledge bases suffer from the problem of missing information. For example, WordNet has good coverage of common nouns, however it has been criticized for being too fine-grained [41]. In addition it contains very few named entities (people, locations, organizations, *etc*.); Wikipedia and Freebase help to bridge this gap, but a great deal of information is still missing [36]. Furthermore, WordNet or Freebase have little or no information related to visual senses and still require extensive manual labeling to create mappings between semantic and visual senses. In contrast, unstructured data sources such as images and text from the web are much larger and more diverse; which can be readily used to discover both semantic and visual senses.

Instead of relying on manually-compiled resources, we focus on automatically discovering multiple senses of a NP in an unsupervised manner. The common unsupervised paradigm is to represent each noun phrase in terms of text features or image features and then cluster these instances to obtain multiple semantic and visual senses of the NP respectively. Since the semantic and visual senses of a NP are closely related, recent approaches have also attempted jointly clustering images and text. Most joint clustering approaches make the simplifying assumption that there exists a one-to-one mapping between semantic and visual senses of a word. This assumption rarely holds in practice, however. For example, while there are two predominant semantic senses of the word "Apple", there exist multiple visual senses due to appearance variation (green vs. red apples), viewpoint changes, *etc*.

We present a generalized co-clustering algorithm that jointly discovers both semantic and visual senses for a given NP . For a given NP (such as "Apple"), we first download webpages which contain both image and text references to the NP. Each webpage is treated as a datapoint and represented in terms of image and text features. We then use our co-clustering algorithm which clusters data points in image and text feature space separately and learns a one-to-many mapping between the clusters in two feature spaces [2] (See Figure 1). We demonstrate the effectiveness of our ap-

[1] http://en.wikipedia.org/wiki/Cyc

[2] This can also be formulated as hierarchical clustering where higher level nodes correspond to clusters in text space and lower level nodes correspond to clusters in visual space

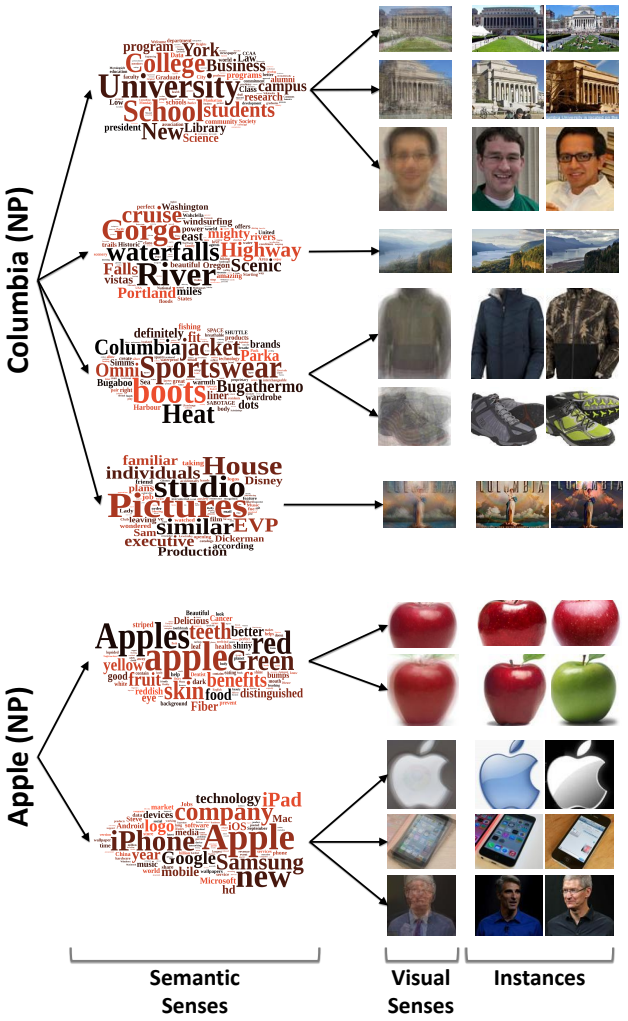

Figure 1: We present a co-clustering algorithm that discovers multiple semantic and visual senses of a given Noun Phrase (NP). In the figure above, we show the multiple senses discovered for the NPs Columbia and Apple. In the case of Columbia, our approach automatically discovers four semantic senses: university, river, sportswear, studio. In case of Apple, it discovers two semantic senses: fruit, company. Our approach also discovers multiple visual senses. For example, the sportswear sense of Columbia corresponds to two visual senses: jacket and shoes. Semantic senses are shown as word clouds with size of each word being proportional to its importance. Visual senses are shown as average images of members belonging to the cluster.

proach using four different experiments including a large scale experiment of co-clustering on ∼2000 NPs. We show how the joint space provides constraints that lead to high purity clusters. But more importantly, this joint learning process allows us to infer an alignment between the semantic and visual senses of the NP.

Why use images and text? We believe that the information in images and text is complementary and one needs to harness both to build a system that robustly discovers multiple senses of a word. For example, using images alone, it is almost impossible to differentiate viewpoint changes from conceptual changes. Similarly, using text based systems alone it is hard to differentiate similar semantic meanings. An example of this is the bike and car meaning of the word "Falcon". In this case, text-based features tends to cluster the bike and the car sense together since both are vehicles but co-clustering in the joint space helps us to differentiate between the two.

