Language Grounding in Massive Online Data

A Dissertation Presented

By

Jianfu Chen

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

December 2015

Stony Brook University

The Graduate School

Jianfu Chen

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

David S. Warren, Advisor
Professor, Computer Science Department

Paul Fodor, Chairman of Defense
Research Assistant Professor, Computer Science Department

I.V. Ramakrishnan
Professor, Computer Science Department

Yejin Choi
Research Assistant Professor, Computer Science Department

Hannaneh Hajishirzi, External Committee Member
Research Assistant Professor, Electrical Engineering, University of Washington

This dissertation is accepted by the Graduate School.

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Language Grounding in Massive Online Data

by

Jianfu Chen

Doctor of Philosophy

in

Computer Science

Stony Brook University

2015

Truly understanding natural language requires grounding language to perceptions and actions in the physical and social world. This goes beyond studying the textual modality alone. Today's web not only has sheer volume of data, but also increasingly multi-modal data, intertwining text with videos, images, audios, and ontologies that are perceptions or abstractions of people's everyday life. Hence the web provides rich and ever growing resources for studying grounded language. This thesis presents a series of investigations of language woven into various types of online data, ranging from ontology and images to time series. We contribute data distillation approaches and large-scale datasets connecting language to vision, a collection of models and algorithms, and multiple novel applications in hierarchical product classification, image description, and photo album summarization.

Contents

Acknowledgements

I deeply appreciate my advisor, Professor David Warren. I regard him as my model of a true scholar. Being open-minded, he encouraged me to explore my own interests. For whatever I work on, he shows amazing curiosity and appreciation. His attitude motivates me to work with greater passion. Discussions with him are always pleasant and thought-provoking. He pushes me to explain ideas clearly, and keep reflecting what do we learn. From principles to practice, his training makes my experience of writing and presentation a real fun blended with real effort. Very carefully and patiently, he proofread my papers and slides word by word, sentence by sentence, and slide and slide!

Many thanks to several professors who also helped me tremendously. Professor Yejin Choi gave me a strong hand in all aspects of research. Her hard-working is infectious. She taught me to really understand what is going on by visualizing and examining many examples. She is awe-inspiring in prioritizing tasks and getting things done. Professor I.V. Ramakrishnan is always encouraging and likes my presentations. Professor C.R. Ramakrishnan supported me and we had many helpful discussions.

I would like to thank my fellow PhD student, Andrey Gorlin, who devoted much time to improve my presentation skill. We hang out quite a lot in our leisure time.

The last but not the least, I would like to thank all students, faculty, and staff at Computer Science Department. They have been maintaining a nice, helpful, and inspiring environment.

Chapter 1

Introduction

Researchers in various areas have discovered that understanding natural language requires grounding semantics to perceptions and actions in the physical world (Wittgenstein, 2010; Pecher et al., 2011). This goes beyond looking at the textual modality *alone*. For instance, it is impossible to learn the meaning of all words from only dictionary definitions. Because the definition of each word is based on other words recursively, leading to cycles or infinite regress, which is known as the *symbol grounding problem* (Harnad, 1990).

Today's web not only has sheer volume of data, but also increasingly multi-modal data, intertwining text with videos, images, audios, and ontologies that are perceptions or abstractions of people's everyday life. Hence the web provides rich and ever growing resources for studying grounded language.

A thrust of recent works studies language grounded in *conceptual abstractions*, including logical forms (Zettlemoyer and Collins, 2005; Artzi and Zettlemoyer, 2011; Liang et al., 2013), diagrams (Seo et al., 2014; Seo et al., 2015), knowledge bases (Bordes et al., 2011; Bordes et al., 2012), and databases (Riedel et al., 2013; Poon, 2013); and grounded in *perceptions and actions*, e.g., images (Kuznetsova et al., 2012; Le et al., 2013; Silberer and Lapata, 2014), videos (Yu and Siskind, 2013; Krishnamoorthy et al., 2013; Venugopalan et al., 2015), sportscasts (Chen et al., 2010; Bordes et al., 2010; Hajishirzi et al., 2011; Hajishirzi et al., 2012; Koncel-Kedziorski et al., 2014), robot instructions (Matuszek et al., 2014), and navigation instructions (Kim and Mooney, 2012).

This thesis presents a series of endeavours on investigating language woven into various types of online data, ranging from ontology and images to time series.

Our first study relates text to a **conceptual abstraction** (§2). In an online shopping platform, a taxonomy helps customers to explore and find products. We study classifying a textual product description into a given taxonomic ontology. Instead of optimizing 0-1 error rate as standard approaches, we design a classifier based on its *use* in the e-commerce world, that is, a vendor organizes a collection of products with a business goal to maximize revenue.

Our second study straddles text and **vision** (§3). Casual online activities involve images in conjunction with text. Researchers have explored this multi-modal web data to integrate language and vision. The main challenge to tapping into the web data is noise. Although readily available in very large quantities, naturally-existing web images and their captions have varying degrees of semantic correspondence. Everyday captions contain extraneous information (Kuznetsova et al., 2013b; Hodosh et al., 2013) that is not directly relevant to what the image shows. We propose a new approach to harvesting an image-caption dataset that makes better use of the existing web content and the future content exploding with billions of online activities every day. We demonstrate the potential utility of the new dataset in multiple ways.

Our third study aligns text with both **vision** and **time series** (§4). More often than taking random photos individually, people take a sequence of photos when participating a certain scenario, say wedding, camping, and Independence day. This results in a large number of online photo albums with time stamps for each photo. We propose to tap into the context of a photo stream to better understand both photos in sequence and their accompanying captions. The key idea is to ground image captions to prototypical events in a common scenario. For example, from a sequence of photos paired with captions regarding a wedding scenario, we might identify certain typical events in wedding that happen over time, for example, *vows*, *ring exchange*, *reception*, and *dancing*.

We now turn to the above three studies, respectively.

Chapter 2

Cost-Sensitive Hierarchical Product Classification

2.1 Overview

Our first study relates text to a taxonomic ontology, which is a common way to organize information.

E-commerce enables customers to buy products any time and anywhere. E-commerce has expanded rapidly over the last decade, and is predicted to continue its fast growth with the rise of smartphones and tablets.

In an online shopping platform, it is vital to enable customers find desired products quickly. To this end, the two most popular mechanisms are taxonomy organization and keyword search. A taxonomy organizes products by categories grouped hierarchically in a tree structure, from general classes to more specific classes. Two examples are the catalogs of Amazon.com and eBay.com. Such taxonomies complement keyword search. While keyword search is good for quickly finding a very specific product in a customer's mind, a taxonomy organization also has its merits: (1) *facilitates exploring similar products.* Categories are like departments in a supermarket that allow a customer to navigate and roam around a vast number of aisles. Are you looking for gift ideas for a kid? Just browse the sub-categories under the general "Toys&Games" category in Amazon.com. You will find many possibilities in a systematic way, e.g., dolls, puzzles, and building toys. In fact,

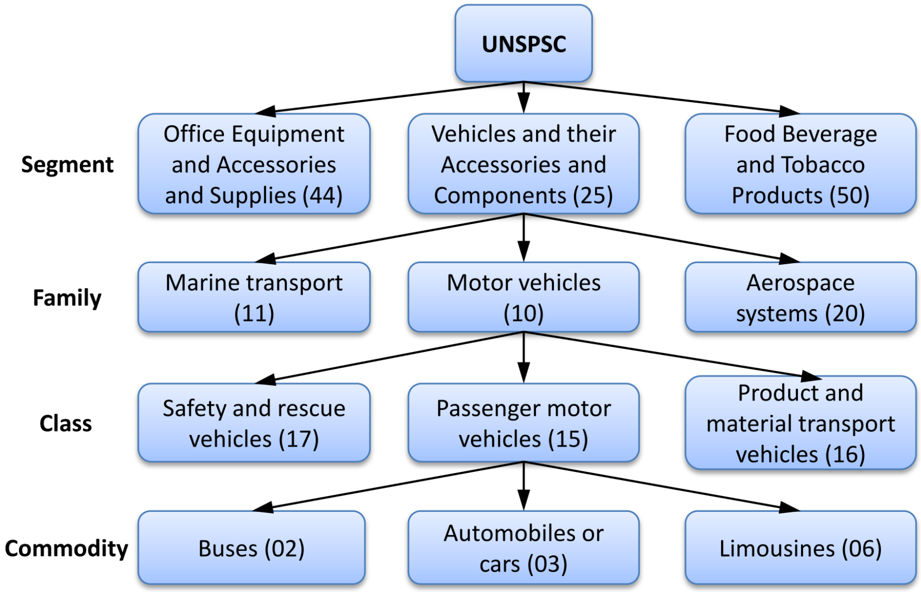

UNSPSC
Segment
Office Equipment and Accessories and Supplies (44)
Vehicles and their Accessories and Components (25)
Food Beverage and Tobacco Products (50)
Family
Marine transport (11)
Motor vehicles (10)
Aerospace systems (20)
Class
Safety and rescue vehicles (17)
Passenger motor vehicles (15)
Product and material transport vehicles (16)
Commodity
Buses (02)
Automobiles or cars (03)
Limousines (06)

Figure 1: Part of the UNSPSC taxonomy

even for a keyword query, many online shopping websites allow a customer to browse the search results by categories organized in a taxonomy. This helps a customer to filter out the really relevant categories. (2) *helps product recommendation.* Intuitively, given the products previously browsed or bought by a customer, we can use a taxonomy to recommend similar products in similar categories. Formally, (Ziegler et al., 2004) studies exploiting large taxonomies for personalized product recommendation.

Given the merits of a taxonomic organization of products, we explore automatic classification of textual product descriptions into a given taxonomy. Assume we are given a taxonomy that is a tree structure, where each product belongs to a single leaf class, and therefore also belongs to its more general ancestor classes. We want to classify a textual description of a product into one of the leaf nodes of a taxonomy. Figure 1 shows part of a product taxonomy called UNSPSC [1] .

In particular, we investigate two essential problems, *performance evaluation* and *learning*, in a synergistic way. Unless we know what is the appropriate performance evaluation metric for a task, we are not going to learn a classifier that has maximum utility for the task. We study them under a unified view of empirical risk (Vapnik, 1999), the average loss or misclassification cost. A performance evaluation metric defines a type of

[1] see www.unspsc.org for details.

misclassification cost. Learning optimizes the average misclassification cost.

Performance evaluation is a seemingly trivial problem, and hence is often neglected in real world applications. However, we argue that we should choose an appropriate performance evaluation metric according to the characteristics of the task, rather than just blindly choosing a common evaluation metric like error rate. We examine the special characteristics of the task of hierarchical product classification, where a vendor classifies products with a business goal of maximizing revenue (§2.2.1). We shed insight into ***how and why*** common evaluation metrics can be misleading (§2.2.3). The analysis covers metrics including error rate, mean error rate, average hierarchical loss, and average F1-score, which is applicable when considering evaluating other real world tasks. Then we design a new evaluation metric that fixes the problems of common evaluation metrics and tailors this task to reflect a vendor's business goal of maximizing revenue. The proposed metric is essentially the average revenue loss, which depends on *both* the potential **revenue** of individual products *and* the hierarchical **distance** of the true class and the predicted class in the taxonomy.

After choosing an appropriate performance evaluation metric for the task of hierarchical product classification, we explore learning a classifier that optimizes the proposed evaluation metric, average revenue loss, rather than error rate as commonly done by standard classifiers (§2.3). We use a generalization of multi-class SVM with margin re-scaling (Tsochantaridis et al., 2006; Crammer and Singer, 2002) to optimize *any loss functions*. It is a general approach to handle cost-sensitive learning. However, margin re-scaling is sensitive to the scaling of loss functions, especially when the loss function, revenue loss, can span a wide range. We propose an approach to normalize the loss function into a fixed range, appropriately calibrating the scaling of the loss functions. Our **loss normalization** approach is applicable to other classification and structured prediction tasks, whenever using structured SVM with margin re-scaling.

Finally, we perform experiments on a large dataset that has more than one million products in about one thousand leaf classes (§2.4). The results show that our approach outperforms standard multi-class SVM in terms of our proposed evaluation metric, significantly reducing the average revenue loss.

Our work is an application of cost-sensitive learning when different misclassification have different costs (Elkan, 2001; Domingos, 1999; Zhou and Liu, 2006; Zadrozny et al., 2003).

Very few works (Beygelzimer et al., 2008) study *both* example-dependent cost *and* class-dependent cost, especially in a practical scenario as we do. Though we study a particular task, this task represents an emerging class of applications that involve both a taxonomy and items with individual values in large scale information management.

2.2 Performance evaluation for hierarchical product classification

In this section, we first dissect the characteristics of the task of hierarchical product classification and envision the properties of a desirable performance evaluation metric. Then under a unified view of empirical risk, we analyze that common performance evaluation metrics, including *accuracy, mean accuracy, and average hierarchical loss,* fail to reflect a vendor's business goal sufficiently. Finally we propose a new evaluation metric that fixes the problems of common evaluation metrics.

2.2.1 Characteristics of the task

A close look at the uses of the classified products in the task motivates us to delve into the issue of classification performance evaluation.

2.2.1.1 Business goal

Consider the application scenario of product classification. A vendor or an online shopping platform classifies a set of products into the leaf classes of a predefined taxonomy. The vendor wants to classify the products as accurately as possible, so that potential customers can explore and find their needed products without obstacles. *We assume that if a product is classified into the correct class, the vendor will realize an expected annual revenue from that product. Otherwise, the vendor will lose some potential revenue, realizing only part of the expected annual revenue of that product, because customers have trouble finding and buying that product.* Hereafter all references to revenue are meant to be calculated within one year. Hence we drop the modifier "annual" for better readability.

A vendor's business goal is to **maximize revenue**. How should we evaluate a classifier's performance? To tailor a vendor's interest, a reasonable measure is the revenue loss caused by the classifier's misclassification.

The next question is: how much revenue will a vendor lose due to the misclassification of a product into a wrong class? Before we quantify the revenue loss in our proposed evaluation metric in 2.2.4, we first do some qualitative comparative analysis in the next section.

2.2.1.2 Taxonomy organization and customer behavior assumption

A qualitative analysis shows *the misclassification cost, or equivalently, revenue loss of a product into a sibling class should be smaller than that into a far-away class in the product taxonomy.* The analysis is based on the properties of the taxonomy organization and an assumption about the customer behavior in online shopping.

A product taxonomy groups product classes hierarchically in a tree structure, from general classes to more specific classes. Classes that share a common parent class are similar to each other. For example, the class "*mouse*" and the class "*computer keyboard*" share a common parent class "*desktop computer and accessories*"; the classes "*hard drive*", "*SSD*", and "*USB*" share a common parent class "*computer storage*". Those child classes are similar in the sense that they have similar functions, or dominant usages in the market.

We assume when a customer browses products by categories, the customer frequently looks at sibling classes. Sibling classes have similar or related products. A customer often compares similar products before buying them. One often wonders between the choice of "*hard drive*" products and "*SSD*" products. A customer also frequently purchases related products, say "*mouse*" and "*computer keyboard*" together.

Given the above properties of a product taxonomy and the assumption about customer behavior, the amount of revenue loss incurred by the misclassification of a product into different false classes should be different. In particular, we differentiate the misclassification cost into a sibling class and that into a far-away class in terms of **hierarchical distance** defined as the length of the shortest path between two nodes. We have assumed that it is highly likely that a customer looks at sibling classes when browsing products by categories. Suppose one product in the *keyboard* class, is misclassified into a sibling class, say *mouse*, even though a customer looking for *keyboard* products cannot find that product in its true class, the customer will likely check the sibling *mouse* class, hence can still find and buy that *keyboard* product. However, if that *keyboard* product is misclassified into a far-away class, say *car*, it is less likely that a customer will find and buy that *keyboard* product, because it is less likely that the customer will browse that far-away class *car* together with the true class *keyboard.* Hence the misclassification cost of a product into a sibling class should be smaller than that into a far-away class.

Given the characteristics of the task, now we turn to treat the problem of performance evaluation formally.

2.2.2 A unified view of classification performance evaluation

A unified view of a classification performance evaluation metric is *empirical risk*, which is the average loss incurred by classifying an example. This view helps us *both* to analyze performance evaluation metrics in 2.2.3 and 2.2.4, *and* to treat learning as empirical risk minimization in 2.3.2.

Suppose we are given a set of labeled examples: $\{(x, y)\}$, where each example $x \in R^N$ represents a product; x belongs to class $y \in Y$, where Y is the set of the leaf classes in the product taxonomy. Assume each labeled example (x, y) is drawn *i.i.d.* from an underlying joint distribution $P(x, y)$. A classifier is a function $f(x)$ that maps a given example x onto a class label $y' \in Y$.

A unified view of common classifier performance evaluation metrics is **empirical risk** (Vapnik, 1999) , which is the average loss incurred by classifying an example. Formally, let the loss function $L(x, y, y')$ represents the loss incurred by classifying an example x that belongs to class y into class y'; the empirical risk is the average loss of the classification over all examples,

$$R_{em} = \frac{1}{m} \sum_{(x,y,y') \in D} L(x, y, y')$$

where m is the total number of examples (m *represents the same meaning hereafter*). Typically, for correct classification, $L(x, y, y') = 0$; otherwise, $L(x, y, y') > 0$. So the lower the empirical risk is, the better the classifier performs.

In particular, we consider a class of loss functions that can be factorized as *a weighted classification error*,

$$L(x, y, y') = w(x) \cdot \triangle(y, y')$$

The **error function** $\triangle(y, y')$ quantifies the error of classifying an example that belongs to class y into class y'. The error function depends on only the true class y and the predicted class y'. Typically, for correct classification, $\triangle(y, y') = 0$; otherwise, $\triangle(y, y') > 0$. The **example weight** $w(x) \geq 0$ represents the importance of the example x's error function.

With such loss functions, we interpret empirical risk as **a weighted sum of classification errors**, ignoring the constant term $\frac{1}{m}$,

$$\frac{1}{m} \sum_{(x,y,y') \in D} w(x) \cdot \triangle(y, y') \tag{1}$$

The higher the example weight $w(x)$ is, the more importance the error function $\triangle(y, y')$ associated with example x has in the weighted sum.

2.2.3 The problems of common performance evaluation metrics

Based on the characteristics of the task of hierarchical product classification, we analyze that common classification performance evaluation metrics, including *error rate*, *mean error rate*, *average hierarchical loss*, and *average F1-score,* do not adequately reflect a vendor's business goal of maximizing revenue.

2.2.3.1 Error rate

The simplest loss function is a boolean function $L(x, y, y') = [y \neq y']$, where $[\cdot]$ is the Iverson bracket that returns the 0/1 boolean value of the inside condition. If the example is misclassified, then the loss is 1; otherwise the loss is 0. Hence it is called *0-1 loss.* The empirical risk as the average 0-1 loss over the set D becomes,

$$\frac{1}{m} \sum_{(x,y,y') \in D} [y \neq y'] \tag{2}$$

This is equivalent to the number of misclassified examples divides by the total number of examples. So the empirical risk with 0-1 loss is called **error rate**.

We interpret Eq.(2) as a special case of a weighted sum of classification errors in Eq.(1), where the example weight $w(x) = 1$, and error function $\triangle(y, y') = [y \neq y']$. We see that error rate has two problems that render it inadequate to reflect a vendor's business goal of maximizing revenue:

(1) *It gives an equal weight to each example.* A direct consequence is that error rate favors the performance in large classes that have many examples relative to other classes. This problem becomes severe when the class distribution is highly skewed, as is the case in our task of product classification. Error rate might neglect the performance on small classes that have relatively very few examples.