Contributions: Our contributions include: (a) We introduce the problem of joint extraction of semantic and visual senses for given noun phrases and provide a novel formulation of this problem. We demonstrate how joint extraction of senses not only improves clustering but also helps us extract relationships between semantic and visual meanings of words. (b) We propose a generalized co-clustering algorithm where the two domains need not have the same granularity of clustering. We achieve this by enforcing a one-to-many constraint during the clustering process instead of a one-to-one mapping. (c) Finally, we introduce a new challenging dataset called CMU Polysemy-30, containing 30NPs for the polysemy problem.

2. Related Work

A significant amount of previous work has investigated the problem of automatically inducing word senses from statistics derived from text corpora [32, 7, 23]. This approach has been quite successful given the small amount of prior knowledge provided: *e.g.*, Yarowsky [46] proposed an unsupervised approach for word sense disambiguation (WSD) but suggested the use of dictionary definitions as seeds. But as pointed out before, most knowledge bases still suffer from a great deal of missing information [36].

Extracting visual senses of polysemous words using web images is an extremely difficult problem. There have been early efforts on automatically training visual classifiers using web data to build large datasets automatically [26, 39, 42, 27, 20], find iconic images [5, 35] and improve image retrieval results [19, 28, 45, 34]. Inspired by the success of mixture models for object detection [18], some recent approaches such as [9, 17] have also explored clustering web data and training detection models. For example, NEIL [9] performs clustering in visual appearance space to generate visual subcategories. But since NEIL only uses visual information, it cannot differentiate between semantic and visual polysemy: that is, it cannot label if two clusters correspond to same semantic meaning. On the other LEVAN [17] uses Google N-grams to first discover different senses for each NP. However, each N-gram leads to a different visual cluster, which results in a lot more clusters than NEIL, *e.g.*, hundreds of senses for the NP "Horse". But similar to NEIL, clustering visually different n-grams ("eating Horse" and "jumping Horse") into one semantic cluster would require using further text information.

To handle these problems, past work has also focused on

using both images and text on the web for discovering visual and semantic senses. For example, Schroff et al. [39] incorporate text features to rerank the images before training visual classifiers. In another work, Berg et al. [4] discover topics using text and then use these topics to cluster the images. However, their approach requires manually selecting the topics for each category. Saenko et. al. [37, 38] presented a model for learning visual senses using images clustered using text, but rely on WordNet's sense inventory. Another approach [44] uses Wikipedia to find the senses of a word. Lucchi et. al [28] used the click-through data and human relevance annotations to learn multiple senses. But the scalability and coverage of such knowledge bases and human annotations is questionable, and therefore in this work we focus on unsupervised approaches. Leoff et al [24] focused on discovering visual senses in completely unsupervised manner by building a joint space of image and text features followed by clustering in this joint space. In addition, Barnard et al. [3] looked at the complementary problem of discovering semantic senses using image data. In our work, we propose an approach to jointly discover multiple semantic and visual senses from web data. We demonstrate that a joint solution (with a one-to-many mapping) allows us to improve the clustering performance and extract relationships between the semantic and visual senses of a NP.

Finally, our work is also closely related to approaches in co-clustering [14, 16] and multi-view clustering based approaches [15]. Previous approaches, however, assume a one-to-one mapping between clusters in two spaces. Instead, we relax this assumption and propose a co-clustering based approach where the mapping between clusters in two domains is one-to-many.

3. Co-Clustering Approach

Given two domains $\mathcal{D}_1$ and $\mathcal{D}_2$, our goal is to jointly cluster the instances in both domains. Previous approaches have tackled this problem by augmenting feature spaces and performing clustering in the joint space. Other approaches have assumed a one-to-one mapping between the clusters in the two domains [14]. In many scenarios, however, the domains have different granularities and therefore a one-one mapping proves too strong an assumption. For example, if one considers semantic, visual and audio domains, the granularity in each domain is quite different. A cluster in the semantic domain might correspond to multiple clusters in both visual (due to viewpoint, appearance differences etc.) and audio domains (due to difference in pronunciations). We break the one-one mapping restriction and propose a generalized co-clustering algorithm.

3.1. Formulation

Let's assume that this one-to-many mapping exists from $\mathcal{D}_1$ to $\mathcal{D}_2$. The input to the algorithm is N data points with each point being represented as $X_i =< x_i^1, x_i^2 >$ (x_i^d is the feature representation of the i^{th} datapoint in domain $\mathcal{D}_d$). The output of our clustering algorithm is a set of clusters in each domain (defined in terms of an assignment of data points to each cluster) and a one-to-many mapping between the clusters in two domains.