To see this formally, we group the error functions of examples by classes, and rewrite error rate as *a weighted sum of the error rates in all classes.* Let the number of classes be K, and the size of class y, that is the number of examples in class y, be S_y; we have,

$$\begin{aligned}
\frac{1}{m} \sum_{(x,y,y') \in D} [y \neq y'] &= \frac{1}{m} \sum_{y=1}^{K} \sum_{(x,y,y') \in D} [y \neq y'] \\
&= \sum_{y=1}^{K} \frac{S_y}{m} \cdot \left\{ \frac{\sum_{(x,y,y') \in D} [y \neq y']}{S_y} \right\} \\
&= \sum_{y=1}^{K} \frac{S_y}{m} \cdot Err_y
\end{aligned}$$

where Err_y denotes the error rate in class y.

We see that error rate is a weighted sum of the error rates in individual classes, where the weight of each class is proportional to its class size. With a highly skewed class distribution, error rate is dominated by the error rates in large classes, while ignores the performance in small classes.

In product classification with a vendor's business goal of maximizing revenue, we should give importance to a class based on the total revenue of the products in that class. The higher the total revenue of a class relative to other classes, the larger weight we should give to that class, emphasizing the importance to classify most of the products in that class correctly. It is inappropriate to give high importance to a class merely because the class has a relatively large number of products.

(2) The second problem of error rate is that *it treats the misclassification cost of an example of class y into all other classes y' equally.* As long as an example is misclassified, the error function $[y \neq y']$ will be 1. However, in hierarchical product classification, we want to discriminate the misclassification cost of a product into different false classes. According to the properties of a product taxonomy and the assumption about consumer behavior, we have shown that the revenue loss of a product into sibling classes should be smaller than that into far-away classes in the taxonomy in 2.2.1.2. If we cannot classify a product into the true class exactly, we want to classify it into a class as close as possible, to minimize the revenue loss.

In the next two sections, another two common performance evaluation metrics, *mean error rate* and *average hierarchical loss* address the above two problems, respectively.

2.2.3.2 Mean error rate

We have shown the first problem of error rate is that it gives an equal weight to each example, therefore favoring the performance in large classes, while neglecting the performance in small classes, when the class distribution is highly skewed. A straightforward remedy is to give equal weights to the error rates in individual classes. This is called *balanced error rate* (Chen and Lin, 2006). We call it mean error rate here, since it is the average error rate over individual classes. This view enables us to generalize it to use different weighting methods later in this section. Formally, we define **mean error rate** as,

$$\sum_{y=1}^{K} \frac{1}{K} \cdot Err_y \tag{3}$$

where K is the number of classes, Err_y is the error rate in class y, the number of misclassified examples in class y divided by the total number of examples in class y; formally $Err_y = \frac{\sum_{(x,y,y')\in D}[y \neq y']}{S_y}$. We rewrite mean error rate to the equivalent form as a weighted sum of classification errors,

$$\frac{1}{m} \sum_{(x,y,y')} \frac{m}{KS_y} \cdot [y \neq y']$$

We see that mean error rate gives an equal weight $\frac{m}{KS_y}$ to each example in class y, such that the sum of the weights in any class is uniformly $\frac{m}{K}$.

Generalized mean error rate. A slight generalization of mean error rate is to give non-uniform weights to the error rates in individual classes in Eq.(3). As long as those weights are non-negative and sum to 1. We define **generalized mean error rate** as,

$$\sum_{y=1}^{K} w_y \cdot Err_y \tag{4}$$

where class weight $w_y \geq 0$ for any class y, subject to $\sum_{y=1}^{K} w_y = 1$; and Err_y is the error rate in class y, defined the same as in Eq.(3). Similarly to mean error rate above, we rewrite generalized mean error rate as a weighted sum of 0-1 error functions,

$$\frac{1}{m} \sum_{(x,y,y')} \frac{mw_y}{S_y} \cdot [y \neq y'] \tag{5}$$

We see that generalized mean error rate gives an equal weight $\frac{mw_y}{S_y}$ to each example in class y, such that the total weight of the examples in class y is mw_y. It is easy to see that **both error rate and mean error rate are special cases of generalized mean error rate, with different class priors, or equivalently class weights.**

How should we choose the class weights w_y's? Consider the first problem of error rate. With a highly skewed class distribution, there are huge differences among the weights of the large classes and those of the small classes. To alleviate this problem, a heuristic way is to let the weight of a class y be proportional to $\log(S_y)$, or similarly $\sqrt{S_y}$, where S_y is the number of examples in class y. In product classification, a more reasonable way is to set the importance of a class proportional to the total revenue of that class.

We have seen how mean error rate and its generalized version try to fix the first problem of error rate of giving an equal weight to every example. Nevertheless, from Eq.(5), we see that both approaches still give equal weights to examples in the same class. However, in product classification, even within the same class, the revenue of different examples can span a wide range. To reflect a vendor's business goal of maximizing revenue, a more reasonable approach is to set the weight of each example proportional to its revenue. We will do this in our proposed metric in 2.2.4.

2.2.3.3 Average hierarchical loss

To fix the second problem of error rate of treating the misclassification cost into all false classes equally, a simple approach is to replace the 0-1 error function $\triangle(y, y') = [y \neq y']$ in the weighted sum of classification errors as Eq.(1) with an error function that differentiates the misclassification errors into different false classes. In particular, we use a loss matrix $L \in R^{K \times K}$ whose entry $L_{yy'}$ specifies the misclassification cost of an example from true class y to predicted class y'. Let the example weight $w(x) = 1$, and error function in Eq.(1) $\triangle(y, y') = L_{yy'}$. We define **average hierarchical loss** as,

$$\frac{1}{m} \sum_{(x,y,y')} L_{yy'}$$

In the case of hierarchical classification, we let $L_{yy'} = f(d(y, y'))$, a monotonically non-decreasing function of the hierarchical distance of y and y' in the taxonomy.

However, this evaluation metric still gives equal weight to each example's error function, leaving the first problem of error rate untouched. Combining both the ideas of example weighting and differentiating the misclassification cost of an example into different classes, we will propose in 2.2.4 a new evaluation metric that fixes both problems of error rate.

2.2.3.4 Average F1-score

For completeness, we analyze the problem of another common evaluation metric, **average F1-score** (Sun and Lim, 2001). It is a *not* a linear function of loss functions on individual examples, hence cannot be naturally cast as a weighted sum of classification errors as the above evaluation metrics. There are two types of average F1-score: *micro-averaged* F1-score and *macro-averaged* F1-score. In a multi-class problem, micro-averaged F1-score is equivalent to accuracy, that is 1 minus error rate, and thus has the same problems as error rate. Macro-averaged F1-score is the average of the F1-score over all classes. The problem of macro-averaged F1-score is that it gives an equal weight to the performance of each class. In product classification, it makes more sense to give importance weight to each class based on the total revenues of the products in that class.

2.2.4 Proposed performance evaluation metric - average revenue loss

Combining both the ideas of example weighting and differentiating the misclassification cost into different false classes, we propose a new evaluation metric. Intuitively, the proposed metric represents the average revenue loss incurred by the classification over the products, therefore directly reflecting a vendor's business goal of maximizing revenue, or equivalently, minimizing revenue loss.

The proposed metric basically quantifies the assumptions we make in 2.2.1. We assume that,

(1) If a product x is classified into the correct class, then its potential customers will be able to find it without hindrance, hence the vendor will *fully* realize a potential revenue $v(x)$ of product x.

(2) If a product x is misclassified into a wrong class, then its potential customers will have trouble to find and buy it, hence the vendor will lose some *percentage* of the potential revenue $v(x)$ of that product, depending on the hierarchical distance of the true class and the predicted class. The further apart the two classes are, the larger the percentage of the revenue the vendor will lose. We call such a percentage as a **loss ratio**. Formally, assume we are given the revenue loss ratio $L_{yy'}$ of classifying any product x of class y to

class y', such that $0 \leq L_{yy'} \leq 1$, and $L_{yy'}$ is a monotonically non-decreasing function of the hierarchical distance $d(y, y')$ between y and y'. In particular, $L_{yy} = 0$ for any class y. In a general sense, a loss ratio gives a partial credit to the classification. Going up to the lowest common ancestor of classes y and y', we have a class that is correct in a coarser grain sense.

Based on the above two assumptions, the *revenue loss* of classifying a product x that belongs to class y into class y' becomes $v(x) \cdot L_{yy'}$. Let the loss function be the revenue loss, $L(x, y, y') = v(x) \cdot L_{yy'}$, we propose a performance evaluation metric to represent the **average revenue loss** caused by the classification over the products in the considered set,

$$\frac{1}{m} \sum_{(x,y,y') \in D} v(x) \cdot L_{yy'}$$

Technically, like other common evaluation metrics discussed, this evaluation metric is a weighted sum of classification errors in Eq.(1). It gives an importance weight $v(x)$ to each product that is proportional to the product revenue. It discriminates the misclassification cost of a product into different false classes based on the hierarchical distance of the true class and the false class. Therefore, it solves both the two problems of error rate in 2.2.3.1. Moreover, it encompasses both error rate and average hierarchical loss as special cases, by setting $L_{yy'} = 1$ and $v(x) = 1$, respectively.

2.3 Cost-sensitive learning for hierarchical product classification

After choosing the *average revenue loss* as the appropriate performance evaluation metric for this task of hierarchical product classification, we consider learning a classifier that performs best with respect to this metric.

Standard classifier learning techniques like SVM (Crammer and Singer, 2002) usually try to optimize error rate by minimizing a convex upper bound of the error rate. Therefore, standard classifiers are expected to be optimal with respect to error rate, while is *not* necessarily optimal in terms of our proposed evaluation metric.

We propose a cost-sensitive learning algorithm to optimize the average revenue loss in the training set. The algorithm is based on multi-class SVM with margin re-scaling. It is a general approach for optimizing any misclassification cost. However, margin re-scaling is sensitive to the scaling of loss functions. We propose a **loss normalization** approach to make margin re-scaling achieve good performance in practice. The loss normalization approach is applicable to other classification and structured prediction tasks whenever using structured SVM with margin re-scaling.

2.3.1 Linear classifiers

We consider linear classifiers that have been shown to achieve state-of-the-art performance for document classification. Given an example $x \in R^N$, a linear classifier uses a weight vector $\theta_y \in R^N$ to score each class $y \in Y$ by the inner product $\theta_y^T x$; then predict the class y' with the highest score,

$$y' = f(x) = \arg\max_y \theta_y^T x$$

where θ_y's are parameters of the classifier. We use θ to represent the collection of all θ_y's.

2.3.2 Minimize empirical risk

Corresponding to that many performance evaluation metrics can be formulated as empirical risk in 2.2.2, many learning algorithms can be formulated as minimizing empirical risk in the training set, additionally with parameter regularization (Teo et al., 2010). Assume the parameters of a linear classifier are θ, learning becomes an optimization problem that finds the optimal parameter θ^* that minimizes the empirical risk $R_{em}(D;\theta)$, the average loss in the training set, with parameter regularization to avoid over fitting,

$$\begin{aligned} \theta^* &= \arg\max_\theta \frac{1}{2}||\theta||^2 + R_{em}(D;\theta) \\ &= \arg\max_\theta \frac{1}{2}||\theta||^2 + \frac{1}{m}\sum_{(x,y,y')\in D} L(x,y,y';\theta) \end{aligned} \tag{6}$$

where $L(x,y,y';\theta)$ is the loss function for an example x of class y and predicted into class

y', given the classifier parameters θ.

2.3.3 Minimize average revenue loss

To learn the parameters of a linear classifier, we use multi-class SVM with margin re-scaling that scales the required margin according to the loss function, the revenue loss in our case. We also propose a loss normalization approach to make margin-rescaling work well in practice.

2.3.3.1 Margin re-scaling

Standard multi-class SVM by Crammer and Singer (Crammer and Singer, 2002) indirectly optimizes error rate by minimizing a convex upper bound of average 0-1 loss. Similarly, we try to optimize average revenue loss by minimizing a convex upper bound.

Let the loss function in Eq.(6) be revenue loss, $L(x, y, y'; \theta) = v(x) \cdot L_{yy'}$, then we optimize the average revenue loss directly. Unfortunately, this is a non-convex objective function, which is difficult to solve.

Similar to the *hinge loss* in multi-class SVM (Crammer and Singer, 2002), we use a convex surrogate of the loss function,

$$\max \left\{ 0, \ \max_{y' \neq y} \left\{ L(x, y, y'; \theta) + \theta_{y'}^T x - \theta_y^T x \right\} \right\} \tag{7}$$

For correct classification, we want the score $\theta_y^T x$ of the *true* class y to be higher than the score $\theta_{y'}^T x$ of any *false* class y'. The larger the difference, the more confident our prediction is. In particular, we require the difference to be greater than the the misclassification cost of x from class y to class y': $\theta_y^T x - \theta_{y'}^T x \geq L(x, y, y'; \theta) = v(x) \cdot L_{yy'}$, which is called a *margin* constraint; otherwise, we pay a positive loss for the margin violation. It is easy to prove that this is a convex upper bound of the loss function $L(x, y, y'; \theta)$.

However, Eq.(7) is still non-differentiable. We reformulate the problem as a constrained optimization problem with parameter regularization similarly to (Crammer and Singer, 2002). To absorb the violations of the margin constraints, we use a slack variables ξ_i for each example. Let (x_i, y_i) be the ith training example, we have the following constrained optimization problem,

$$\begin{aligned} \min_{\theta} \frac{1}{2}||\theta||^2 + \frac{C}{m}\sum_{i=1}^{m} \xi_i \\ s.t.\ \forall i,\ \forall y' \neq y_i :\ \theta_{y_i}^T x_i - \theta_{y'}^T x_i \quad &\geq \quad L(x_i, y_i, y') - \xi_i \\ \xi_i \quad &\geq \quad 0 \end{aligned} \tag{8}$$

where C is the regularization hyper parameter, $L(x_i, y_i, y') = v(x_i)L_{y_i y'}$ is the loss function. It is not difficult to prove that the solution to the above problem is also the solution to the unconstrained optimization problem as Eq.(6) with the hinge loss Eq.(7). So $\xi_i \geq L(x_i, y_i, y')$ for any y'; the objective is an upper bound of the empirical risk $\frac{1}{m}\sum_i L(x_i, y_i, y_i')$ (up to a constant factor C).

Eq.(8) is a special case of structured SVM with **margin-rescaling** (Tsochantaridis et al., 2006). It is also a generalization of multi-class SVM (Crammer and Singer, 2002), which uses 0-1 loss $L(x_i, y_i, y') = [y_i \neq y']$ in Eq.(8).

Multi-class SVM with margin re-scaling shows a general way to do cost-sensitive learning, by scaling the margin proportional to any loss function, hence minimizing the upper bound of the average loss, empirical risk. In our experiments, we compare margin rescaling by our proposed loss function (denoted as **REVLOSS**) to by three other loss functions, (1) the **0-1** loss, $L(x_i, y_i, y') = [y_i \neq y']$, giving the standard multi-class SVM; (2) the **VALUE** loss, that is the revenue of the product $L(x_i, y_i, y') = v(x_i)[y_i \neq y']$. This assumes misclassification causes losing the whole potential revenue. The misclassification cost does not depend on hierarchical distance of the true class and the predicted class. (2) the **TREE** loss, the height of the lowest common ancestor of the true class and the predicted class $L(x_i, y_i, y') = L_{y_i y'}$, with $L_{yy} = 0$. The misclassification cost does not depend on the product revenue.

2.3.3.2 Loss normalization

As §2.2.5 in (Tsochantaridis et al., 2006) pointed out, margin re-scaling is sensitive to the scaling of the loss function. One should be careful in calibrating the scaling of the loss function with respect to the scaling of the feature values.

Indeed, in our experiments, margin re-scaling using the original revenue values in dollars performs significantly worse than standard multi-class SVM, even after rescaling the revenue by different units, say million \$. Because the annual revenue of individual products span from the order of hundred dollars to million dollars, hence so are the revenue loss. Therefore the required margin for high revenue products can be much larger than those for low revenue products. This means either (1) the required margin for large revenue products are too large, but the feature values are word frequencies most of which are 1 to 2, which tends to force the norm of the parameters to be large to reach large margin; or (2) the required margin for small revenue products are too small, close to zero, which essentially don't apply margin constraints for them. Both cases lead to larger generalization error.

To adjust the difference between the influence of extreme high revenue loss and that of extreme low revenue loss, we propose to linearly scale the loss function to a fixed range $[M_{min}, M_{max}]$,

$$L^s(x, y, y') = M_{min} + \frac{L(x, y, y') - L_{min}}{L_{max} - L_{min}} \cdot (M_{max} - M_{min})$$

where $L_{min} = \min\{L(x, y, y')\}$ is the minimum possible revenue loss in the training set, calculated as the product of the minimum revenue value times the minimum loss ratio; and similarly $L_{max} = \max\{L(x, y, y')\}$ is the maximum possible revenue loss. So that L_{min} is mapped to M_{min}, and L_{max} is mapped to M_{max}.

It is not hard to see that minimizing the empirical risk with normalized loss is equivalent to minimizing the original empirical risk in terms of choosing θ, since they differ only in a linear transformation.

We recommend to use $L_{min} = 1$, and tuning L_{max} in a development set. In this way, the objective in Eq.(8) with normalized loss has two appealing properties: (1) It upper bounds 0-1 loss, which makes the optimization meaningful for minimizing error rate; (2) It upper bounds the empirical risk with normalized loss $\sum_i L^s(x_i, y_i, y')$ whose minimization is equivalent to the original empirical risk $\sum_i L(x_i, y_i, y')$, which makes the optimization meaningful for minimizing the original empirical risk, the average revenue loss in our case. The key to prove both properties is $\xi_i \geq L(x_i, y_i, y')$.

Regarding implementation, the large scale of the task requires highly efficient optimization method. We solve Eq.(8) in dual space along the line of (Crammer and Singer, 2002; Keerthi et al., 2008), with a slight modification of the dual objective to reflect the desired margin as the new loss function $v(x^{(i)})L_{y^{(i)}y'}$. We choose *sequential dual* method (Keerthi et al., 2008) for optimization. Our implementation uses the LIBLINEAR package (Fan et al., 2008) and modifies the sequential dual solver specified with option "-s 4". Empirically, we find it very efficient on our large dataset, and is more than ten times faster than cutting-plane method (Tsochantaridis et al., 2006).

An alternative to margin re-scaling is slack re-scaling (see (Tsochantaridis et al., 2006) §2.2.5 for more discussion). However, our implementation with the cutting plane method in SVM^{struct} (Tsochantaridis et al., 2006) is way too slow on the large dataset in our experiments. While we have very efficient sequential dual solvers in LIBLINEAR for margin re-scaling. Hence we sought to improve margin re-scaling, which is widely used in structured prediction problems (Roller, 2004). Our **loss normalization** approach is applicable to general classification and structured prediction tasks, whenever using structured SVM with margin-rescaling, especially when the loss function spans a wide range.

In experiments we compare three loss normalization approaches: **IDENTITY** approach that uses original revenue in dollars, **UNIT** normalization that scales the revenue by different units, and our proposed **RANGE** normalization.

2.4 Experiments

Experiments show that our cost-sensitive learning algorithm with margin re-scaling and loss normalization outperforms standard multi-class SVM, in terms of our proposed evaluation metric, average revenue loss.

2.4.1 Dataset

2.4.1.1 The UNSPSC dataset

We do experiments on a large dataset of more than 1 million products in 1073 classes. Each product has a textual description of 5 fields: *manufacturer name*; *UNSPSC code*

# examples	1,439,097
# leaf (4th level) classes	1073
# 3rd level classes	300
# 2nd level classes	99
# 1st level classes	33
avg.±std. of description length	39.5 ± 23.6
# features	784,813

Table 1: Statistics of the UNSPSC dataset haha

that is the class label explained below; *product name*; *description*; and *detailed description* that is possibly empty. The dataset is collected from multiple online marketplaces oriented for Department of Defense and Federal government customers, including GSA advantage and DoD EMALL. It covers a wide range of products and services.

Each product is labeled by an 8-digit code as belonging to a leaf class in a large taxonomy. The taxonomy is called *United Nations Standard Product and Service Code* (**UNSPSC**)[2] . It is the de factor standard in US industry for hierarchical classification of general products and services. It has four levels, representing *segment*, *family*, *class*, and *commodity*, respectively. Each node in a level is specified by a 2-digit code and a text description. We identify a leaf class by an 8-digit code, concatenating the 2-digit codes along the path from the first level to the fourth level.

The whole UNSPSC taxonomy has more than 17,000 leaf categories and is still increasing. Our dataset covers products in more than one thousand classes. We discard small classes with less than 10 products, and consider a sub taxonomy with 1073 leaf classes. The statistics of the preprocessed dataset used in our experiments is shown as Table 1. The class distribution is highly skewed as shown in Figure 2, where the X-axis is the class ranking from 1 to 1073 by size, that is, the number of examples in the class; and the Y-axis is the log2 of the class size.

2.4.1.2 Revenue generation

The dataset does not come with the expected annual revenue for products. So we simulate the revenue, similarly to other works in cost-sensitive learning (e.g., (Domingos, 1999; Zhou and Liu, 2006)). We first generate the price and sales independently, then multiply

[2] see www.unspsc.org for details.

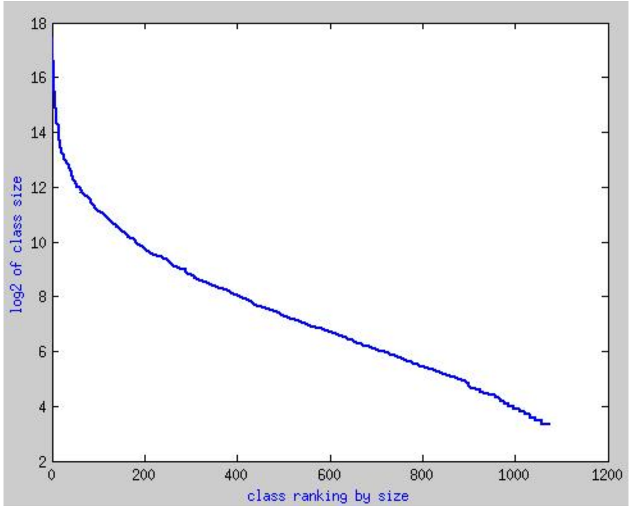


Figure 2: The class distribution in the UNSPSC dataset

them as the revenue.

Our price model assumes the prices of products from one class are drawn from one log-normal distribution $\ln \mathcal{N}(\mu, \sigma^2)$. Different classes have different log-normal price distributions. Log-normal distribution is used in economics to model prices (Lee et al., 2010). To generate the parameters μ and σ^2 for each class, we use two Gamma distributions as prior distributions, respectively. Gamma distribution is commonly used as a conjugate prior for a parameter in Bayesian statistics. We choose $Gamma(k = 1, \theta = 100)$ to sample μ for each class. To generate σ^2 and control the amount of price fluctuations to be moderate for most classes, we let $\sigma = \mu \cdot sigma\_ratio$, and prefer $sigma\_ratio$ to be more likely in $[0.2, 0.5]$. To generate such $sigma\_ratio$'s for each class, we use $Gamma(k = 30, \theta = 0.01)$. After generating μ and σ^2 for a class, we sample prices for the products in that class from the log-normal distribution $\ln \mathcal{N}(\mu, \sigma^2)$.

Similarly, our sales model assumes the sales of products from one class are drawn from one $Pareto(m, k)$ distribution. The survival function is given by $Pr(X > x) = (m/x)^k$, $m > 0, k > 0, x \geq m$, where m is the minimum possible value of x. Pareto distribution is an instance of the power law distribution that has been used to model a wide range of social and natural phenomena, including the relationship between weekly sales and sales ranks of books at Amazon.com (Anderson and Andersson, 2007). To generate the parameters m and k for each class, we again use two priors. To sample the minimum sales m for each class, we use a Weibull distribution that is often used to model extreme value distributions. We choose $Weibull(\lambda = 2, k = 5)$ such that m has higher prior in $[1, 3]$ with thousand as unit. To sample the shape parameter k, we again use Gamma distribution. We choose $Gamma(k = 50, \theta = 0.1)$ to let k have high prior in $[4, 6]$ so that most classes have reasonably skewed sales distribution.

2.4.1.3 Preprocessing

Data cleaning. We remove duplicate product records in the dataset by comparing both manufacturer names and product names. We perform tokenization using simple delimiter patterns like punctuations and spaces. All tokens are transformed into lower case.

Feature extraction. We use word frequencies as features. Each token corresponds to a feature. To utilize the field information of manufacture name and product name, we add a special prefix like "$MNFT_" to each token appearing in both fields, respectively.

2.4.2 Results

2.4.2.1 Experimental setting

We randomly split the UNSPSC dataset into *training*, *development*, and *test* set by size ratios 4 : 3 : 3, and by stratified sampling per classes. Development set is used for selecting optimal parameters. The regularization parameter C is selected from $\{0.01, 0.1, 1\}$, as we empirically find that the performance is usually best with $C = 0.1$, with monotonically decreasing performance on both sides. With larger C, the optimization also takes much longer on our large dataset. Similarly, we select revenue rescaling unit in $\{10^2, 10^4, 10^6, 10^7\}$ in dollars; loss normalization range as $[1, M_{max}]$, where M_{max} in $\{2, 5, 10, 20\}$. The best M_{max} is usually 10.

We generate 5 sets of revenues for the products using our revenue model. All results reported are averaged over 5 runs of experiments with different sets of revenue samples. Since the UNSPSC taxonomy is a 4-level balanced tree, there are only four different hierarchical distances between two leaf nodes. We specify loss ratios $L_{yy'}$ as 0.2, 0.4, 0.6, and 0.8, for hierarchical distances between the true class and the predicted class as 2, 4, 6 and 8, respectively.

2.4.2.2 Results and discussion

Table 2 compares the performance of multi-class SVM with margin re-scaling by *four loss functions* and *three loss normalization* approaches discussed in 2.3.2, in terms of our proposed evaluation metric, average revenue loss. The table's columns correspond to different loss functions. The *0-1* loss corresponds to the *baseline*, standard multi-class SVM. The rows correspond to different loss normalization approaches.

The combination of *REVLOSS* margin re-scaling with *RANGE* loss normalization achieves the smallest average revenue loss. It reduces as much as **7.88%** average revenue loss incurred by standard SVM, which is significant with *pairwise one-tailed t-test* at significant level $p < 0.01$. Such an amount of reduction is remarkable, because it is achieved when the error rate of standard SVM is already as low as 3.8% (see 4), which means most products already have zero revenue loss, so any further reduction of revenue loss is non-trivial.

Comparing margin-rescaling with different *loss functions*, TREE loss increases average revenue loss, while VALUE loss achieves a significant reduction of average revenue loss, which is taken further by REVLOSS. This shows that most of the revenue loss reduction comes from exploiting the revenue of individual products; differentiating misclassification cost into different classes in REVLOSS further reduces the revenue loss slightly.

Comparing different *loss normalization* approaches (only applicable to VALUE and REVLOSS), RANGE normalization effectively improves the performance than UNIT rescaling and IDENTITY (no normalization).

To further look into what products and how the four *loss functions* with RANGE normalization tend to misclassify, Table 3 describes the mean revenue and mean tree loss (the height of the lowest common ancestor of the true class and the predicted class) of the *misclassified products* by those approaches with a single random set of revenue values. Comparing to standard SVM (0-1 loss) and TREE loss, VALUE and REVLOSS are able to tap into the product revenue information, and tend to misclassify products of significantly lower revenues on average. They tend to trade the accuracy of low revenue products for the accuracy of high revenue products; even though the final error rate of VALUE and REVLOSS is slightly higher than that of 0-1 (see Table 4), the average revenue loss of VALUE and REVLOSS is lower. On the other hand, TREE loss achieves smallest mean

	0-1	TREE	VALUE	REVLOSS
IDENTITY	4.745	4.964	47.708	48.082
UNIT			5.092	5.082
RANGE			4.387	**4.371**

Table 2: Average revenue loss of different algorithms. All revenue loss reduction and increase compared to standard SVM (0-1) are significant at $p < 0.01$ with pairwise one-tailed t-test at the corresponding direction (either decrease or increase loss). Revenues in both tables are of unit thousand dollar ($K\$$).

measure	0-1	TREE	VALUE	REVLOSS
mean revenue	124.4±192	116.1±185	**111.5±172**	112.9±172
mean tree loss	2.342	**2.156**	2.330	*2.328*

Table 3: Statistics of misclassified products by different algorithms

tree loss among the misclassified products. REVLOSS combines the advantage of both TREE and VALUE loss, yielding a desirable behaviour: If it cannot classify all products correctly, it tends to sacrifice the performance of low revenue products; If it cannot classify a product into the true class, it tends to place the product into a class as close as possible.

Table 4 shows the performance of four different margin rescaling approaches with RANGE loss normalization, in terms of three common evaluation metrics. Standard SVM (0-1 loss) performs best. VALUE and REVLOSS have slightly lower but comparable performance. It might be confusing that TREE loss performs worse, even in terms of average tree loss that it is aiming to minimize. However, from Table 3, TREE loss achieves smallest mean tree loss among misclassified products, so it does its job; but unfortunately it tends to increase the error rate too much, thereby increasing the average tree loss in the whole test set.

We also explored classification methods that exploits the hierarchical structure. In particular, we experimented with cascading classifiers in a top-down way. We cascade two-level

measure	0-1	TREE	VALUE	REVLOSS
error rate	3.8	4.3	3.9	3.9
mean error rate	11.8	13.1	12.1	12.0
avg. tree loss	0.089	0.092	0.092	0.090

Table 4: Performance of different algorithms by common evaluation metrics

classifiers. The results show similar amount of performance improvement with REVLOSS margin-rescaling than standard SVM, which is not shown here due to page limits. However, cascading classifiers leads to error propagation. Both error rate and mean revenue loss is slightly higher than the above flat approaches. We leave as future work to explore other hierarchical approaches like global approaches that have been shown to have higher accuracy than flat classifiers, as they leads to larger model and requires more computing resource for large taxonomies.

2.5 Related works

Product classification. (Shen et al., 2012) study learning a hierarchy from the data for product classification. (Kannan et al., 2011) improves product classification using images. (Shen et al., 2009) classifies product queries. Those works usually use common evaluation metrics, while we study the appropriate performance evaluation in product classification when a vendor's business goal is to maximize revenue, and the corresponding cost-sensitive learning that optimizes the proposed metric.

Hierarchical classification. On *performance evaluation* for hierarchical classification, see (Costa et al., 2007) and (Sun and Lim, 2001) for detailed reviews. Most works generalize evaluation metrics designed for binary classification like precision, recall, and F1-score to multi-class and hierarchical classification case. They try to be applicable to general tasks. Though we design an evaluation metric that tailors the specific task of product classification, the proposed metric has a very general form involving both example-dependent cost and class-dependent cost. Most works tend to propose metrics similar to F1 score that are non-linear function of the loss, hence they are not ideal for optimization, unlike our treatment of performance evaluation metric as empirical risk that is suitable for optimization. On *classifier learning*, see (Freitas and de Carvalho, 2007) for a survey of numerous works. (Dekel et al., ; Cai and Hofmann, 2004) extend the max-margin principle of SVM to hierarchical classification. Our experiment on cascading classifiers in a top-down way is proposed in (Dumais and Chen, 2000).

Cost-sensitive learning. Most work studies misclassification costs that are either example-dependent (Zadrozny et al., 2003) or class-dependent (Elkan, 2001; Domingos, 1999; Zhou and Liu, 2006), according to Zhou's nomenclature (Zhou and Liu, 2006). The former give misclassification costs according to different examples. The latter give different misclassification costs according to different predicted class, while the misclassification cost into a particular false class is the same for all examples within a single class. Very few works (Beygelzimer et al., 2008) study both of them. We study misclassification cost that is both example-dependent and class-dependent. It specializes to one type of them if we set the other type of cost uniform.

2.6 Summary

This chapter studies hierarchical product classification. In particular, we investigate two problems, *performance evaluation* and *learning*, in a synergistic way, under a unified view of empirical risk. Performance evaluation chooses an appropriate misclassification cost. Learning minimizes the average misclassification cost. We emphasizes the importance to design an appropriate performance evaluation metric for a real world task, otherwise we are optimizing the wrong objective. We show how to apply such a synergistic way to address the specific task of hierarchical product classification, and demonstrate its effectiveness by experiments on a large dataset. We obtain general insight into how and why several common evaluation metrics can be misleading, which is applicable to the treatment of performance evaluation of other real world tasks. We propose a general cost-sensitive learning algorithm that minimizes the upper bound of any loss functions, using multi-class SVM with margin re-scaling and loss normalization. The loss normalization approach is also applicable to general classification and structured prediction tasks when using structured SVM with margin re-scaling.

Our work is an application of cost-sensitive learning. Very few works study both class-dependent and example-dependent misclassification cost, especially in a practical scenario as we do. However, application scenarios involving both types of cost are not rare, even becoming more and more common in big data era, forming an emerging class of applications for large scale information and knowledge management. For example, Google Inc. detects and classifies adversarial advertisements that violates Adword policies into fine-grained classes for the benefits and safety of users (Sculley et al., 2011). The misclassification cost of an advertisement can depend on its potential revenue, and the relation between the true class and the predicted class. Another example is that a company manages numerous clients of different potential values by a taxonomy. We conjecture that such applications will become increasingly pervasive, because both taxonomies and value measures play increasingly bigger roles in modern economy and big data era. Taxonomies like Wikipedia, semantic web and patent taxonomies are widely used to organize information. On the other hand, items of monetary values abound everywhere in modern economic world. Moreover, we can assign values to them as the importance of classifying them correctly. Say in image or document classification, we assign higher values to items from certain websites to emphasize their correct classification. Discriminating values of different pieces of information takes care of important information given the sheer amount of data available nowadays.

Chapter 3

Image Description using Bipartite Cross-modal Association Structure

3.1 Introduction

Our second study straddles language and vision, the two fundamental modalities with which human perceive the world.

The use of multimodal web data has been a recurring theme in many recent studies integrating language and vision, e.g., image captioning (Ordonez et al., 2011; Hodosh et al., 2013; Mason and Charniak, 2014; Kuznetsova et al., 2014), text-based image retrieval (Rasiwasia et al., 2010; Rasiwasia et al., 2007), and entry-level categorization (Ordonez et al., 2013; Feng et al., 2015).

However, much research integrating complex textual descriptions to date has been based on datasets that rely on substantial human curation or annotation (Hodosh et al., 2013; Rashtchian et al., 2010; Lin et al., 2014), rather than using the web data in the wild as is (Ordonez et al., 2011; Kuznetsova et al., 2014). The need for human curation limits the potential scale of the multimodal dataset. Without human curation, however, the web data introduces significant noise. In particular, everyday captions often contain extraneous information that is not directly relevant to what the image shows (Kuznetsova et al., 2013b; Hodosh et al., 2013).

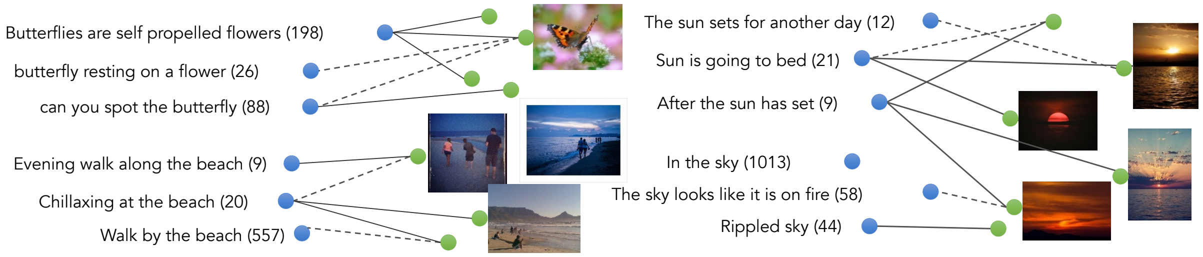

Butterflies are self propelled flowers (198)
butterfly resting on a flower (26)
can you spot the butterfly (88)
Evening walk along the beach (9)
Chillaxing at the beach (20)
Walk by the beach (557)
The sun sets for another day (12)
Sun is going to bed (21)
After the sun has set (9)
In the sky (1013)
The sky looks like it is on fire (58)
Rippled sky (44)

Figure 3: The image-caption association graph of *Déjà Image-Captions*. Solid lines represent original captions and dotted lines represent paraphrase captions. This corpus reflects a rich spectrum of everyday narratives people use in online activities including figurative language (e.g., *"Sun is going to bed"*), casual language (e.g., *Chillaxing at the beach"*), and conversational language (e.g., *"Can you spot the butterfly"*). The numbers in the parenthesis show the cardinality of images associated with each caption. Surprisingly, some of these descriptions are highly expressive, almost *creative*, and yet not unique — as all these captions are repeated almost verbatim by different individuals describing different images.

In this chapter, we present a new approach to harvesting a large-scale, high quality image-caption corpus that makes a better use of already existing web data with no additional human efforts. Figure 3 shows sample captions in the resulting corpus, e.g., *"butterfly resting on a flower"* and *"evening walk along the beach"*. Notably, some of these are figurative, e.g., *"rippled sky"* and *"sun is going to bed."*

The key idea is to focus on *Déjà Image-Captions*, i.e., naturally existing image captions that are *repeated almost verbatim* by more than one individual for different images. The hypothesis is that such captions represent common visual content across multiple images, hence are more likely to be free of unwanted extraneous information (e.g., specific names, time, or any other personal information) and better represent visual concepts. A surprising aspect of our study is that such a strict data filtration scheme can still result in a large-scale corpus; sifting through 760 million image-caption pairs, we harvest as many as 4 million image-caption pairs with 180K unique captions.

The resulting corpus, *Déjà Image Captions*, provides several unique properties that complement human-curated or crowd-sourced datasets. First, as our approach is fully automated, it can be readily applied to harvesting a new dataset from the ever changing multimodal web data. Indeed, a recent internet report estimates that billions of new photographs are being uploaded daily (Meeker, 2014). In contrast, human-annotated datasets are costly to scale to different domains.

Second, datasets that are harvested from the web can complement those based on prompted human annotations. The latter in general are literal and mechanical readings of the visual scenes, while the former reflect a rich spectrum of natural language utterances in everyday narratives, including figurative, pragmatic, and conversational language, e.g., *"can you spot the butterfly"* (Figure 3). Therefore, this dataset offers unique opportunities for grounding figurative and metaphoric expressions using visual context.

In conjunction with the new corpus, publicly shared at `http://www.cs.stonybrook.edu/~jianchen/deja.html`, we also present three new tasks: *visually situated paraphrases* (§3.5); *creative image captioning* (§3.7), and *creative visual paraphrasing* (§3.7). The central algorithm component in addressing all these tasks is a simple and yet effective approach to image caption transfer that exploits the unique association structure of the resulting corpus (§3.3).

Our empirical results collectively demonstrate that when the web data is available at such scale, it is possible to obtain a large-scale, high-quality dataset with significantly less noise. We hope that our approach would be only one of the first attempts, and inspire future research to develop better ways of making use of ever-growing multimodal web data. Although it is unlikely that the automatically gathered datasets can completely replace the curated descriptions written in a controlled setting, our hope is to find ways to complement human annotated datasets in terms of both the scale and also the diversity of the domain and language.