We represent the clusters and the one-to-many mapping as a bipartite graph $G = (V^1, V^2, E)$, where V^1 and V^2 are the set of clusters in domain $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively. E represents the set of edges between clusters in V^1 and V^2; therefore, $E_{k,l} = 1$ indicates cluster k in $\mathcal{D}_1$ corresponds to cluster l in $\mathcal{D}_2$. We enforce the one-to-many constraint by ensuring that for each l, $\sum_k E_{k,l} = 1$. Each cluster node (k, l) in domain $(\mathcal{D}_1, \mathcal{D}_2)$ is associated with model parameters (θ_k^1, θ_l^2). For each data point $X_i =< x_i^1, x_i^2 >$, its cluster membership is represented by a corresponding pair of cluster labels $Y_i =< y_i^1, y_i^2 >$ (y_i^d represents the membership of i^{th} datapoint in domain $\mathcal{D}_d$). Therefore, given X, our goal is to infer G^*, Θ^*, Y^* such that it maximizes the scoring function $\mathcal{S}(G, \Theta, Y, X)$, which is defined as:

$$\sum_{d \in \{\mathcal{D}_1, \mathcal{D}_2\}} \left(\overbrace{\sum_i \Psi^d(x_i^d, y_i^d, \Theta^d)}^{\text{data likelihood}} + \overbrace{\sum_{i,j} \Phi^d(x_i^d, x_j^d, y_i^d, y_j^d)}^{\text{smoothness}} \right) + \overbrace{\sum_{i,j} \Phi^{12}(x_i^2, x_j^2, y_i^1, y_j^1)}^{\text{cross-domain}}$$

The first term in the scoring function corresponds to the data likelihood term in the two domains. This term prefers coherent clusters within each domain independently, which are explained using the model parameters Θ^d. The second term in the scoring function is the smoothness term. This term attempts to ensure that if two data points x_i and x_j are similar in domain $\mathcal{D}_d$ they get assigned to the same cluster in that domain. Note that if the structure (*e.g.* number of clusters) is fixed, then the smoothing term is redundant, but here it is essential to 1) regularize the likelihood term and avoid trivial solution (one cluster for each data point - high likelihood and good mapping), and 2) provide both intra- and inter cluster distance metrics to make proper structure movements. The third and final term in the scoring function is the cross-domain term which indicates that if two instances are similar in domain $\mathcal{D}_2$, then these data points should be assigned to same cluster in domain $\mathcal{D}_1$. Note this term is the asymmetric due to asymmetric nature of one-to-many relationship between the two domains. In Section 4, we define the specific Ψ^d, Φ^d and Φ^{12} that instantiate this approach in our text-vision application.

3.2. Optimization using Iterative Approach

Optimizing the above equation is in general an NP-hard problem. We therefore use an iterative optimization approach inspired by structure-EM for maximizing the above scoring function.

Given a fixed graph structure (fixed number of clusters and mapping), we iterate over estimating Θ and Y using

hard-EM style iterations [22] to maximize the scoring function $\mathcal{S}$. That is, given Θ, we perform inference to assign data points to nodes in each domain by estimating the membership variable Y. We enforce the one-to-many constraint using a cautious approach [46], dropping datapoints which are not congruent with the mapping in structure G. Specifically, we treat the low-scoring data points as noise and discard them. Once we estimate membership Y, we use this new membership to estimate the new parameters Θ.

After estimating $\mathcal{S}(\cdot)$ for a given G_t, we then take a structure step. Here, we create proposals for changes in structure (splitting a node into two or merging two nodes into one). We greedily select the best proposal G using an approximation function. Using the newly proposed G_{t+1}, we re-estimate the scoring function $\mathcal{S}$ using EM over Θ and Y. If the new estimated score is higher than the score at previous iteration, we accept the structure step and continue. If the estimated score is lower, we reject the structure step and switch back to $G_{t+t} = G_t$. For initialization, G_0, we use a single node in domain $\mathcal{D}_1$ and K nodes in domain $\mathcal{D}_2$ (We use a high-value of K to ensure that the clusters are consistent). Therefore, the structure steps are split proposals in domain $\mathcal{D}_1$ and merge steps in $\mathcal{D}_2$. The pseudo code is shown in Algorithm 1.

Algorithm 1: Iterative Approach to Maximize Scoring Function

Input: Datapoints: X where $X_i =< x_i^1, x_i^2 >$, x_i^d = feature in Domain d
Output: Clustering in Two Domains: $Y_i =< y_i^1, y_i^2 >$, (G,Θ)
// Initialization: 1 Cluster in $\mathcal{D}_1$, K Clusters in $\mathcal{D}_2$
$G_0, \Theta_0, Y_0 \leftarrow$ InitializeGraph(X,K)
$\mathcal{S}_0 \leftarrow \mathcal{S}(G_0, \Theta_0, Y_0)$; // Estimate Initial Score
while *Rejects<R* **do**
  // Propose New Structure G_{t+1} based on Split/Merge Proposal
  $G_{t+1} \leftarrow$ GenerateNewProposal(G_t, Y)
  // Use EM to estimate Θ_t, Y_t
  while Y_t *not converged* **do**
    // Estimate $\Theta_{t'+1}$ based on G_{t+1} and $Y_{t'}$
    $\Theta_{t'+1} \leftarrow$ TrainNewClassifiers(G_{t+1},$Y_{t'}$)
    // Estimate $Y_{t'+1}$ based on G_{t+1} and $\Theta_{t'+1}$
    $Y_{t'+1} \leftarrow$ AssignPointstoClusters(G_{t+1},$\Theta_{t'+1}$)
  // Estimate new score $\mathcal{S}_{t+1}$
  $\mathcal{S}_{t+1} \leftarrow \mathcal{S}(G_{t+1}, \Theta_{t+1}, Y_{t+1})$
  if $\mathcal{S}_{t+1} < \mathcal{S}_t$ **then**
    // Reject if score decreases
    $G_{t+1} \leftarrow G_t, \mathcal{S}_{t+1} \leftarrow \mathcal{S}_t, Y_{t+1} \leftarrow Y_t, \Theta_{t+1} \leftarrow \Theta_t$
    Rejects $\leftarrow$ Rejects+1
  else
    // Accept the Proposal
return G_t, Θ_t