3.2 Dataset - captions in repetition

Our corpus consists of three components (Table 5):

main set The first step is to crawl as many image-caption pairs as possible. We use flickr.com search API to crawl 760 million pairs in total. The API allows searching images within a given time window, which enables exhaustive search over any time span. To ensure visual correspondence between images and captions, we set query terms using 693 most frequent nouns from the dataset of Ordonez et al. (2011), and systematically slide time windows over the year 2013.[1] For each image, we segment its title and the first line of its description into sentences.

The crawled dataset at this point includes a lot of noise in the captions. Hence we apply initial filtering rules to reduce the noise. We retain only those image-sentence pairs in which the sentence contains the query noun, and does not contain personal information indicators such as first-person pronouns. We want captions that are more than simple keywords, thus we discard trivial captions that do not include at least one verb, preposition, or adjective.

The next step is to find captions in repetition. For this purpose, we transform captions into *canonical forms*. We lemmatize all words, convert prepositions to a special token "IN"[2] , and discard function words, numbers, and punctuations. For instance, "*The bird flies in blue sky*" and "*A bird flying into the blue sky*" have the same canonical form, "*bird fly IN blue sky*". We then retain only those captions that are repeated with respect to their canonical forms by more than one user, and for distinctly different images to ensure the generality of the captions.

Retaining only captions that are repeated verbatim may seem overly restrictive. Nonetheless, because we start with as many as 760 million pairs, this procedure yields nearly 180K unique captions associated with nearly 4M images.[3] What is more surprising, as will be shown later, is that many of these captions are highly expressive. Table 6 shows the distribution of the number of images associated with each caption.[4] The median and mean are 10 and 22.4 respectively, showing a high degree of connectivities between captions and images.

paraphrase set Our dataset collection procedure finds *one-to-many* relations between captions and images. To extend these relations to *many-to-many*, we introduce *visually-situated paraphrases* (or *visual paraphrases* for shorthand) (§3.5). A visual paraphrase relation is a triple (i, c, p), where image i has an original caption c, caption p is the visual

[1] To ensure enough number of images are associated with each caption, we further search captions with no more than 10 associated images across *all* years.

[2] We do this transformation so as not to over-count unique captions with trivial variations, but merging prepositions can sometimes combine prepositions that are not semantically compatible. We therefore also keep original captions with original prepositions.

[3] We also keep user annotated image tags if available.

[4] Without counting additional edges created by visual paraphrasing (§5).

set	# captions	# images
MAIN	176,780	3,967,524
PARAPHRASE	7,570 human-annotated triples 353,560 auto-generated triples	
FIGURATIVE	6,088 quotations	180,185
	18,179 quotations + predicted figurative captions	413,698

Table 5: Corpus Statistics

	mean	std	25%	50%	75%	max
#imgs.	22.4	47.6	4	10	25	4617
#tokens	4.9	3.3	3	4	5	178

Table 6: Percentiles of the image count associated with each caption and the number of tokens in each caption.

paraphrase for c situated in image i. We collect visual paraphrases for sample images in our dataset, using both crowd sourcing (7,570 triples) and an automatic algorithm (353,560 triples) (see §3.5 for details). Figure 4 shows example visual paraphrases.

Formally, our corpus represents a bipartite graph $G = (T, V, E)$, in which the set of captions T and the set of images V are connected by typed edges $e(c, i, t)$, where caption $c \in T$, image $i \in V$, and edge type $t \in \{original, paraphrase\}$, which denotes whether the image-caption association is given by the original caption or by a visual paraphrase.

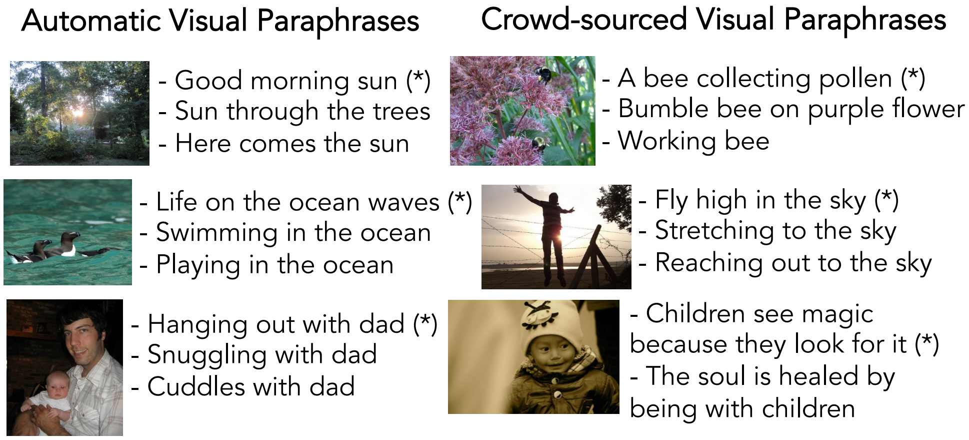

Automatic Visual Paraphrases
Crowd-sourced Visual Paraphrases
- Good morning sun (*)
- Sun through the trees
- Here comes the sun
- A bee collecting pollen (*)
- Bumble bee on purple flower
- Working bee
- Life on the ocean waves (*)
- Swimming in the ocean
- Playing in the ocean
- Fly high in the sky (*)
- Stretching to the sky
- Reaching out to the sky
- Hanging out with dad (*)
- Snuggling with dad
- Cuddles with dad
- Children see magic because they look for it (*)
- The soul is healed by being with children

Figure 4: Example visual paraphrases: automatic (left) and crowd-sourced (right). The first caption marked with * indicates the original caption of the corresponding image. Some paraphrases are not strictly equivalent to the original caption if considered out of context, while they are pragmatically adequate paraphrases given the image.

figure of speech	#caps.	% in fig.	example (#imgs.)
quotation&idiom	70	41%	The early bird gets the worm (77)
personification	43	25%	Meditating cat (38)
metaphor	24	14%	Wine is the answer (7)
question	18	11%	Do you see the moon (82)
dialog	11	6%	Hello little flower (37)
anaphora	6	4%	Beads, beads and more beads (62)
simile	5	3%	The lake is like glass (23)
hyperbole	1	< 1%	In the land of a billion lights (3)

Table 7: Distribution of figurative language out of 1000 random captions (171 figurative captions in total). The column "% in fig." shows the percentages of different figures of speeches among figurative captions. They add up to more than 100% because some captions uses more than one figures of speeches.

polarity	% in all caps.	mean/median #imgs. per cap.	example (#imgs)
pos.	8%	20 / 8	Happy bride and groom (282) The rock and pool, is nice and cool (4)
neg.	2%	19.5 / 7	Bad day at the office (269) Crying lightning (147)

Table 8: Distribution of caption sentiment. The polarity is determined by comparing number of positive words and negative words (>: positive; <: negative) according to a sentiment lexicon (Wilson et al., 2005) (counting only words of *strong* polarity).

figurative set We find that many repeating captions are surprisingly lengthy and expressive, most of which turn out to be idiomatic expressions and quotations, e.g., *"faith is the bird that feels the light when the dawn is still dark"* from Tagore's poem. We look up goodreads.com and brainyquotes.com to identify 6K quotation captions illustrated by 180K images. We also present a manual labeling on a small subset of the data (Table 7) to provide better insights into the degree and types of figurative speech used in natural captions. Using these labels we build a classifier (§3.7) to further detect 18K figurative captions associated with 410K images.

Insights As additional insights into the dataset, Figure 5 shows statistics of the visual content, Table 9 shows syntactic types of the captions, and Table 8 shows positive and negative sentiment in captions.

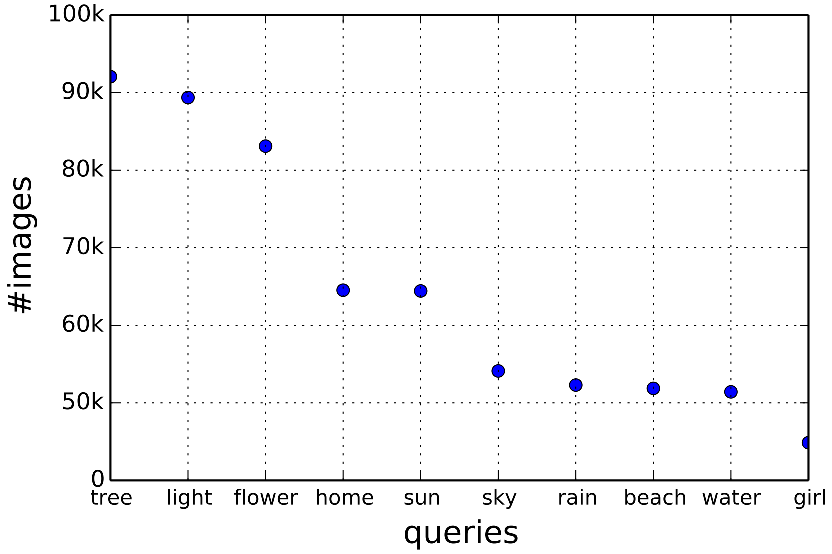


Figure 5: Top 10 queries with the largest number of images and unique captions

type	%caps.	%imgs.	mean #imgs.	std #imgs.
verb	45%	44%	22	9
	be, have, do, look,		Sky is the limit (3057)	
	go, make, come, get,		Home is where the heart is (2480)	
	wait, take, love, play,		Lunch is served (2443)	
	walk, fly, see, watch,		Let them eat cake (2193)	
	find, live, sleep, fall		Follow the yellow brick road (2077)	
prep	44%	41%	21	9
	in, of, on,		On the road (4617)	
	at, with, for,		After the rains (4450)	
	from, by,		Under the bridge (3443)	
	over, through		At the beach (3203)	
adj	11%	15%	30	15
	old, little, new,		Home sweet home (2398)	
	red, blue, more,		Good morning sun (1122)	
	white, big, beautiful,		Cabbage white butterfly (976)	
	black		Next door neighbors (838)	

Table 9: Statistics on the syntactic composition of captions. *verb*: captions with at least one verb. *prep*: prepositional phrases (without any verbs). *adj*: adjective phrases (without any verbs and prepositions). For each caption type, we also show the *top words* that appear in the most number of captions (left), and the *top captions* that are associated with largest number of images (right).

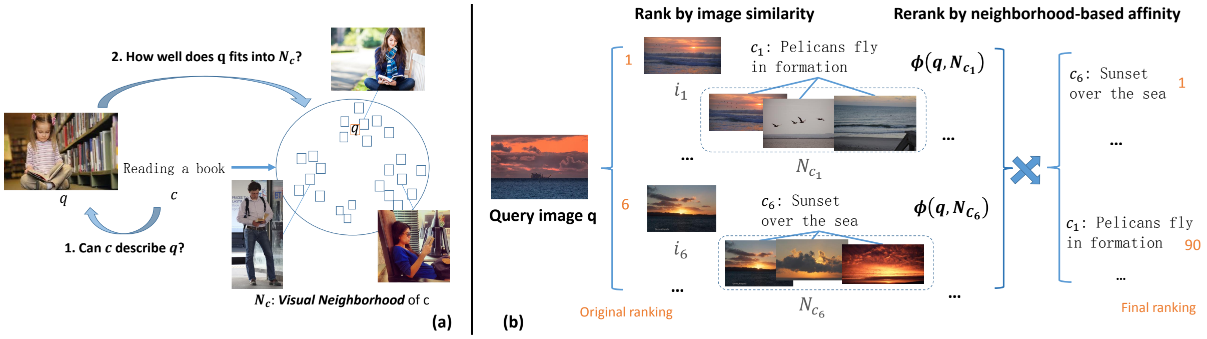

2. How well does q fits into N_c?
q
Reading a book
c
1. Can c describe q?
N_c: Visual Neighborhood of c
(a)
(b)
Rank by image similarity
Rerank by neighborhood-based affinity
Query image q
1
c_1: Pelicans fly in formation
i_1
$\phi(q, N_{c_1})$
N_{c_1}
...
6
c_6: Sunset over the sea
i_6
$\phi(q, N_{c_6})$
N_{c_6}
Original ranking
c_6: Sunset over the sea
1
c_1: Pelicans fly in formation
90
Final ranking

Figure 6: (a) Using the association structure, we retrieve a caption for which the query image is likely to be a *prototypical* visual rendering. We hypothesize that there can be multiple visual prototypes of a caption. (b) Reranking by visual neighbourhood proximity.

3.3 Image captioning using association structure

We demonstrate the usefulness of the association between images and captions via retrieval-based image captioning. Given a query image q and the corpus $G = (T, V, E)$, the task is to find a caption $c \in T$ that maximizes an affinity function $\mathcal{A}(q, c)$, which measures how well the caption c fits the query image q,

$$c^* = \arg\max_{c \in T} \{\mathcal{A}(q, c)\} \tag{9}$$

Visual Neighborhood: Each textual description, e.g., *"reading a book"*, can associate with many different visual instantiations (Figure 6a). Our dataset $G = (T, V, E)$ serves as a database to navigate the possible visual instantiations of descriptive captions as observed in online photo sharing communities. Let $\mathcal{N}_c = \{i | e(c, i, original) \in E\}$ denote the set of adjacent nodes (i.e., visual instantiations) of a caption c. To quantify how well a caption c describe a query image q, we propose to examine caption c's visual neighborhood $\mathcal{N}_c$ as provided in our dataset. Concretely, the affinity $\mathcal{A}(q, c)$ of a query image q to a caption c is a function $\phi(q, \mathcal{N}_c)$ of q and the visual neighborhood $\mathcal{N}_c$ defined as:

$$\mathcal{A}(q, c) = \phi(q, \mathcal{N}_c) = \frac{1}{\sigma} \sum_{i=1}^{\sigma} sim(q, \mathcal{N}_c^i) \tag{10}$$

where σ is a parameter; $sim(\cdot, \cdot)$ is a similarity function of two images; and $\mathcal{N}_c = [\mathcal{N}_c^1, \mathcal{N}_c^2, ..., \mathcal{N}_c^{|\mathcal{N}_c|}]$ is sorted by $sim(q, \mathcal{N}_c^i)$ in *descending* order.

Figure 6a illustrates the key insight: instead of directly transferring the caption of the single image with the closest visual similarity to the query image (Ordonez et al., 2011), we propose to retrieve a caption based on the aggregated visual similarity between its visual neighborhood and the query image. The idea is to prefer a caption for which the query image is likely to be a *prototypical* visual rendering (Ordonez et al., 2013; Deselaers and Ferrari, 2011), hence avoid an unusual association between the text and the visual information. Also, we hypothesize that there could be several diverse visual prototypes of any given textual description c, so we focus on only the top σ nearest members of $\mathcal{N}_c$.

We apply the neighborhood-based affinity for image captioning via reranking (Figure 6b): first we retrieve a pool of K candidate captions by finding top K closest images based on their direct visual similarity to the query image, then compute the neighborhood-based affinity to rerank the captions.[5] The proposed approach is similar in spirit to the non-parametric K nearest neighbor approach of (Boiman et al., 2008) in modeling image-to-concept similarity rather than image-to-image similarity, but differs in that our work is in the context of image description generation rather than classification.

3.4 Experiments: association structure improves image captioning

Baselines: The proposed approach (to be referred as ASSOC) requires one-to-many mappings between captions and images *at scale* — a unique property of our dataset. We compare against two baselines: instance-based retrieval of (Ordonez et al., 2011) (INSTANCE) and Kernel Canonical Correlation Analysis (KCCA) (Hardoon et al., 2004; Hodosh et al., 2013). We implement KCCA with Hardoon's code[6] . We use a linear kernel since non-linear kernels like RBF showed worse performance.

[5] We set $K = 100$ and choose parameter σ using a held-out development set of 300 images. If there are less than σ available images, we use them all.

[6] http://www.davidroihardoon.com/Professional/Code_files/kcca_package.tar.gz

method	BLEU	METEOR
INSTANCE	*0.125*	*0.029*
KCCA	0.118	0.024^{**}
ASSOC$^{\mathsf{gi}}$ w/ all	0.130	0.031
ASSOC$^{\mathsf{g+t}}$ w/ all	0.133	0.030
ASSOC$^{\mathsf{ti}}$ w/ all	0.126	0.029
ASSOC$^{\mathsf{gi}}$ w/ σ	0.172^{**}	0.033^{*}
ASSOC$^{\mathsf{g+t}}$ w/ σ	0.159^{**}	0.033^{*}
ASSOC$^{\mathsf{ti}}$ w/ σ	**0.184****	**0.034****

Table 10: Automatic evaluation for image captioning: The superscripts denote the image feature for reranking; gi: GIST; ti: Tinyimage; g+t:= gi + ti. We report the best setting (gt) for INSTANCE and KCCA. Results statistically significant compared to INSTANCE with two-tailed t-test are indicated with * ($p < 0.05$) and ** ($p < 0.005$).

Configurations: For image features, we follow (Ordonez et al., 2011) to experiment with two global image descriptors and their combination: a) the GIST feature that represents the dominant spatial structure of a scene (Oliva and Torralba, 2001); b) the Tinyimage feature that represents the overall color of an image (Torralba et al., 2008); c) a combination of the two. We compute the similarity as $sim(Q, I) = -\|Q - I\|^2$. The INSTANCE and the KCCA approaches use the feature combination. The ASSOC approach also use the combination for preparing candidate captions, but can use different features for reranking.

Dataset: We randomly sample 1000 images with unique captions as test set. The rest of the corpus is the pool of caption retrieval after *discarding*: (1) the original caption c and all of its associated images, to avoid potential unfair advantage toward ASSOC and (2) the 10K captions used for training KCCA and all of their associated images (about 280K).

Evaluation. Automatic evaluation remains to be a challenge (Elliott and Keller, 2014). We report both BLEU (Papineni et al., 2002) at 1 without brevity penalty, and METEOR (Banerjee and Lavie, 2005) with balanced precision and recall. Table 10 shows the results: the ASSOC approach (w/ σ) significantly outperforms the two baselines. The largest improvement over INSTANCE is 60% higher in BLEU, and 44% higher in METEOR, demonstrating the benefit of the innate association structure of our corpus. Using *all* visual neighborhood (ASSOC w/ all) does not yield as strong results as selective neighborhood

reranking feature	INSTANCE	ASSOC
gi	42%	58%
g+t	50%	50%
ti	46%	54%

Table 11: Human evaluation for image captioning: the % of cases judged as visually more *relevant*, in pairwise comparisons. gi: GIST; ti: Tinyimage; g+t:= gi+ti.

(ASSOC w/ σ), confirming our hypothesis that each visual concept can have diverse visual renderings.

We also compute crowd-sourced evaluation on a subset (200 images) randomly sampled out of the test set. For each query image, we present two captions generated by two competing methods in a random order. Turkers choose the caption that is more relevant to the visual content of the given image. We aggregate the choices of three turkers by majority voting. As shown in Table 11, ASSOC shows overall improvement over baselines, where the difference is more pronounced when reranking is based on feature sets that differ from the one used during the candidate retrieval.

3.4.1 Good and bad examples

Fig.7 (a) shows some *good* examples where the ASSOC approach retrieves captions of better visual relevance than the INSTANCE method for the given query images; and (b) shows some *bad* examples.

Good examples. *Example 1.* By looking at *multiple* images in the visual neighborhoods, the ASSOC approach is able to match the caption "*Castle at dusk*" that is semantically much more accurate than "*Agave in bloom*". The ASSOC method tends to be *semantically more accurate and reliable* than the INSTANCE method that looks at a single most similar image.

Example 2. The ASSOC method tends to retrieve more *general* captions that are more likely be transferable to many images. Since general captions usually have larger number of examples in its associated visual neighborhood, it could be easier to find several images in its neighborhood that are very close to the query image, and hence have higher Visual Neighborhood Closeness. General captions tend to be accurate at a coarser semantic level (cite: entry-level paper).