4. Discovering Semantic and Visual Senses

We now adapt our generalized co-clustering algorithm to the task of discovering multiple semantic and visual senses. The outline of our approach is shown in Figure 2. In this case, $\mathcal{D}_1$ is the text domain and $\mathcal{D}_2$ is the visual domain. Our input data points are obtained by querying Google Image Search for each NP and downloading the top-1000 webpages. We now describe our text and image features, the likelihood, smoothness and the cross-domain terms.

4.1. Text Domain

Given a NP and a webpage containing the NP, we extract x_i^1 as follows: first, we use the Stanford parser [13] to perform syntactic parsing of the sentences. For each mention of the NP in the webpage, we extract features which include dependency paths of length one and two steps away from the NP head. We also include bag-of-word (BOW) features from the webpage. In many cases, the associated text might contain topics irrelevant to the NP. To handle this, the BOW representation is constructed based on the sentences which mention the NP. Note that we also use the part-of-speech tags as well to form the BOW representation (therefore, amber_ADJ is treated differently from amber_NN). This leads to a very high-dimensional feature vector; to address this, we use an LDA topic model [6] (learned from 1M webpages) and project the extracted BOW features to the topics to obtain the final unit-norm feature vector, x_i^1.

Next, we discuss how each cluster is represented in the text domain and what are the associated parameters. We represent each text cluster with the mean feature vector of all the cluster members. Given this representation, we simply model the likelihood term as the histogram intersection $\chi(\cdot,\cdot)$ [2] of the mean and the input feature vector. We define the smoothness term as follows:

$$\Phi^1(x_i^1, x_j^1, y_i^1, y_j^1) = \sum_{i,j} \chi(x_i^1, x_j^1)\mathcal{I}(y_i^1 = y_j^1) \qquad (1)$$

where $\chi(x_i^1, x_j^1)$ is histogram intersection similarity and $\mathcal{I}(\cdot,\cdot)$ is the indicator function. This term rewards the highly similar data points if they have the same label. The reward is proportional to the dot-product $\Delta(\cdot,\cdot)$ between the two feature points.

4.2. Image Domain

To represent the visual data, we first extract image based features. However, in the case of images, modeling the likelihood is quite tricky since the object location inside the image is unknown. To overcome this problem we use the algorithm for the clustering proposed in [10]. Given the set of input images for a NP, this algorithm generates the set of bounding boxes which are the location of objects in those images. It also clusters the visual data into K clusters which we use as initialization for G_0. Once the object location is known we represent the object (x_i^2) using HOG features [12].

Given HOG based representation of object, we represent each cluster in terms of a linear-SVM weight vector(θ_k^2). This linear-SVM is trained by treating cluster members as