Bad examples. *Example 3.* The ASSOC method might match a caption with a neighborhood of images of very similar *background* as the query image, while the foreground object (the bridge) in the matched caption is missing in the original image. Such unstability could be due to two reasons: (1) We use global image descriptors. (2) The visual neighborhood of the matched caption has only two images in this case, which might not be enough for estimating Visual Neighborhood Closeness reliably. We address (2) in §3.5.

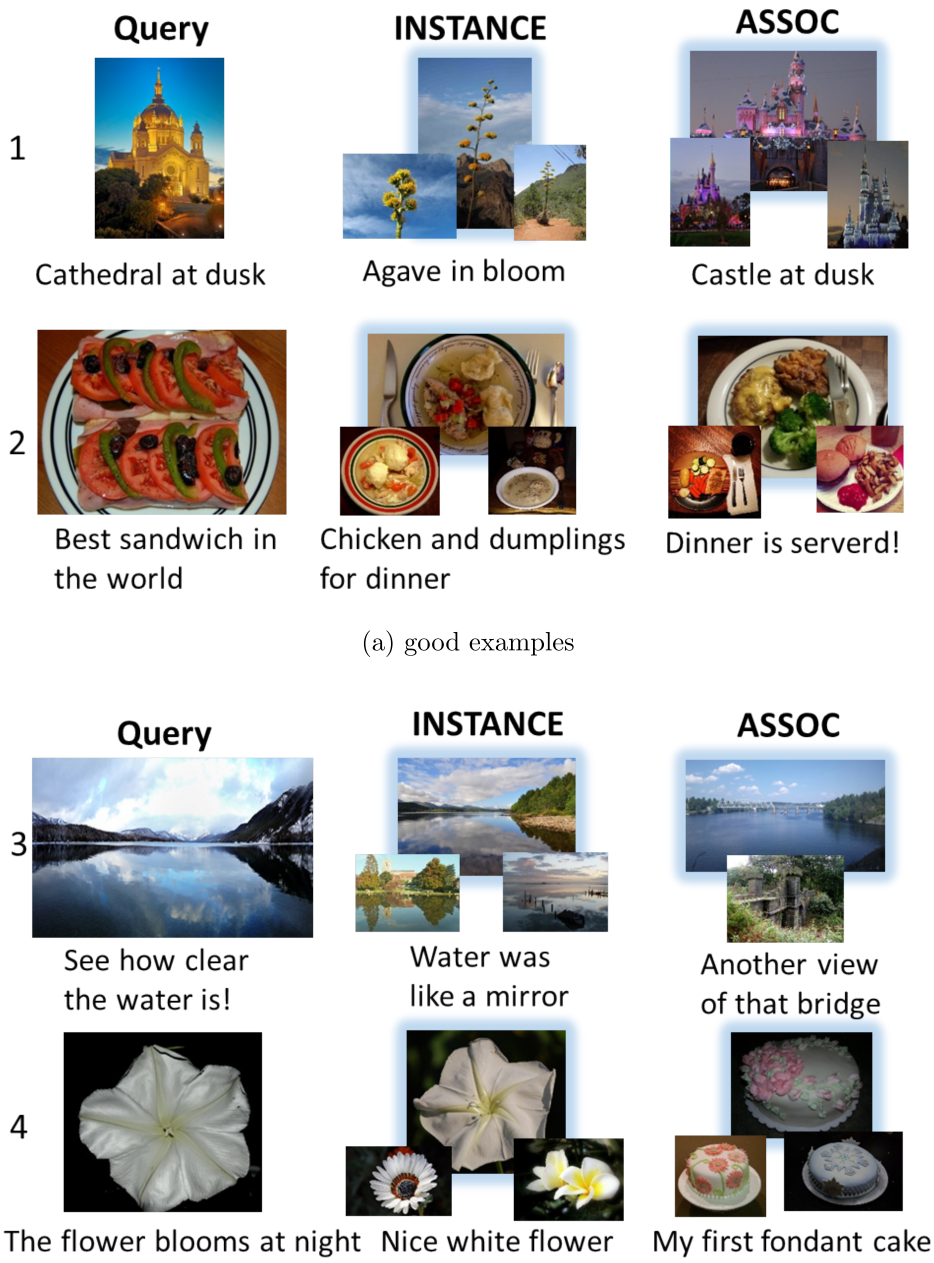
Example 4. When there is an *almost identical* image to the query image, the simple INSTANCE method works well. The ASSOC approach could be susceptible to the *visual diversity* of the neighborhood of images associated with candidate captions. In this example, the ASSOC approach lowers the rank of the good caption "*nice white flower*" because the visual content in its image cluster is diverse, so the average similarity of the query image to the top σ images are larger than that to the other caption "*My first fondant cake*".

3.5 Image captioning using visual paraphrases

We present an exploration of *visually situated paraphrase* (or *visual paraphrase* in short hand), and demonstrate their utility for image captioning. Formally, given our corpus $G = (T, V, E)$, a visual paraphrase relation is a triple (i, c, p), where given an image $i \in V$ and its original caption $c \in T$ (i.e., $e(c, i, original) \in E$), $p \in T$ is a visual paraphrase for c situated in a visual context given by the image i (i.e, $e(p, i, paraphrase) \in E$). We collect visual paraphrases using both human annotation and an automatic algorithm.

(1) Visual Paraphrasing using Crowd-sourcing: We use Amazon Mechanical Turk to annotate visual paraphrases for a subset of images in our corpus. Given each image with its original caption, we showed 10 randomly sampled candidate captions from our dataset that share at least one physical-object noun[7] with the original caption. Turkers

[7] under the WordNet "physical_entity.n.01" synset



(a) good examples

(b) bad examples

Figure 7: Good and bad examples. *Good (or bad) examples where the ASSOC approach retrieves captions of better (or worse) visual relevance than the INSTANCE approach. In each row, the left is the query image and its original caption; the middle and the right show the caption retrieved by the INSTANCE and the ASSOC method, respectively, shown with top similar images to the query image in its visual neighborhood. The INSTANCE approach looks at a single image (the largest image in the center with glow). The ASSOC method consider a collection of images associated with the captions.

choose all candidate captions that could also describe the given image. We collect 7,570 (i, c, p) paraphrase triples in total.

(2) Visual Paraphrasing using Associative Structure: We also propose an algorithm for automatic visual paraphrasing by adapting the ASSOC algorithm for image captioning (§3.3) as follows: given an image-caption pair (i, c), it first prepares a set of candidate captions that share the largest number of physical-object nouns with c, which are likely to be semantically close to c; then we rerank the candidate captions using the same neighborhood-based affinity as described in §3.3.

We apply this algorithm to generate a large set of visual paraphrases. For each caption in our corpus, we randomly sample two of its associated images, and generate one visual paraphrase for each image-caption pair, which yields 353,560 (i, c, p) triples. See Figure 4 for example paraphrases.

3.5.1 Image captioning using visual paraphrasing

We propose to utilize automatically-generated visual paraphrases to improve the ASSOC approach (§3.3) for image captioning. One potential limitation of the ASSOC approach is that for some captions, the number of associated images might be too small for reliable estimations of the neighborhood based affinity. We hypothesize that for a caption with a small visual neighborhood, merging its neighborhood with those associated with its *visual paraphrases* will give a more reliable estimation of the affinity between a query image and that caption. Thus we modify the ASSOC approach as follows.

After preparing a pool of K candidate captions $\{c_1, c_2, \ldots, c_K\}$, automatically generate a visual paraphrase (i_i, c_i, p_i) for each (i_i, c_i); then rerank the candidate captions by the following affinity function that merges the visual neighborhood from the paraphrase,

$$\mathcal{A}(q, C_i) = \phi(q, \mathcal{N}_{c_i} \cup \mathcal{N}_{p_i}) \tag{11}$$

method	BLEU	METEOR	AMT
INSTANCE	0.125	0.029	N/A
ASSOC$^{\mathsf{gi}}$	0.172	0.033	45%
ASSOC$^{\mathsf{gi}}_{\mathsf{para}}$	**0.187**	**0.036**	55%
ASSOC$^{\mathsf{ti}}$	0.184	0.034	45%
ASSOC$^{\mathsf{ti}}_{\mathsf{para}}$	**0.197**	**0.036**	55%

Table 12: Automatic and human evaluation of exploiting visual paraphrases for image captioning. The superscripts represent the image feature used in the reranking step; gi: GIST; ti: Tinyimage. The AMT column shows the percentages of captions preferred by human as of better visual relevance, in pairwise comparisons. The improvement of ASSOC$_{\mathsf{para}}$ over ASSOC is significant at $p < 0.002$ for BLEU, and $p < 0.03$ for METEOR with two tailed t-test.

3.6 Experiments: visual paraphrasing improves image captioning

The experimental configuration basically follows §3.4. We compare ASSOC$_{\mathsf{para}}$, the visual-paraphrase augmented approach, to the vanilla ASSOC approach. The image feature setting is the one with which the ASSOC approach performs best. Both approaches use the GIST+Tinyimage feature to prepare candidate captions, then use either the GIST or Tinyimage feature for reranking.

Table 12 shows that the ASSOC$_{\mathsf{para}}$ approach significantly improves the vanilla ASSOC method under both automatic and human evaluation. As a reference, the first row shows the performance of the INSTANCE method (§3.4). The ASSOC method significantly improves over the INSTANCE method. On a similar vein, the ASSOC$_{\mathsf{para}}$ method further improves over the ASSOC method, as automatic paraphrases provide a better visual neighborhood. This improvement is remarkable since the paraphrasing association is added automatically without any supervised training. This demonstrates the usefulness of the bipartite association structure of our corpus.

3.7 Image captioning with creativity

Naturally existing captions reflect everyday narratives, which in turn reflect figurative language use such as metaphor, simile, and personification. To gain better insights, one of the authors manually categorized a set of 1000 random captions. About 17% are identified as figurative. Table 7 shows the distribution of different types of figurative captions.

Creative Language Classifier: Using the small set of labels described above, we train a simple binary classifier to identify captions with creative language.[8] Using this classifier, we can control the degree of literalness or creativity in generated captions. Based on 5-fold cross-validation, the classifier performs with 77% precision and 43% recall.

Importantly, a high-precision and low-recall classifier suffices our purpose. It is because in the context of creative captioning and creative paraphrasing presented below, we only need to detect *some* figurative captions, not *all*.

3.7.1 Creative image captioning

Given a query image q, we describe it with the most appropriate figurative caption. We propose the $\textsc{assoc}_{\mathsf{creative}}$ approach that alters the $\textsc{assoc}$ approach (§3.3) to return a *figurative* caption from the candidate pool, excluding *literal* captions.

3.7.2 Creative visual paraphrasing

Given a query image q and its *original* caption c, we rephrase c to a more creative and inspirational caption that still describes q. We use the $\textsc{para}_{\mathsf{creative}}$ approach that changes our automatic visual paraphrasing algorithm (§3.5), by retrieving only figurative captions.

[8] We use a random forest classifier with features including words indicating reasoning (but, could, that), generality (never, always), caption length, abstract nouns (life, and hope), and whether the caption is a known idiom or quotation.

method	creativity	relevance
ASSOC	33%	41%
$\text{ASSOC}_{\text{creative}}$	67%	59%

Table 13: Human evaluation for creative captioning: % of captions preferred by judges in pairwise comparisons

3.8 Experiments: creative image captioning and paraphrasing

3.8.1 Creative captioning

We compare the $\text{ASSOC}_{\text{creative}}$ approach to the vanilla ASSOC approach. With the ASSOC approach, the top-rank caption is usually literal. Both approaches use the GIST+Tinyimage feature for preparing candidate captions, and the Tinyimage feature for reranking, which is the best setting for the ASSOC approach (§3.4).

Similarly to §3.4, we sample 200 test images from our corpus, and use AMT to compare two algorithms in terms of *visual relevance* and *creativity* separately. For creativity, we ask turkers to choose one of the two captions that is more creative and inspirational than the other to describe each given test image. Results are shown in Table 13.

(1) *Creativity.* For 2/3 of the query images, captions produced by the $\text{ASSOC}_{\text{creative}}$ method are judged as more creative than those produced by the ASSOC method. This result indirectly validates that the figurativeness classifier has a reasonable precision to control the literalness of the system caption.

(2) *Visual relevance.* Interestingly, not only the captions from the $\text{ASSOC}_{\text{creative}}$ method are favored as creative, they are also judged as visually more relevant than those from the ASSOC method, despite that each figurative caption has lower neighborhood-based affinity than the literal counterpart. We conjecture that it is easier for human judges to be imaginative and draw visual relevance between the query image and figurative captions than the literal counterparts. This result also suggests that figurative language may be of practical use in image caption applications as a means to smooth the potentially brittle system output. Figure 8 shows example system output.

3.8.2 Creative visual paraphrasing

We test 200 images that are associated with *literal* captions as predicted by the figurativeness classifier. The PARA$_{\text{creative}}$ approach competes against two baselines: 1) the ORIGINAL captions , and 2) a text-only variant of the PARA$_{\text{caption}}$ approach sans visual processing: it randomly chooses a figurative caption that shares the largest number of physical-object nouns with the original caption, without looking at the query image. This is for evaluating the effect of visual context.

In addition to the evaluations as in §3.8.1, we also use a *multiple-choice* setting that allows a turker to choose zero to two captions that are visually relevant to the query image. See Table 14 for results, and Figure 8 for example outputs.

I. Comparing original captions with creative paraphrases (Original vs. para$_{\text{creative}}$): The paraphrases are preferred over the original literal captions as more creative most of the time. As for the visual relevance, the original captions are favored over the paraphrases most of the time in the single-choice competition. However, when we use a multiple-choice setting, paraphrases has a reasonable relevance rate (60%), despite the simplicity of the algorithm. The fact that the original captions has a high relevance rate (87%) shows that in our corpus the captions have high visual relevance to their associated images most of the time.

II. Creative paraphrasing with and without the visual context (para$_{\text{caption}}$ vs. para$_{\text{creative}}$): In terms of creativity, the PARA$_{\text{caption}}$ method is preferred over the PARA$_{\text{creative}}$ method. We conjecture that without conditioning on the visual content, PARA$_{\text{caption}}$ method tends to retrieve more unexpected captions that make turkers think they are more fun and creative. As for the visual relevance, by conditioning on the visual context given by query images, the PARA$_{\text{creative}}$ method significantly improves the visual relevance over the text-only counterpart, PARA$_{\text{caption}}$ method. This result highlights the pragmatic differences between visually-situated paraphrasing and text-based paraphrasing.

method	creativity	relevance	
		single	*multiple*
ORIGINAL	32%	80%	87%
PARA$_{\text{creative}}$	68%	20%	60%
PARA$_{\text{caption}}$	56%	47%	63%
PARA$_{\text{creative}}$	44%	53%	74%

Table 14: Human evaluation for creative visual paraphrasing

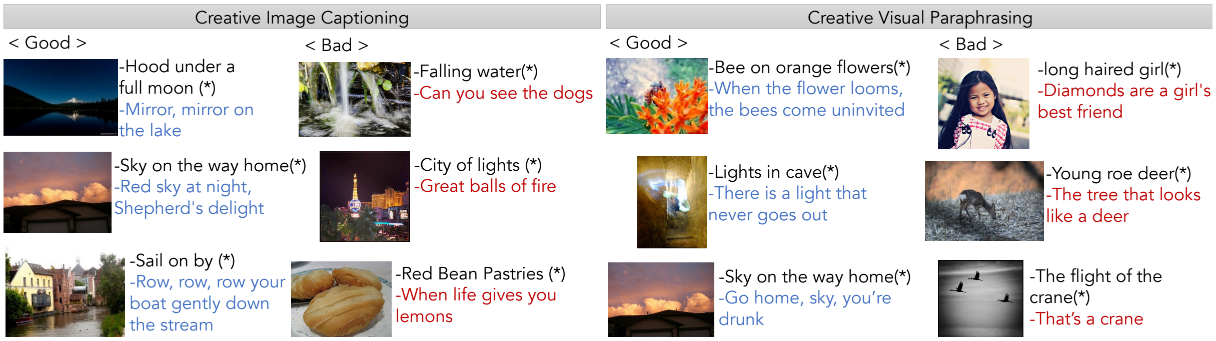

Creative Image Captioning
Creative Visual Paraphrasing
< Good >
< Bad >
< Good >
< Bad >
-Hood under a full moon (*)
-Mirror, mirror on the lake
-Falling water(*)
-Can you see the dogs
-Bee on orange flowers(*)
-When the flower looms, the bees come uninvited
-long haired girl(*)
-Diamonds are a girl's best friend
-Sky on the way home(*)
-Red sky at night, Shepherd's delight
-City of lights (*)
-Great balls of fire
-Lights in cave(*)
-There is a light that never goes out
-Young roe deer(*)
-The tree that looks like a deer
-Sail on by (*)
-Row, row, row your boat gently down the stream
-Red Bean Pastries (*)
-When life gives you lemons
-Sky on the way home(*)
-Go home, sky, you're drunk
-The flight of the crane(*)
-That's a crane

Figure 8: Examples of creative captioning and creative visual paraphrasing. The left column shows good examples in blue, and the right column shows bad examples in red. The captions marked with * are the original captions of the corresponding query images.

3.9 Related works

Image-caption corpus: Our work contributes to the line of research that makes use of internet web imagery and text (Ordonez et al., 2011; Berg et al., 2010) by detecting the visually relevant text (Dodge et al., 2012) and reducing the noise (Kuznetsova et al., 2013b; Kuznetsova et al., 2014). Compared to datasets with crowd-sourced captions (Hodosh et al., 2013; Lin et al., 2014), in which each image is annotated with several captions, our dataset presents several images for each caption, a subset of which also includes visually situated paraphrases. The association structure of our dataset is analogous to that of ImageNet (Deng et al., 2009). Unlike ImageNet that is built for nouns (physical objects) listed under WordNet (Miller, 1995), our corpus is built for expressive phrases and full sentences and constructed without human curation. Our corpus has several unique properties to complement existing corpora. As explored in a very recent work of (Gong et al., 2014), we expect that it is possible to combine crowd-sourced and web-harvested datasets and achieve the best of both worlds.

Image captioning: Our work contributes to the increasing body of research on retrieval-based image captioning (Ordonez et al., 2011; Hodosh et al., 2013; Hodosh and Hockenmaier, 2013; Socher et al., 2014), by providing a new large-scale corpus with unique association structure between images and captions, by proposing an algorithm that exploits the structure, and by exploring two new dimensions: (i) visually situated paraphrasing (and its utility for retrieval-based image captioning), and (ii) creative image captioning.

Paraphrasing: Most previous studies in paraphrasing have focused exclusively on text, and the primary goal has been learning *semantic* equivalence of phrases that would be true out of context (e.g., (Barzilay and McKeown, 2001; Pang et al., 2003; Dolan et al., 2004; Ganitkevitch et al., 2013)), rather than targeting *situated* or *pragmatic* equivalence given a context. Emerging efforts began exploring paraphrases that are situated in video content (Chen and Dolan, 2011), news events (Zhang and Weld, 2013), and knowledge base (Berant and Liang, 2014). Our work is the first to introduce *visually situated paraphrasing* in which the task is to find paraphrases that are conditioned on both the input text as well as the visual context. (Chen and Dolan, 2011) collected situated paraphrases only through crowd sourcing, while we also explore automatic collection, and further test the quality of automatic paraphrases by using the learned paraphrases in an extrinsic evaluation setting.

Figurative language: There has been substantial work for detecting and interpreting figurative language (Shutova, 2010; Li et al., 2013; Kuznetsova et al., 2013a; Tsvetkov et al., 2014), while relatively less work on *generating* creative or figurative language (Veale, 2011; Ozbal and Strapparava, 2012). We probe data-driven approaches to creative language generation in the context of image captioning.

3.10 Summary

To conclude, we have provided insights into making a better use of multimodal web data in the wild, resulting in a large-scale corpus, *Déjà Image-Captions*, with several unique properties to complement datasets with crowdsourced captions. To validate the usefulness of the corpus, we proposed new image captioning algorithms using the associative structure, which we extended to several related tasks ranging from visually situated paraphrasing to enhanced image captioning. In the process we have also explored several new tasks: visually situated paraphrasing, creative image captioning, and creative caption paraphrasing.

Chapter 4

Multimodal Temporal Knowledge Modelling with Photo Albums

4.1 Overview

Chapter 3 revealed and utilized the bipartite cross-modal association structure hidden in the ocean of online image-caption pairs. This chapter further leverages the temporal structure innate in online photo albums recording common scenarios, to understand both images and text in context.

Activities and events in our lives are procedural, be it the process of sausage making, or a trip to camping, or a ceremony of tying the knot. Many of them exhibit common temporal-spatial patterns. For example, a wedding ceremony typically consists of a sequence of events such as walking down the aisle, exchanging vows, cutting the cake, and dancing. In addition, it is typical to see people in formal attires, toasting champagne, and playing a violin, while less likely to see people reading books or chopping trees.

This observation of structural patterns in common events was at the heart of early AI research. Scripts (Schank and Abelson, 1977), one of the earliest representation formalisms, were developed to encode the knowledge of common events to support an inference engine. However, purely symbolic approaches, as extensively pursued in 70s and early 80s, required hand-coded representation of knowledge, which turned out to be too brittle when used for reasoning and prohibitively difficult to scale.

In recent years, however, there has been emerging research to learn scripts-like knowledge statistically from large-scale unstructured natural language corpora; *Narrative schema* (Chambers and Jurafsky, 2009) discovers the common sequence of verbs that describe events such as book publishing and lawsuits, while (McIntyre and Lapata, 2009) learns the typical story lines and plot structure from children's stories.

In parallel, another line of research actively pursued in recent years is large scale grounding of natural language text with web imagery, e.g., learning the typical textual descriptions given a situation captured in an image (Ordonez et al., 2011; Ordonez et al., 2013; Mason and Charniak, 2014; Kuznetsova et al., 2014), projecting multimodal signals into a common semantic representation (Hodosh et al., 2013; Socher et al., 2014), and aligning part-based semantic correspondences between images and their corresponding textual descriptions (Kuznetsova et al., 2012; Karpathy et al., 2014).

Drawing inspirations from both these avenues of research, in this chapter, we present the first study to learn multimodal knowledge of common events from a large collection of photo albums. A unique aspect of our work, compared to most prior efforts connecting language and vision, is the temporal dimension. A photo album comprises of a sequence of images, each with a time stamp and a corresponding caption. User contributed photo albums, abundantly available at online communities, provide new opportunities to learn procedural knowledge of common events that people experience and record.

Compared to online videos, as studied in recent work for generating short video descriptions (Venugopalan et al., 2015), photo albums as sequences of images have a few advantages. They span over much longer temporal spans (e.g., a camping trip over a few days), accompany noisy but rich textual annotations as provided by online users, and are significantly more manageable in terms of data storage and processing. In this study we have compiled and organized 34,818 albums over 12 common scenarios such as ceremonies and travels (see Figure 9). The resulting dataset includes nearly 1.5 million pairs of images and captions. We share the dataset publicly at www.cs.stonybrook.edu/~jianchen/albums.

We formulate the unsupervised learning of event structure as a sequential clustering problem. Specifically, we aim to find a sequence of sub-events characterized by groups of images and captions. Using the learned multimodal event model (§4.3), we propose a collective inference algorithm based on Integer Linear Programming (ILP) that infers the events

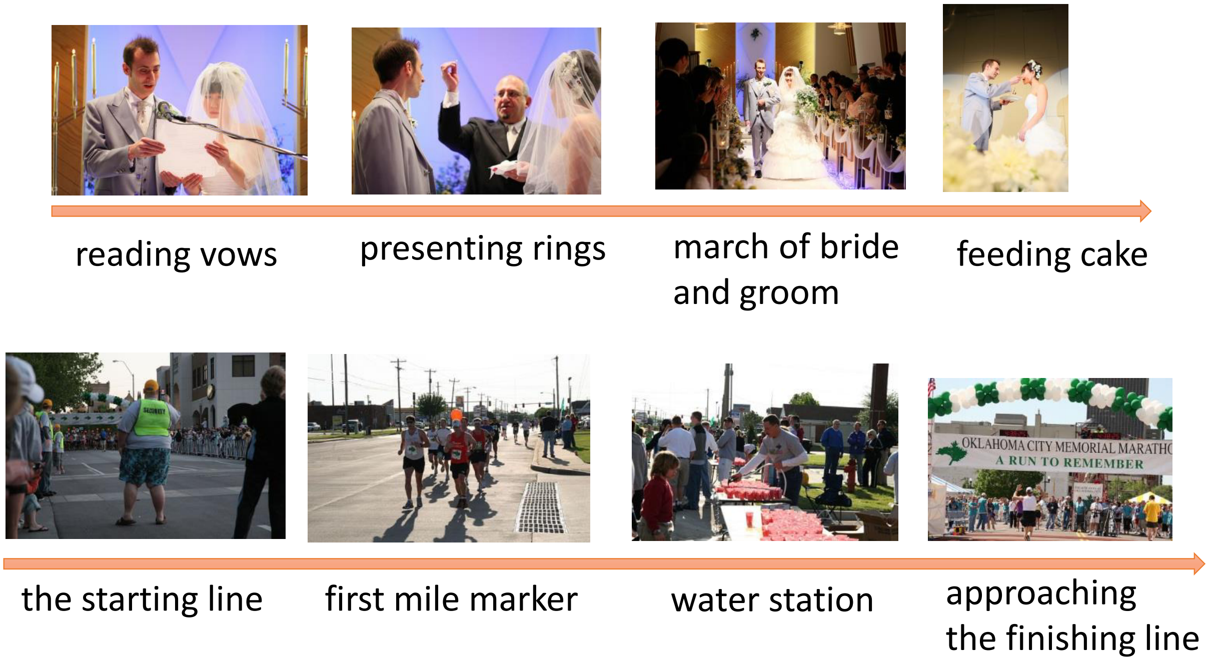


Figure 9: The part of two example albums in the wedding (top) and marathon scenarios (bottom), respectively.

of each photo in a given album (§4.4). We then evaluate the quality and the usefulness of the learned knowledge via two tasks (§4.5): (1) *multimodal event segmentation* that partition a given album into coherent segments, and (2) *multimodal album summarization* that selects a few representative images and caption them to highlight the major events in a given album. Our experiments demonstrate that our approaches based on multimodal event model have a better understanding of image sequences and identifies more representative photos in summaries than competitive baselines; and the collective inference algorithm helps to better identify coherent event segments in an album.

4.2 Dataset

We compiled a new dataset that consists of about 1.5M image-caption pairs organized into 35K albums. The album size ranges from 10 to over 1000 photos.

Collection We use the Flickr API to collect images at flickr.com across 12 common scenarios in which people frequently participate and take photos in their daily life, e.g.,

Scenario	# of albums	# of images
Wedding	4689	192374
Camping	4063	158869
Paris Trip	4603	306171
New York Trip	4205	267677
Independence Day	548	22053
Funeral	781	28182
Thanksgiving	5928	152514
Barbecue	735	21661
Marathon	3961	157813
Christmas	3449	97575
Cooking	1168	36369
Baby Birth	688	20738

Table 15: Statistics of the Dataset

weddings, barbecues, and camping trips (see Table 15). We search for these images using a scenario name and its variations (e.g. Paris Trip, Paris Travel, Paris Vacation). For scenarios with large quantities of images, we limit our search to the past 3 years. For scenarios with less, we search images in all years. We collect an image only if the scenario name we're looking for is included in the image's title, or its description. We then generate a caption of the image by concatenating its title and the first sentence of its description. In addition, we store the timestamp of every photo.

For each scenario, we assemble albums by sorting image-caption pairs by user and timestamp. We sequentially scan images over a certain period of time for a given user to form albums. For example, for wedding scenario, we regard consecutive photos of the same user up to 24 hours as an album; for travelling scenario, the time span of a single album is up to 5 days.

Filtering Many online albums have few informative captions, since most flickr users are in general lazy to caption every photo. For each scenario, we first keep up to top 10K albums with largest number of unique captions. Then clean the titles and descriptions (e.g., removing non-ascii characters, detecing and removing automatically-generated captions and advertisement captions).

By now, some albums in a given scenario, say wedding, might have many unique captions, but the album does not actually capture about a typical wedding; it's crawled only because one or two caption happen to mention the word "wedding". To further filter albums with high relevance to a given scenario, we count unique *topic words* for each album that are highly indicative of a given scenario. Example topic words for the wedding scenario include *bride, groom, ring, flower, vow, and so on.* We use heuristics based on the topic word count to filter albums of high relevance to a given scenario, for example, we discard albums that have less than *thr* topic words (we set *thr* to 7) and hence are less likely to be relevant to a given scenario. To be precise, the topic words of a scenario are defined as the top 200 most discriminative words in a given scenario. The descriminativeness score of a word w in scenario S is defined as the posterior probability $P(S|w)$ using Bayes formula,

$$P(S|w) = \frac{P(s,w)}{P(w)} = \frac{P(w|S)P(S)}{\sum_{S'} P(w|S')P(S')} \tag{12}$$

where

$$P(w|S) = \frac{|\{s_j|w \ appears \ in \ s_j, \ s_j \in S\}|}{|S|} \tag{13}$$

s_j is the jth album in scenario S.

4.3 Event modelling

The overarching goal of this chapter is to learn statistical knowledge about record-worthy common events that people experience and share through online photo albums.

Events are inherently hierarchical. In this work, we assume a simple two-level structure where the higher-level event is given (e.g., wedding, camping, funeral), and the goal is to learn the lower-level events by clustering images and captions based on their content similarities as well as their sequential regularities. Hereafter, we refer to the higher-level events as *scenarios*, and lower-level events as *sub-events* or simply as *events*.

4.3.1 Representation

Given a set of albums that belong to the same scenario (e.g., wedding), we want to learn a set of prototypical *events* and their *temporal ordering* in that scenario. For example, the prototypical events in a wedding includes vows, ring exchange, reception, and dancing. And vows usually happens early in the wedding scenario, while dancing usually happens

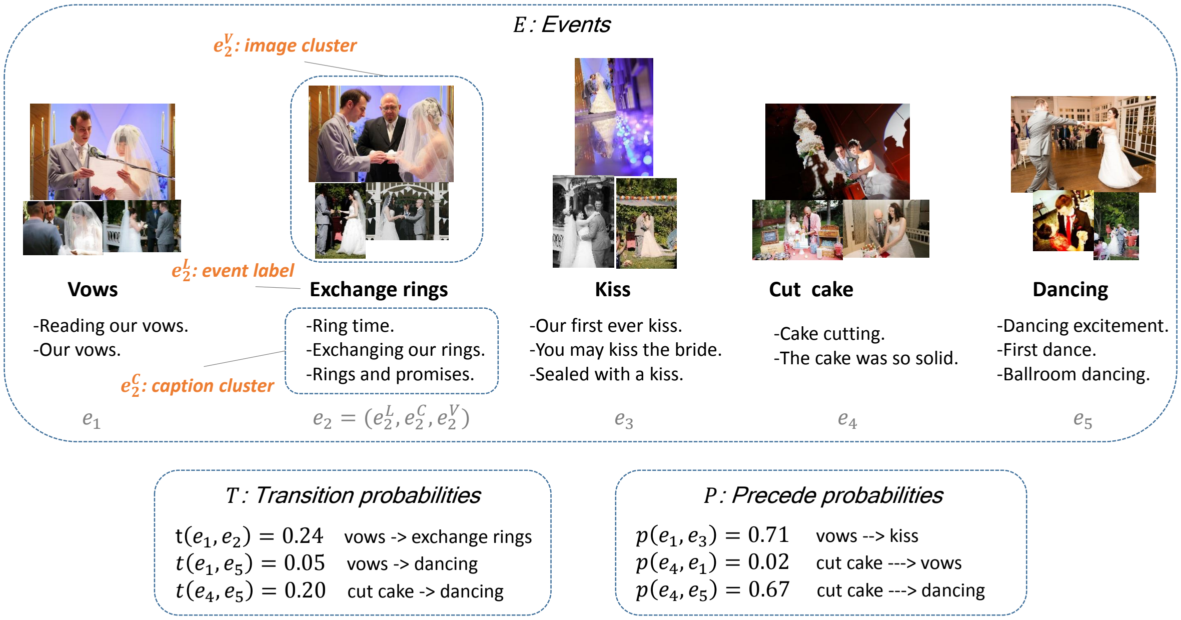


Figure 10: The multimodal event model.

later. Such script-like event model encodes background knowledge of the context of a scenario. We will demonstrate the event model helps better understand images and captions in sequence in three tasks (§4.5).

Specifically, we model events in a given scenario as a triple $\mathcal{M} = (E, T, P)$ of three sets, where E is a multimodal, non-parametric representation of the events, T and P represents the probabilistic temporal relations of those events (see Figure 10).

(1) E is a set of **events** $\{e_i\}$, in which each event in turn is a triple $e_i = (e_i^N, e_i^C, e_i^V)$, where the event label e_i^L is a short and prototypical expression of that event (e.g., exchange rings), e_i^C is a set of alternative captions describing e_i (e.g., ring time; exchanging our rings), and e_i^V a set of images that are associated with captions in e_i^C.

(2) T is a set of **transition probabilities** $\{t(e_i, e_j)\}$, where $t(e_i, e_j)$ is the probability that the successive event of e_i is e_j.

(3) P is a set of **precede probabilities** $\{p(e_i, e_j)\}$, where $p(e_i, e_j)$ is the probability that event e_i happens before e_j (there could be other events in between).

4.3.2 Learning

Given a set of albums in a specific scenario, we first identify prototypical events using the K-means clustering algorithm, then compute the transition and precede probabilities between event pairs using observed empirical counts.

Identifying Events Given a scenario, we collect all captions across all albums in that scenario and cluster them using the K-means algorithm[1] . Each caption is represented by the unigram feature of its content words (nouns, verbs, adjectives, and adverbs) weighted by its discriminative score (Eq 12). We use 40 cluster centers, and perform the K-means clustering for 300 iterations with 10 random initializations, then choose the one that obtains the lowest inertia across the entire sample.

From each caption cluster, we learn a prototypical event $e_i = (e_i^L, e_i^C, e_i^V)$, where e_i^C is the caption cluster, and e_i^V is the image cluster associated with e_i^C. We extract the event label e_i^L as the most frequent words that appear in more than 80% of the captions in that cluster. If no such words exist, we use the single most frequent word.

Note that the largest cluster in a scenario is usually a special one we call *miscellaneous cluster*. Since the captions in it does not represent a coherent event as do the remaining clusters.

Estimating temporal relations The above clustering gives each caption in every album an event label corresponding to its cluster assignment. We estimate the transition ($t(e_i, e_j)$'s) and precede ($p(e_i, e_j)$'s) probabilities between event pairs using observed empirical counts as follows,

$$t(e_i, e_j) = \frac{\#(e_i \to e_j)}{\#(e_i \to \cdot)} \tag{14}$$

where $\#(e_i \to e_j)$ is the number of *event transition pairs* $(e_i \to e_j)$ across all albums, and $\#(e_i \to \cdot)$ is the sum of number of event transition pairs starting with e_i. An event transition pair $(e_i \to e_j)$ corresponds to two consecutive captions labelled with event e_i followed by e_j. We count each event transition pair at most once per album.

[1] In each album, we collect only unique captions.

$$p(e_i, e_j) = \frac{\#(e_i \rightarrowtail e_j)}{\#(e_i \rightarrowtail e_j) + \#(e_j \rightarrowtail e_i)} \tag{15}$$

where $\#(e_i \rightarrowtail e_j)$ is the number of *event preceding pairs* $(e_i \rightarrowtail e_j)$ across all albums. An event preceding pair $(e_i \rightarrowtail e_j)$ corresponds to a pair of captions where one caption labelled with event e_i precedes another caption labelled with event e_j (there might be other captions in between). We count each event preceding pair at most once per album.

Both estimations remove all miscellaneous events when considering transition and preceding relations. We set transition and precede probabilities to zero for event pairs that do not co-occur in any albums. For transition probabilities, we add two special events to each album, representing the starting and ending of that album, respectively.

4.4 Event inference

We use event model to better understand images and captions in sequence. Given an album of images with or without captions, we infer the event of each image in that album. We jointly infer the events of all images in an album using Integer Linear Programming (ILP). The ILP formulation aims to find an event assignment that has both high individual affinity of each photo to its assigned event, and the mostly likely temporal ordering of events (see Figure 11).

4.4.1 Notation

Suppose we are given a sequence of m photos $P = \{p_1, \ldots, p_m\}$ in a known scenario with an event model $\mathcal{M} = (E, T, P)$, where each photo is an (image, caption) pair $p_i = (p_i^I, p_i^C)$. Note that we will study event inference with two kinds of input: (1) the input album has both the images p_i^I's and the original captions p_i^C's; (2) the input album has only images but no captions. There are n events in $E = \{e_1, \ldots, e_n\}$. We assign each photo to a single event. The decision variable $b_{i\alpha}$ indicates whether photo p_i is assigned to event e_α.

To configure global event transitions, we think with segments. A segment is a sequence of consecutive photos that belong to the same event. Two adjacent segments belong to different events.

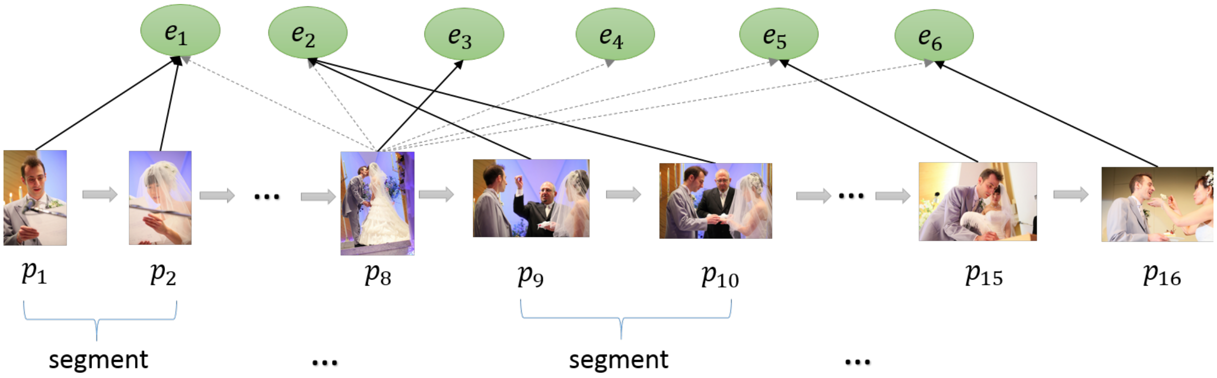


Figure 11: Event inference using Integer Linear Programming. Given an album, we assign each photo to a single event in a given event model. A boolean variable $b_{i\alpha}$ represents whether photo p_i is assigned to event α, which is shown as an arrow between p_i and e_α (a solid arrow represents true and dotted one false; only one solid arrow is connected with each photo). The consecutive photos that belong to the same event form a *segment*. There is an *event transition* between two consecutive photos that belong to two different segments (e.g., there is an event transition from event e_3 at photo p_8 to event e_2 at photo p_9).