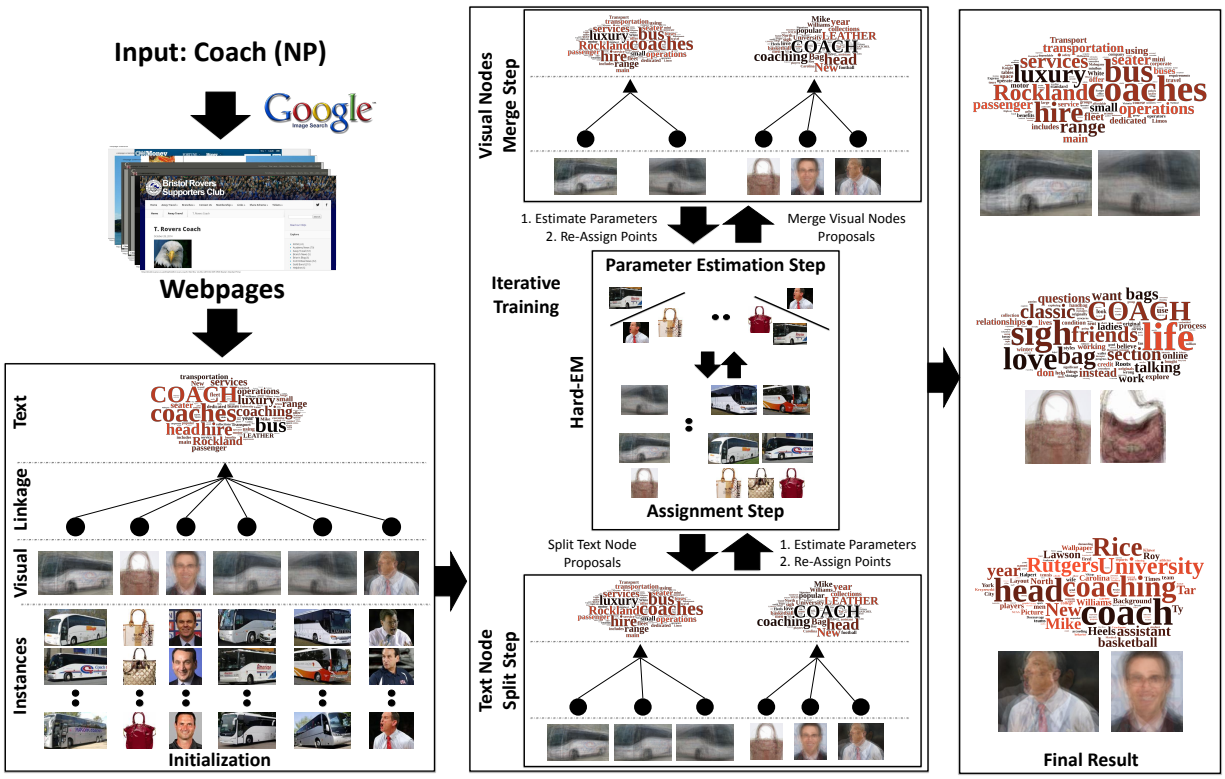

Figure 2: We discover semantic and visual senses using the structure-EM approach shown above. Given a NP (e.g., Coach), we first download images and webpages. We then initialize the algorithm with a single semantic sense and with visual senses initialized using the clustering algorithm described in [9]. Then, at each iteration, semantic senses are refined and visual senses are generalized using an iterative approach.

positive data points and bounding boxes from random images [3] as negative data points.

Therefore, the likelihood score of a point, x_i^2, coming from visual cluster k is defined as the ${\theta_k^2}^T x_i^2$. For modeling the smoothness term, we compute the similarity $(\Delta(x_i^2, x_j^2))$ between two feature vectors x_i^2 and x_j^2 using the dot product on whitened HOG feature [21, 29]. Given this similarity metric, the smoothness term is similar to the text term where the reward is proportional to the similarity between the two images:

$$\Phi^2(x_i^2, x_j^2, y_i^2, y_j^2) = \sum_{i,j} \Delta(x_i^2, x_j^2)\mathcal{I}(y_i^2 = y_j^2) \qquad (2)$$

4.3. Cross-Domain Term

The final term we need to model is the cross domain term. This term ensures that data points which are similar in the visual domain are assigned to same cluster in the text domain. The term is defined as follows:

$$\Phi^{12}(x_i^2, x_j^2, y_i^1, y_j^1) = \sum_{i,j} \Delta(x_i^2, x_j^2)\mathcal{I}(y_i^1 = y_j^1) \qquad (3)$$

[3] These random images are scene images obtained from Google Image Search.

where $\Delta(\cdot,\cdot)$ is the similarity defined as the dot-product of whitened-HOG features of the datapoint i and j.

4.4. Optimization

Given these terms in the scoring function, we now optimize using the structure-EM approach described in Section 3.2. In the structure step, we alternate between text split proposals and visual merge proposals. We now describe how we generate the split proposals for nodes in the text domain and how we generate merge proposals in the visual domain.

Split Proposals: Given the set of text nodes V^1, at every alternate iteration we generate proposals by splitting every text node into two nodes. These splits are generated based on the one-to-many mapping between the text node and the visual nodes. Intuitively, we try to create splits by generating new possible semantic senses based on one of the visual senses. Formally, let us suppose, a text node V_l^1 is linked to the visual nodes $V_{l_0}^2 ... V_{l_m}^2$. We generate split proposals by selecting pair of visual nodes and training a text classifiers such that instances belonging to one visual node is treated as positive and the instances belonging to other visual node is treated as negative data. This allows us to create $\binom{m}{2}$ split proposals and we select the best proposal based on the

Airbus A380	732	AK 47	654	Apple	635	Bass	804	Bean	902	Black Swan	507
Chicken	845	Coach	754	Columbia	779	Corolla	667	Daybed	763	F18	611
Falcon	831	Football	559	Jordan	662	Los Angeles Lakers	651	M 16	664	Mouse	838
Mitsubishi Lancer	663	Motorbike	850	Note	766	Robin	830	Sofa	585	Sparrow	859
Subway	803	Tuna	814	Wagon	824	Whitefish	817	Wolf	779	Yellow Tail	682

Table 1: CMU Polysemy-30 Dataset for Discovery of Visual and Semantic Senses

regularized empirical risk of the trained classifier.

Merge Proposals: Given the set of visual nodes that belong to same semantic node, we create proposals for merging two visual nodes based on the likelihood scores. If members of visual cluster l receive high likelihood scores from the classifier associated with cluster l' and the two nodes are assigned to the same semantic node, then we create a proposal to merge the nodes l and l'.