We use $i, j, k, \ldots \in \{1, \ldots, m\}$ to index photos, $\alpha, \beta, \gamma, \ldots \in \{1, \ldots n\}$ to index events, $s, t \in \{0, 1, \ldots, q, q{+}1\}$ to index segments (where q upper bounds the number of segments), function-like notation $\mathsf{NewVar}(\mathsf{args})$ to define a new variable based on existing variables (see sec.4.4.5 for an explanation).

4.4.2 Objective function

The event inference maximizes a sum of four components,

$$\begin{aligned} obj \;=\; & \sum_{i=1}^{m}\sum_{\alpha=1}^{n} A_{i\alpha}b_{i\alpha} + \sum_{i=1}^{m-1} \mathsf{isim_i SameEvent(i, i+1)} + \sum_{i=1}^{m-1} \mathsf{csim_i SameEvent(i, i+1)} + \\ & \sum_{s\geq 0}^{q}\sum_{\alpha=1}^{n}\sum_{\beta=1}^{n} \mathsf{tp}_{\alpha\beta}\mathsf{Transit(s}, \alpha, \beta) \end{aligned}$$

(1) **Photo-Event-Affinity (EA)**, the sum of the photo-event affinity $A(p, e)$, the affinity of each individual photo p to its assigned event e. $A_{i\alpha}$ is the affinity between the photo

$p_i = (p_i^I, p_i^C)$ and the event e_α (see §4.4.3 for the constraints),

- If the input album has *both* images *and* the original captions, let $A_{i\alpha}$ be the **caption-event affinity** $A_C(p_i^C, e_\alpha)$, which is the cosine similarity between the caption p_i^C and the center of the caption cluster e_α^C associated with e_α. We use only captions but not images to compute $A_{i\alpha}$ because textual signal is more robust.

$$A_{i\alpha} = A_C(p_i^C, e_\alpha) = \frac{f(p_i^C) \cdot f_\alpha}{\|f(p_i^C)\| \|f_\alpha\|} \tag{16}$$

where $f(p_i^C)$ is the feature of the caption p_i^C, and f_α is the feature of the center of e_α^C. Our experiments use the same caption feature as in §4.3.2.

- If the input album has *only images*, let $A_{i\alpha}$ be the **image-event affinity** $A_I(p_i^I, e_\alpha)$, which is the visual neighbourhood based affinity between the image p_i^I and the image cluster e_α^V associated with e_α,

$$A_{i\alpha} = A_I(p_i^I, e_\alpha) = \mathcal{A}(p_i^I, e_\alpha^V) \tag{17}$$

which is the average cosine similarity between the image of p_i and its top $\sigma = 15$ most similar images in e_α^V (see Eq.10). Our experiments represent images using the CNN features as in (Karpathy et al., 2014).

(2) **Image Similarity (IS)**, a term to reward similar consecutive images having the same event assignment. The coefficient $\mathsf{isim_i}$ is the cosine similarity between p_i^I and p_{i+1}^I. A boolean variable $\mathsf{SameEvent(i, i+1)}$ indicates whether p_i and p_{i+1} belongs to the same event. It is defined based on some other boolean variables. See §4.4.3 for details and related constraints.

(3) **Caption Similarity (CS)**, a term to reward similar consecutive captions having the same event assignment. The coefficient $\mathsf{csim_i}$ is the cosine similarity between p_i^C and p_{i+1}^C.

(4) **Transition Probability (TP)**, a term to reward likely transitions between two consecutive *segments* according to the learned event model $\mathcal{M}$. The coefficient $\mathsf{tp}_{\alpha\beta} = t(e_\alpha, e_\beta)$ is the transition probability between the event α and the event β. A boolean variable $\mathsf{Transit(s}, \alpha, \beta)$ indicates whether there is an event transition from e_α at segment

s to e_β at segment $s+1$. See §4.4.3 for details and related constraints. We use $s = 0$ and $s = q + 1$ to index the virtual starting and ending segments added to each album, respectively.

4.4.3 Constraints and additional boolean variables

We add the following linear constraints to ensure the validity of event assignment and the intended semantics of all boolean variables.

(1) Each photo belongs to one and only one event,

$$\forall i, \sum_{\alpha} b_{i\alpha} = 1$$

(2) The boolean variable $\mathsf{SameEvent(i,j)}$ represents whether photos p_i and p_j are assigned to the same event, which is defined as follows,

$$\mathsf{SameEvent(i,j)} = \sum_{\alpha} \delta_{ij\alpha}$$

where the boolean variable $\delta_{ij\alpha} = b_{i\alpha} \wedge b_{j\alpha}$ indicates whether photos p_i and p_j are both assigned to event α (see §4.4.5 for how to construct $\delta_{ij\alpha}$ using linear constraints). Then $\mathsf{SameEvent(i,j)} \in \{0, 1\}$, since each photo has to be assigned to one and only one event, $\delta_{ij\alpha}$'s for all events α's are either all 0's, or all 0 except a single 1.

(3) The boolean variable c_{is} indicates whether photo p_i belongs to the segment $s \in \{1, \ldots q\}$. Each photo belongs to one and only one segment,

$$\forall i \sum_{s=1}^{q} c_{is} = 1$$

(4) The segment index s starts from 1 and increases by 1 with each event transition, which is guaranteed by the following constraints,

Base case: the first photo has segment index 1,

$$c_{11} = 1$$

Continue Segment: If two consecutive photos have the same event, then they also have the same segment,

$$\forall i < m, s,\ c_{i+1,s} \geq \mathsf{SameEvent(i, i+1)} + c_{is} - 1$$

End Segment: If two consecutive photos have different events, then the segment index of the latter photo increases by 1,

$$\forall i < m, s,\ c_{i+1,s+1} \geq c_{is} - \mathsf{SameEvent(i, i+1)}$$

We define $c_{i,q+1} = 0$ for all i to handle the second to the last photo correctly.

(5) The boolean variable $e_{s\alpha}$ indicates whether the segment s belongs to the event e_{α}. The following constraints specify the relation among the *segment-level* variables $e_{s\alpha}$'s, c_{is}'s, and the *photo-level* variables $b_{i\alpha}$.

The first photo belongs to the first segment,

$$\forall \alpha,\ e_{1\alpha} = b_{1\alpha}$$

If a photo p_i is assigned to the segment s and the event α, then the segment s belongs to the event α,

$$\forall s, \alpha, i \geq s,\ c_{is} + b_{i\alpha} - 1 \leq e_{s\alpha}$$

where $i \geq s$ because each segment has at least one photo, and segment s has to start at least from photo s, that is, $\forall i < s,\ c_{is} = 0$.

Otherwise, the following constraints force $e_{s\alpha}$ to be 0,

Each segment is assigned to at most one event (not all of the q segments are used),

$$\forall s,\ \sum_{\alpha} e_{s\alpha} \leq 1$$

If no photo is assigned to segment s (segment s is not used), then all $e_{s\alpha}$'s are 0,

$$\forall s,\ \sum_{\alpha} e_{s\alpha} \leq \sum_{i} c_{is}$$

(5) To avoid too many scattering small segments that belong to the same event, we allow each event appear in at most thr segments,

$$\forall \alpha,\ \sum_{s} e_{s\alpha} \leq \mathsf{thr}$$

4.4.4 Approximation

The above ILP formulation yields tens of millions of constraints for an album with about 300 photos, which incurs very slow inference. To speed up inference, we merge consecutive photos with similar images and close timestamps as super nodes according to a heuristic, as they are likely to belong to the same event. We then run the ILP inference with super nodes whose number is much fewer than individual photos.

4.4.5 Defining a boolean variable

We use the following constraints to guarantee a new boolean variable $c = a \wedge b$ as the conjunction of two existing variables (a and b),

$$c \leq a$$

$$c \leq b$$

$$c + (1 - a) + (1 - b) \geq 1$$

4.5 Experiments

We evaluate our event inference algorithm (§4.4) based on event modelling in two tasks: album segmentation and album summarization.

4.5.1 Segmentation

Our first experiment tested how well we managed to segment the photos in the albums into coherent events, which is a foundation for photo sequence understanding and album summarization.

Data and annotation. We had an impartial annotator label where they thought events began and ended in the album and tested how well our model could replicate these boundaries. The annotator labels 10 albums for each of the three scenarios: wedding, camping, Paris trip. Each album is relatively long with about 100 to 150 photos.

Performance evaluation. We evaluate performance using 4 metrics: precision, recall, F1, and the event number difference between our models and the annotations, *d*. We base precision, recall, and F1 scores on the model's ability to recover the start of an event.

Method	Precision	Recall	F1	*d*
CLUSTER	.189	**.598**	.270	32.8
ILP$_{\text{EA+TP}}$	.509	.420	.425	4.9
ILP$_{\text{EA+TP+CS}}$	.434	.379	.379	3.8
ILP$_{\text{EA+TP+CS+IS}}$	**.484**	.449	**.438**	**2.9**

Table 16: Segmentation performance. F1 is computed based on the segment starting points. The column *d* shows the average difference between the number of segments of the annotation and that of the corresponding algorithm.

Methods. We compare the independent event assignment (CLUSTER) to ILP event inference with objective ablation. Each input album has both images and original captions. CLUSTER assigns event to each photo independently to maximize each individual photo-event affinity (§4.4.2). The remaining three methods correspond to the ILP event inference with different objectives denoted by the subscripts. Each objective is a sum of 2 to 4 terms separated by + (see §4.4.2 for the meaning of each two-letter acronym).

Results. Table 16 shows that the ILP inference with different objective functions all achieves a higher F1 score than CLUSTER. Furthermore, the full ILP formulation (ILP$_{\text{EA+TP+CS+IS}}$) gives the best precision/recall balance and also the lowest *d* value. While

CLUSTER does have a better recall, we attribute this performance to the sheer number of fragmented events that it identifies (32.8 more than annotated on average).

4.5.2 Multimodal album summarization

People share many long albums with hundreds of photos in social media websites. We propose the task of multimodal album summarization: Given a photo album, we pick a few representative photos and give each of them a caption, which highlights the major events in the album and tell a story over time.

Summarization based on event inference We explore summarization with two kinds of input: (1) The input album has *both* images *and* original captions, and the a summarization algorithm selects images and use their *original* captions in the summary; (2) The input album has *only images*, and a summarization algorithm *generates* a caption for each selected image. The latter is a much more challenging task than the former, since it tries to directly understand visual content.

Given a budget B and a photo album with or without captions, we first use the ILP inference (§4.4) to assign an event to each photo. Then choose the top B most important unique events as the events with largest (image or caption) cluster sizes (excluding the miscellaneous event that always has the largest cluster). The cluster size approximates the importance of an event. For each chosen event e, we select a single photo p that maximizes the photo-event affinity $A(p, e)$ to e among all photos labelled with e. The photo-event affinity is computed similarly to §4.4.2 depending on whether captions are in the input,

i. If the input album has both images and captions, the photo-event affinity $A(p, e)$ is a weighted sum of the image-event affinity (Eq.17) and the caption-event affinity (Eq.16),

$$A(p, e) = w \cdot A_I(p^I, e) + (1 - w) \cdot A_C(p^C, e)$$

We set $w = 2/3$ in our experiments to make both terms in about the same range. Since the range of $A_I(p^I, e)$ is about $[0, 0.5]$, and the range of $A_C(p^C, e)$ is about $[0, 1]$ that is about twice as large as the former. See Figure 12 for example summaries.

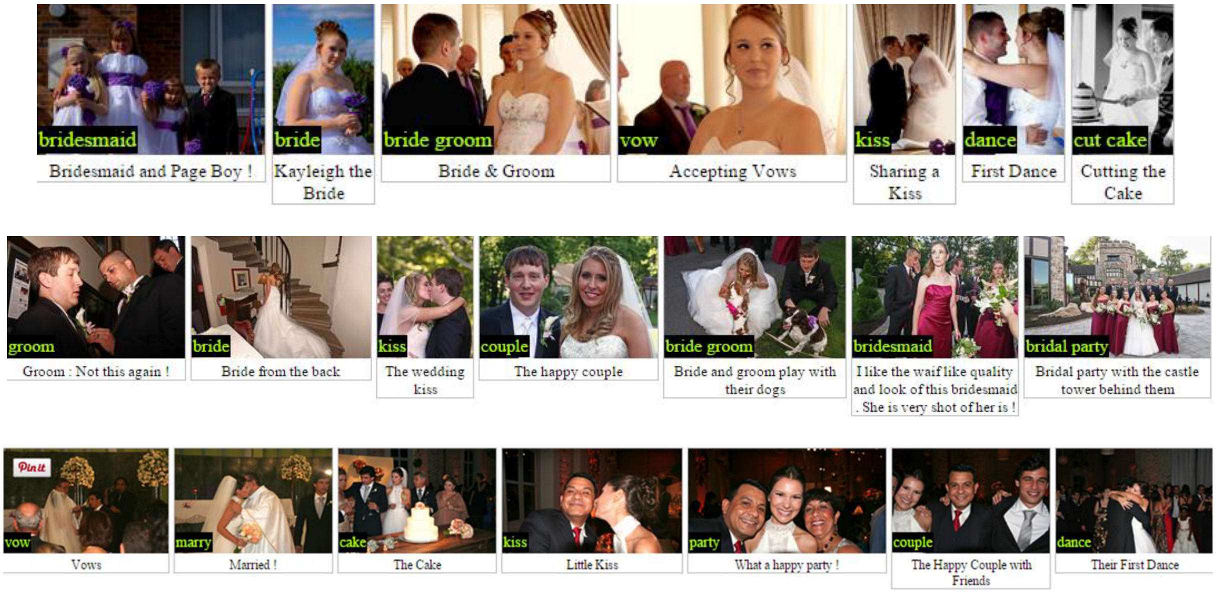


Figure 12: Example summaries generated by the ILP event inference (with *both* the images *and* their original captions as the input). Each row shows a summary of one album. The green tag at the bottom-right of each image shows the label of the inferred event of that image.

ii. If the input album has only images but no captions, the photo-event affinity $A(p, e)$ is set to the image-event affinity (Eq.17),

$$A(p, e) = A_I(p^I, e)$$

After selecting the photo p for a chosen event $e = (e^L, e^C, e^I)$, we transfer the caption of p's most similar image in the image cluster e^I (in terms of cosine similarity) as the caption of p. See Figure 13 for example summaries.

Experimental settings We randomly select 100 long albums in the wedding scenario[2] with at least 50 photos as a test set. The remaining albums in the scenario form the training set from which we learn the event model. We set the budget $B = 7$ photos.

We compare the above summarization algorithm based on event inference (denoted as ILP) to the following two baselines,

(1) KTH, choose every $K = n/B$ photos starting from the n/Bth photo. If the input

[2] Though we evaluated only on the wedding scenario, our approach is applicable to all scenarios.

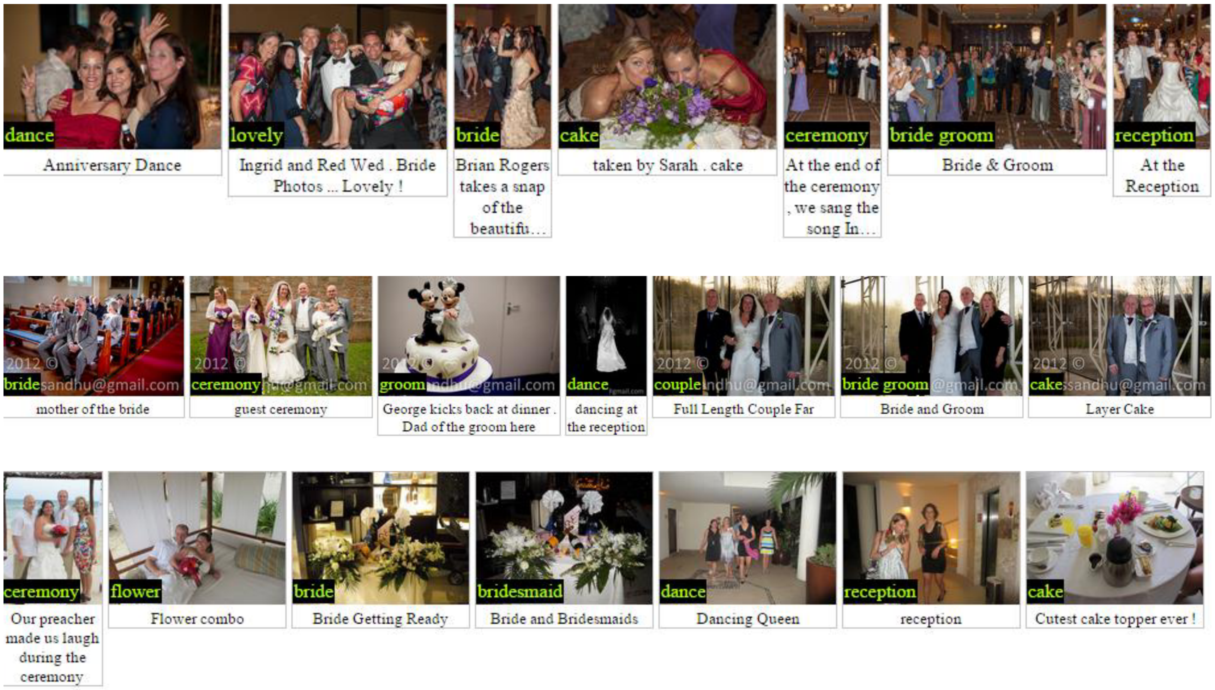


Figure 13: Example album summaries generated by the ILP event inference (with *only the images* as the input). Each row is a summary of one album. The green tag at the bottom-right of each image shows the label of the inferred event of that image.

has only images, we transfer the caption of the query image's most similar image in the training set. This baseline is *not* informed by any event models as the background knowledge.

(2) CLUSTER, similar to ILP except that the first step (event inference) assigns an event e to each photo p *independently*, by maximizing the photo-event affinity $A(p, e)$. If the input album has only images, we set $A(p, e) = A_I(p^I, e)$ (Eq 17, the image-event affinity), otherwise $A(p, e) = A_C(p^C, e)$ (Eq 16, the caption-event affinity).

In both ILP and CLUSTER, if there are only N unique inferred non-miscellaneous events and $N < B$, we select the remaining $B - N$ photos from the photos labelled with the miscellaneous event using the KTH method, or randomly select remaining photos from the entire album if there are no photos labelled with miscellaneous event.

Results We evaluate the performance of the three methods using Amazon Mechanical Turk (AMT). The test set has 100 albums held out from the training set from which we learn the event model. For each album, in random order we present two summaries

methods	selection rates (img.+cap.)		selection rates (img. only)	
ILP VS. KTH	59%	41%	45%	55%
ILP VS. CLUSTER	47%	53%	54%	46%
CLUSTER VS. KTH	57%	43%	53%	47%

Table 17: Human evaluation of album summarization: the percentages of summaries preferred by judges in pairwise comparisons. *img.+cap.* represents each input album has both images and original captions. *img. only* represents each input album has only images (the summarization algorithm also generates captions for the selected images).

generated by two algorithms, respectively. Turkers are instructed to choose a better summary considering both the images and the captions. For each task, we aggregate answers from three turkers by majority voting.

Table 17 shows the percentages of summaries preferred by human judges in pairwise comparisons.

(1) When the input albums have *both* the images *and* the original captions (img.+cap.), both ILP and CLUSTER perform significantly better than KTH (Figure 14). This demonstrates that our event model helps to identify major events and choose representative photos effectively.