4.5. Implementation Details

In order to handle noise in the Google search results, we create an extra cluster/node on the vision side. This allows us to handle outliers in the clustering process. Unlike other clustering approaches which tend to partition the whole feature space, our approach only focuses on extracting semantic and visual senses from the subset of data which is considered high confidence in either domain. The low confidence data points are assigned to the noise cluster and are not considered part of the scoring function.

Some of the data points also have missing data from one domain. For example, we might have images but no text associated with it. Instead of ignoring such data points, we prefer to assign them based on image features alone. This is necessary as in many cases our visual clusters are data starved and using these extra data points help us learn better visual classifiers.

5. Experimental Results

We now show the effectiveness of our co-clustering algorithm using extensive experimental analysis. We perform four different experiments and implement several baselines. First, we introduce a new challenging dataset for this task (CMU Polysemy-30). This dataset consists of 30 NPs and $\sim$1000 webpages for each NP are downloaded from Google Image Search. We do a clean up step and after accounting for broken links, webpages not reachable, we end up with $\sim$750 images per NPs. Table 1 shows the list of NPs and the number of data points for each NP. The dataset is publicly available for download[4].

In order to evaluate the performance of the sense extraction we manually listed all the possible semantic senses for each NPs. We then manually labeled $\sim$5600 instances with one of the listed semantic senses. We use accuracy (AC) as one way to measure the clustering performance. Before evaluation, we first obtain a mapping between the ground truth clusters and clusters obtained using Kuhn-Munkres algorithm [33]. We also use the standard normalized mutual information (NMI) [30] metric to evaluate our clustering. Note that higher mutual information implies better clustering performance.

To overcome the human labeling bottleneck, we also perform another experiment which creates pseudo-words to evaluate sense disambiguation [40] (Sec 5.3). Next, we perform a retrieval experiment on the MIT ISD dataset [37] which has 5 polysemous concepts (Sec 5.4). Finally, we perform a large-scale experiment where we run our algorithm on $\sim$2000 NPs (Sec 5.5). The list of these concepts is obtained using the NEIL knowledge base [9].

5.1. Baselines

We compare the performance of our approach against multiple baselines which use text and image features. For all the baselines, we use two versions: pre-defined number of clusters (Fixed) and using Eigen-Gap criterion [43] (Eigen) to automatically compute the number of clusters. Our baselines are:

Spectral Clustering on TF-IDF Features (TF-IDF): Our first baseline uses text based features only and constructs a feature representation of a webpage using TF-IDF [30] features over the context words. These features are used in conjunction with cosine-distance to create an affinity graph. Finally, we perform spectral clustering over this affinity graph.

Spectral Clustering on BOW (BOW): Our second baseline uses the BOW text features to represent each webpage. Similar to our text feature construction, we build BOW over the word and their Part-of-Speech tags. We use histogram intersection as the similarity metric to create the affinity graph.

Image BOW over SIFT and Color Histogram (I-BOW): Our third baseline uses image-based features. Specifically, we use SIFT and Color Histogram features. To create a BOW representation, we perform vector quantization with 1000 words for SIFT and 400 words for Color histogram. Given this visual BOW representation, we create an affinity graph using histogram intersection similarity metric.

Spectral Clustering on Topic Model based Representation: Our fourth baseline uses text-based features used in our algorithm to represent a webpage and histogram intersection is used as the similarity metric to create the affinity graph. Note that this baseline is an unsupervised variant of the approach taken by [37], which uses extra information (WordNet) to determine the underlying senses.

Clustering in Joint Space: As our final set of baselines, we implemented the algorithms of [24, 45] which perform

[4] http://www.neil-kb.com/poly.html

Figure 3: Examples of semantic and visual sense discovery of our algorithm. For example, our algorithm automatically discovers two semantic senses for *Bass*: fish and musical instrument. For *Bean*, it discovers semantic senses of jelly beans, bean food and Mr. Bean. Last two examples in the third row show two failure cases.

clustering in the joint space of image and text features. [24] uses BOW for both images and text, while [45] uses topic model based representations.

The code for [28] was not available and requires click-through data to train which is proprietary. But we did compare qualitatively against Google Image Search query expansions and use human annotators to quantitatively compare performance against them.

5.2. CMU Polysemy-30 Dataset

Qualitative Results: We first show qualitative performance of our algorithm. Figure 3 shows the qualitative result of our clustering algorithm on some NPs from CMU Polysemy-30: Yellow Fish, Wagon, Subway, Bass, Bean, M16 and Tuna. Notice how our algorithm discovers the two semantic meanings of Subway: the metro and the sandwich brand. Specifically, note the text features in the word clouds. For the subway (metro) sense, the main distinguishing features include: New, York, Station, City, tracks, line, transit. For the subway (sandwich) sense, the main distinguishing text features are: sandwich, Surfers, restaurants, art, food, sandwiches. Also notice the associated visual senses. For example, for subway(metro), the visual senses are the station and the metro train itself. Similarly, for subway(sandwich chain), visual senses include the subway logo and the sandwich itself.

Another interesting example is the M16 shown on middle right. Our algorithm perfectly discovers the right semantic senses: nebula, music album and the rifle. Notice the associated text features for each semantic sense. For the nebula sense, the most important words are: Eagle (also known as eagle nebula), Nebula, Telescope, cluster etc. For the music album sense, the important words are preview, buy and iTunes. Figure 3 also shows a couple of cases (last two, Mouse and Chicken) where our algorithm fails to discover all the associated senses.

We also qualitatively compared to Google Image Search query expansions. For example, for Whitefish NP, Google misses the lighthouse location sense and only captures the resort sense. (see Figure 4).

Quantitative Results: We now discuss the quantitative results and compare the performance of our algorithm against several baselines. Table 2 shows the comparative performance for semantic sense discovery. As the quantitative results indicate, our approach outperforms all the baselines by a significant margin. Note that our approach automatically discovers the true number of semantic senses and outperforms the baselines even when the true number of senses

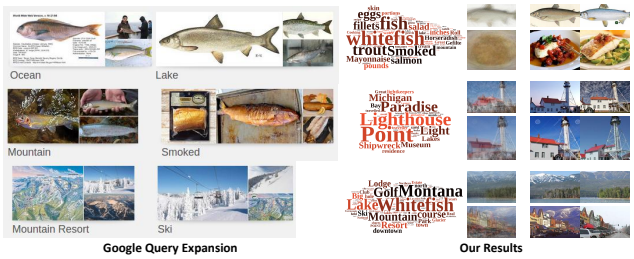

Figure 4: Comparison with Google Query Expansion for NP Whitefish.

are provided to the baselines (fix the number of clusters before spectral clustering) due to noise.

For the evaluation of visual senses, since it is hard to obtain ground truth labels, we computed the purity (how many instances belong to the same semantic sense) for all the visual senses obtained by [10] and our approach. As expected, our algorithm improves purity over iterations, giving 3% boost in clustering performance. The boost is significantly higher when two different senses have similar visual appearances (Apple logo looks very similar to apple fruit).

5.3. Pseudo-word Based Evaluation

One bottleneck for the evaluation of sense disambiguation algorithms is the requirement of human labeling. A neat way out of this is the commonly used approach of pseudo-words [40]. More specifically, pseudo polysemous NPs can be created by combining together multiple single-sense NPs. For example, we can combine the webpages downloaded for Accord and Boeing (non-polysemous words) and treat them as retrieval for a pseudo polysemous word "accord-boeing". Now, by construction, we have the labels for semantic sense since the webpages for accord are sense 1 and webpages for boeing are sense 2.

For our experiments, we combine four NPs: accord, boeing, tire, cricket ball. For these four NPs, there are $2^4-1=15$ possible combinations with these pseudo words having somewhere between 1-4 semantic senses per word. Table 3 show the comparative performance of our approach against all the baselines. Again, in this case, our approach outperforms all the baselines significantly.

5.4. MIT ISD Dataset

Next, we apply our unsupervised approach for the task of re-ranking image search engine outputs. We use MIT ISD Dataset [37], which was collected automatically from the Yahoo Image Search. It consists of 5 NPs: Bass, Face, Mouse, Speaker and Watch. For image retrieval, one target sense is picked for each NP: the fish, the human face, the pointing device, the audio device, and the timepiece. We evaluated the retrieval performance with Area Under the (ROC) Curve. Compared to Yahoo Image Search, our unsupervised approach is able to obtain 20% performance gain on average.

	AC		NMI	
	Fixed	Eigen	Fixed	Eigen
TF-IDF	79.31	76.66	11.04	19.65
BOW	77.94	70.79	15.97	12.64
I-BOW	77.86	70.17	6.84	8.04
[37]	80.52	73.21	18.69	17.75
[24]	78.39	79.89	8.96	27.00
[45]	80.62	73.13	19.03	18.26
Our Approach	**86.70**		**34.11**	

Table 2: Quantitative Evaluation on the CMU Polysemy-30 Dataset.

	AC		NMI	
	Fixed	Eigen	Fixed	Eigen
TF-IDF	53.69	49.12	10.32	6.07
BOW	58.98	68.95	15.97	31.64
I-BOW	61.66	57.47	22.03	16.03
[37]	70.04	77.20	46.55	44.52
[24]	54.83	55.52	12.80	18.72
[45]	69.08	72.83	32.80	43.33
Our Approach	**83.89**		**53.81**	

Table 3: Quantitative Evaluation on the Pseudo-NP Dataset.

5.5. Large-Scale Sense Discovery Experiments

Finally, our sense discovery approach is linear in the number of categories and therefore scales reasonably well. As an extension of the CMU Polysemy-30 dataset, we also evaluate our algorithm on 1.8 million images and websites from Google Image search, using a list of ∼2000 NPs from NEIL [9] as queries. We randomly evaluated the sense discovery for 100 keywords and found our algorithm can recover 82% of senses from Wikipedia.

6. Conclusion

This paper presents an approach for co-clustering in two domains. Most co-clustering algorithms make the simplifying assumption that there exists a one-to-one mapping between two domains. In our proposed algorithm, we relax this assumption and allow one-to-many mapping which is useful when the granularity of clustering in two domains is different. We apply our co-clustering algorithm for the task of discovering multiple semantic and visual senses of a given NP. We show that our algorithm is effective in not only figuring out the right senses and clustering the data but it also generates the right mapping between semantic and visual senses. We compare our performance against several baselines and show significant gains over these baselines.

Acknowledgements: This research is supported by ONR MURI N000141010934, Yahoo-CMU InMind program and a gift from Google. AG and XC were partially supported by Bosch Young Faculty Fellowship and Yahoo Fellowship respectively. The authors would also like to thank Yahoo! for the donation of a computing cluster to the project NEIL.

References

[1] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction for the web. In *IJCAI*, 2007. 1

[2] A. Barla, F. Odone, and A. Verri. Histogram intersection kernel for image classification. In *ICIP*, volume 3, pages III–513. IEEE, 2003. 4

[3] K. Barnard and M. Johnson. Word sense disambiguation with pictures. In *AI*, 2005. 3

[4] T. Berg and D. Forsyth. Animals on the web. In *CVPR*, 2006. 3

[5] T. L. Berg and A. C. Berg. Finding iconic images. In *CVPR Workshops*. IEEE, 2009. 2

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 2003. 4

[7] S. Brody and M. Lapata. Bayesian word sense induction. In *EACL*, 2009. 2

[8] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, 2010. 1

[9] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *ICCV*, 2013. 1, 2, 5, 6, 8

[10] X. Chen, A. Shrivastava, and A. Gupta. Enriching visual knowledge bases via object discovery and segmentation. In *CVPR*, 2014. 4, 8

[11] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP*, 2007. 1

[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 4

[13] M.-C. De Marneffe, B. MacCartney, C. D. Manning, et al. Generating typed dependency parses from phrase structure parses. In *LREC*, pages 449–454, 2006. 4

[14] V. R. De Sa. *Unsupervised classification learning from cross-modal environmental structure*. PhD thesis, University of Rochester, 1994. 3

[15] V. R. de Sa. Spectral clustering with two views. In *Workshop on Learning with Multiple Views*, 2005. 3

[16] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *SIGKDD*, 2001. 3

[17] S. K. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, pages 3270–3277. IEEE, 2014. 1, 2

[18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 32(9):1627–1645, 2010. 2

[19] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *ECCV*, 2004. 2

[20] E. Golge and P. Duygulu. Conceptmap: Mining noisy web data for concept learning. In *ECCV*, pages 439–455. Springer, 2014. 2

[21] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*. 2012. 5

[22] M. Kearns, Y. Mansour, and A. Y. Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Learning in graphical models*. 1998. 4

[23] J. Krishnamurthy and T. M. Mitchell. Which noun phrases denote which concepts? In *ACL*, 2011. 2

[24] N. Leoff, C. Alm, and D. Forysth. Discriminating image senses by clustering with multi-modal features. In *ACL*, 2006. 3, 6, 7, 8

[25] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, 1986. 1

[26] L. Li, G. Wang, and L. Fei-Fei. A visual category filter for google images. In *CVPR*, 2006. 2

[27] Q. Li, J. Wu, and Z. Tu. Harvesting mid-level visual concepts from large-scale internet images. In *CVPR*, pages 851–858. IEEE, 2013. 2

[28] A. Lucci and J. Weston. Joint image and word sense discrimination for image retrieval. In *ECCV*, 2012. 2, 3, 7

[29] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 5

[30] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. 6

[31] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 1

[32] P. Pantel and D. Lin. Discovering word senses from text. In *SIGKDD*, 2002. 2

[33] D. Plummer and L. Lovász. *Matching Theory*. North-Holland Mathematics Studies. Elsevier Science, 1986. 6

[34] S. Qiu, X. Wang, and X. Tang. Visual semantic complex network for web images. In *ICCV*, 2013. 2

[35] R. Raguram and S. Lazebnik. Computing iconic summaries of general visual concepts. In *CVPR Workshops*. IEEE, 2008. 2

[36] A. Ritter, L. Zettlemoyer, Mausam, and O. Etzioni. Modeling missing data in distant supervision for information extraction. *TACL*, 2013. 1, 2

[37] K. Saenko and T. Darrell. Unsupervised learning of visual sense models for polysemous words. In *NIPS*, 2008. 3, 6, 8

[38] K. Saenko and T. Darrell. Filtering abstract senses from image search results. In *NIPS*, 2009. 3

[39] F. Schroff, A. Criminsi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, 2007. 2, 3

[40] H. Schütze. Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123, 1998. 6, 8

[41] R. Snow, S. Prakash, D. Jurafsky, and A. Y. Ng. Learning to merge word senses. In *EMNLP*, 2007. 1

[42] D. Tsai, Y. Jing, Y. Liu, H. A. Rowley, S. Ioffe, and J. M. Rehg. Large-scale image annotation using visual synset. In *ICCV*, pages 611–618. IEEE, 2011. 2

[43] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 6

[44] K. Wan, A. Tan, J. Lim, T. Chia, and S. Roy. A latent model for visual disambiguation of keyword-based image search. In *BMVC*, 2009. 3

[45] K.-W. Wan, A.-H. Tan, J.-H. Lim, L.-T. Chia, and S. Roy. A latent model for visual disambiguation of keyword-based image search. In *BMVC*, 2009. 2, 6, 7, 8

[46] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*, 1995. 2, 4