(2) When the input albums have *only images* (img. only, see Figure 15), CLUSTER is also preferred over KTH, confirming that knowing the event model improves visual understanding, hence gives a better summarization. However, ILP performs worse than KTH. Our conjecture is that our estimation of the image-event affinity (Eq.17) requires improvement. So far it is based on the global image similarity using CNN feature, which might have its limitations in discriminating the subtle difference of different events in the same scenario (say in a wedding scenario, many events all show groups of people with slightly different actions). More advanced visual features regarding actions or training a probabilistic classifier to compute image-event affinity might be helpful.

4.6 Related works

Researchers have explored unsupervised induction of the salient content structure by exploiting a large collection of text exhibiting redundancies in content. In their pioneering

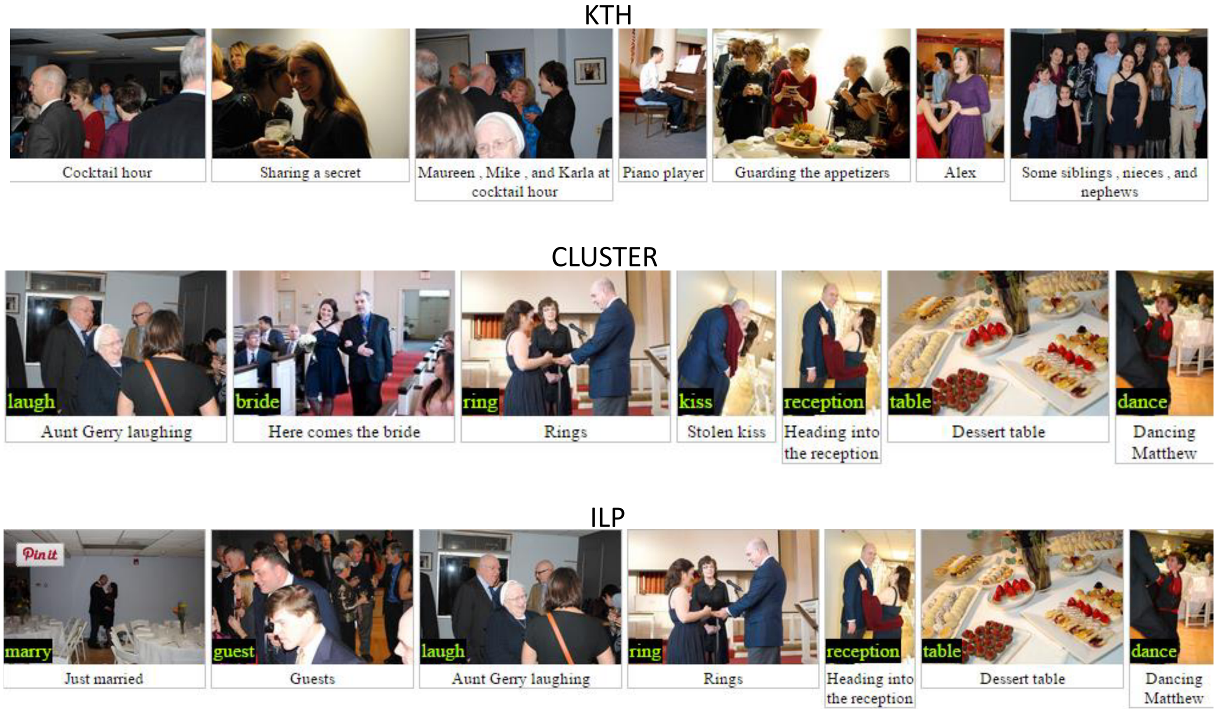


Figure 14: Comparisons of summaries generated by three algorithms (with *both* the images *and* their captions as the input). The green tag at the bottom-right of each image shows the label of the inferred event of that image. Both CLUSTER and ILP select photos that highlight important and representative events (e.g., ring exchange, kiss, and dance) in general, while KTH tends to select more photos that shows random and miscellaneous events.

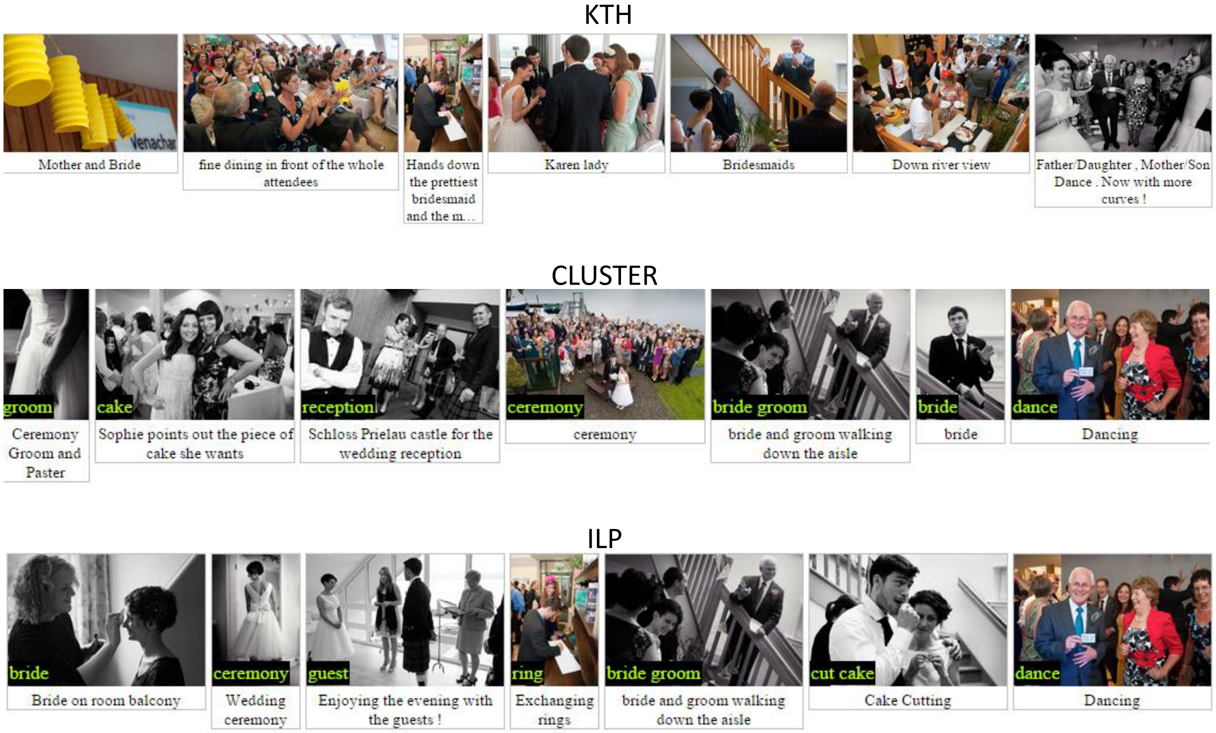

KTH
Mother and Bride
fine dining in front of the whole attendees
Hands down the prettiest bridesmaid and the m...
Karen lady
Bridesmaids
Down river view
Father/Daughter , Mother/Son Dance . Now with more curves !
CLUSTER
groom
Ceremony Groom and Paster
cake
Sophie points out the piece of cake she wants
reception
Schloss Prielau castle for the wedding reception
ceremony
ceremony
bride groom
bride and groom walking down the aisle
bride
bride
dance
Dancing
ILP
bride
Bride on room balcony
ceremony
Wedding ceremony
guest
Enjoying the evening with the guests !
ring
Exchanging rings
bride groom
bride and groom walking down the aisle
cut cake
Cake Cutting
dance
Dancing

Figure 15: Comparisons of summaries generated by three algorithms (with *only the images* as the input). The green tag at the bottom-right of each image shows the label of the inferred event of that image. Both CLUSTER and ILP select photos that highlight important and representative events (e.g., ring exchange, kiss, and dance) in general, while KTH tends to select more photos that shows random and miscellaneous events. The captions of both CLUSTER and ILP have more accurate interpretation of the image content than KTH.

work, Barzilay and Lee (2004) presented the first empirical study demonstrating how statistical regularities in the content flow in newswire articles can be modelled using Hidden Markov Models, for specific topic domains such as earthquakes for which many similar articles exist. The learned content models find applications in automatic summarization and coherency detection. In addition, discovering temporal structure from text has been studied before by building a temporal graph (Bramsen et al., 2006). More recently, new work has been introduced in extracting storylines and summarizing complex events in newswire text (Xu et al., 2013), and learning hierarchical events in social media text (Gu et al., 2011; Gu et al., 2013).

Another line of research that finds the common event structure from a collection of related text, where the learned motifs and plot structure were used to stochastically generate new stories (McIntyre and Lapata, 2009; Goyal et al., 2010; Goyal et al., 2013), or the learned common sense knowledge is used for question answering about a story (Hajishirzi and Mueller, 2011) or sportscasts (Hajishirzi et al., 2011). Our work similarly learns the typical temporal patterns that define common events and experiences, but in an entirely different genre and domain of online photo albums.

Compared to the recent stream of research that learns narrative schemas from natural language corpora (Chambers and Jurafsky, 2008; Chambers and Jurafsky, 2009; Chambers, 2013; Cassidy et al., 2014), or compiles script knowledge from crowd sourcing (Modi and Titov, 2014), our work explores a new source of knowledge that allows grounded schema learning with temporal dimensions, resulting in a new dataset that includes event and scenario types that are not naturally accessible from news wire or literature.

Finally, a few recent studies have explored videos as a source of discovering complex events and learning the sequential patterns of events (Tang et al., 2012; Kim and Xing, 2014a; Kim and Xing, 2014b; Tschiatschek et al., 2014), but explored only the visual modalities without drawing a connection to natural language descriptions.

Chapter 5

Conclusion

To summarize, this thesis investigates language grounded in massive online data, ranging from ontology, images, to time series. We have presented three studies as follows,

5.1 Cost-sensitive hierarchical product classification

Our first study relates text to a **conceptual abstraction** (§2). We study classifying a textual product description into a given taxonomic ontology. Instead of optimizing 0-1 error rate as standard approaches, we design a classifier based on its *use* in the e-commerce world, that is, a vendor organizes a collection of products with a business goal to maximize revenue.

In particular, we investigate two problems, *performance evaluation* and *learning*, in a synergistic way, under a unified view of empirical risk. Performance evaluation chooses an appropriate misclassification cost. Learning minimizes the average misclassification cost. We emphasizes the importance to design an appropriate performance evaluation metric for a real world task, otherwise we are optimizing the wrong objective. We show how to apply such a synergistic way to address the specific task of hierarchical product classification, and demonstrate its effectiveness by experiments on a large dataset. We obtain general insight into *how* and *why* several common evaluation metrics can be misleading, which is applicable to the treatment of performance evaluation of other real world tasks. We propose a general cost-sensitive learning algorithm that minimizes the upper bound of any loss functions, using multi-class SVM with margin re-scaling and loss normalization. The loss normalization approach is also applicable to general classification and structured prediction tasks when using structured SVM with margin re-scaling.

Our work is an application of cost-sensitive learning. Very few works study both class-dependent and example-dependent misclassification cost, especially in a practical scenario as we do. However, application scenarios involving both types of cost are not rare, even becoming more and more common in big data era, forming an emerging class of applications for large scale information and knowledge management.

5.2 Image description using bipartite cross-modal association structure

Our second study straddles text and **vision** (§3). The main challenge to tapping into the online image-caption data is noise. Everyday captions contain extraneous information that is not directly relevant to what the image shows. We provide insights into making a better use of the existing web content and the future content exploding with billions of online activities every day. The key idea is to focus on *Déjà Image-Captions*, i.e., naturally existing image captions that are *repeated almost verbatim* by more than one individual for different images. The hypothesis is that such captions are more likely to be free of unwanted extraneous information (e.g., specific names, time, or any other personal information) and better represent visual concepts. The new corpus of *Déjà Image Captions*, publicly shared at http://54.69.114.42:8080, comprises four million image-caption pairs with about 180K unique captions.

We demonstrate the potential utility of *Déjà Image Captions* in multiple ways: new approaches to image caption retrieval using the associative structure of the corpus (§3.3); strengthening the association structure via *visually-situated paraphrases* to further enhance image captioning (§3.5); *creative* image captioning and paraphrasing that control literalness or figurativeness of the automatic captions (§3.7).

5.3 Multimodal temporal knowledge modelling with photo albums

Our third study aligns text with both **vision** and **time series** (§4). We propose to tap into the context of a photo stream to better understand both photos in sequence and their accompanying captions. The key idea is to ground image captions to prototypical events in a common scenario. For example, from a sequence of photos paired with captions regarding a wedding scenario, we might identify certain typical events in wedding happen over time, for example, *vows*, *ring exchange*, *reception*, and *dancing*.

Concretely, we collect a large-scale dataset of online photo albums aligned with narrative captions and time stamps, for 12 common scenarios in which people participate in their daily life. We propose to learn a multimodal temporal model from albums about a given scenario, which identify prototypical events and their typical temporal ordering in that scenario. Each prototypical event has a set of visual instantiations (images) and a set of textual instantiations (captions). Based on the event model, we then propose a collective event inference algorithm that infers the events of each photo in a given album, which serves as an understanding of the given photo sequence. We demonstrate the effectiveness of our event inference algorithm based on event modelling with two tasks: album segmentation that segment a photo sequence into coherent segments, and album summarization that summarizes a photo album with a few representative images paired with narrative descriptions.

5.4 Contributions

The major contributions of this thesis include:

- ***Data.*** Two large-scale datasets with language grounded in vision: *Déjà Image-Captions* with a bipartite association structure bridging nearly 180K captions and about 4M images (§3); *Common Scenario Albums* that has thousands of photo streams aligned with narrative captions for each of 12 common scenarios, including wedding, camping, and so on (§4).
- ***Models and algorithms.*** (i) A general cost-sensitive learning algorithm based

on multi-class SVM with margin re-scaling and a new loss normalization approach (§2.3); (ii) Image captioning algorithms using bipartite association structure (§3.3 and §3.5); (iii) Automatic visual paraphrasing algorithm (§3.5); (iv) Algorithms for creative image captioning and creative visual paraphrasing, respectively (§3.7); (v) A multi-modal event model of prototypical events and their temporal ordering learned from photo albums in a common scenario (§4.3); (vi) A collective event inference algorithm based on the multimodal event model using Integer Linear Programming to infer the events of each photo in a given album; (vii) A multi-modal summarization algorithm to give an overview of a photo album with a few representative images paired with narrative captions (§4.5.2).

- ***Novel applications.*** (i) Hierarchical Commercial product classification with a business goal of maximizing revenue (§2); (ii) Visually-situated paraphrases (§3.5); (iii) Creative image captioning and creative visual paraphrasing (§3.7); (iv) Multi-modal summarization of online photo albums (§4.5.2).

Bibliography

[Anderson and Andersson2007] Chris Anderson and Mia Poletto Andersson. 2007. *The long tail*. Bonnier fakta.

[Artzi and Zettlemoyer2011] Yoav Artzi and Luke Zettlemoyer. 2011. Bootstrapping semantic parsers from conversations. In *Proceedings of the conference on empirical methods in natural language processing*, pages 421–432. Association for Computational Linguistics.

[Banerjee and Lavie2005] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

[Barzilay and Lee2004] Regina Barzilay and Lillian Lee. 2004. Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization. In *NAACL-HLT*.

[Barzilay and McKeown2001] Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 50–57. Association for Computational Linguistics.

[Berant and Liang2014] Jonathan Berant and Percy Liang. 2014. Semantic Parsing via Paraphrasing. In *Association for Computational Linguistics (ACL)*.

[Berg et al.2010] Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. 2010. Automatic attribute discovery and characterization from noisy web data. In *ECCV 2010*, pages 663–676. Springer.

[Beygelzimer et al.2008] Alina Beygelzimer, John Langford, and Bianca Zadrozny. 2008. Machine learning techniques—reductions between prediction quality metrics. In *Performance Modeling and Engineering*, page 3–28. Springer.

[Boiman et al.2008] Oren Boiman, Eli Shechtman, and Michal Irani. 2008. In defense of Nearest-Neighbor based image classification. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–8, June.

[Bordes et al.2010] Antoine Bordes, Nicolas Usunier, and Jason Weston. 2010. Label ranking under ambiguous supervision for learning semantic correspondences. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 103–110.

[Bordes et al.2011] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Conference on Artificial Intelligence*.

[Bordes et al.2012] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *International Conference on Artificial Intelligence and Statistics*, pages 127–135.

[Bramsen et al.2006] Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006. Inducing Temporal Graphs. In *EMNLP*.

[Cai and Hofmann2004] Lijuan Cai and Thomas Hofmann. 2004. Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, page 78–87, New York, NY, USA. ACM.

[Cassidy et al.2014] Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An Annotation Framework for Dense Event Ordering. In *ACL*.

[Chambers and Jurafsky2008] Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *ACL*, pages 789–797.

[Chambers and Jurafsky2009] Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint*

Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, pages 602–610.

[Chambers2013] Nathanael Chambers. 2013. Event Schema Induction with a Probabilistic Entity-Driven Model. In *EMNLP*.

[Chen and Dolan2011] David L. Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics.

[Chen and Lin2006] Yi-Wei Chen and Chih-Jen Lin. 2006. Combining SVMs with various feature selection strategies. In *Feature Extraction*, page 315–324. Springer.

[Chen et al.2010] David L. Chen, Joohyun Kim, and Raymond J. Mooney. 2010. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37(1):397–436.

[Costa et al.2007] E. Costa, A. Lorena, ACPLF Carvalho, and A. Freitas. 2007. A review of performance evaluation measures for hierarchical classifiers. In *Evaluation Methods for Machine Learning II: papers from the AAAI-2007 Workshop*, page 1–6.

[Crammer and Singer2002] Koby Crammer and Yoram Singer. 2002. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292.

[Dekel et al.] Ofer Dekel, Joseph Keshet, and Yoram Singer. Large margin hierarchical classification. In *In Proceedings of the Twenty-First International Conference on Machine Learning*, page 209–216.

[Deng et al.2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

[Deselaers and Ferrari2011] Thomas Deselaers and Vittorio Ferrari. 2011. Visual and semantic similarity in imagenet. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1777–1784. IEEE.

[Dodge et al.2012] Jesse Dodge, Amit Goyal, Xufeng Han, Alyssa Mensch, Margaret Mitchell, Karl Stratos, Kota Yamaguchi, Yejin Choi, Hal Daumé III, Alexander C.

Berg, and others. 2012. Detecting visual text. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 762–772. Association for Computational Linguistics.

[Dolan et al.2004] Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics.

[Domingos1999] Pedro Domingos. 1999. MetaCost: a general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, page 155–164.

[Dumais and Chen2000] Susan Dumais and Hao Chen. 2000. Hierarchical classification of web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, page 256–263, New York, NY, USA. ACM.

[Elkan2001] Charles Elkan. 2001. The foundations of cost-sensitive learning. In *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, page 973–978.

[Elliott and Keller2014] Desmond Elliott and Frank Keller. 2014. Comparing automatic evaluation measures for image description. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 452–457.

[Fan et al.2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: a library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

[Feng et al.2015] Song Feng, Sujith Ravi, Ravi Kumar, Polina Kuznetsova, Wei Liu, Alexander C. Berg, Tamara L. Berg, and Yejin Choi. 2015. Refer-to-as Relations as Semantic Knowledge. In *AAAI Conference on Artificial Intelligence.*

[Freitas and de Carvalho2007] Alex A. Freitas and Andre CPFL de Carvalho. 2007. A tutorial on hierarchical classification with applications in bioinformatics.

[Ganitkevitch et al.2013] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of NAACL-HLT*, pages

758–764, Atlanta, Georgia, June. Association for Computational Linguistics.

[Gong et al.2014] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. 2014. Improving Image-Sentence Embeddings Using Large Weakly Annotated Photo Collections. In *ECCV 2014*, pages 529–545. Springer.

[Goyal et al.2010] Amit Goyal, Ellen Riloff, and Hal Daumé III. 2010. Automatically Producing Plot Unit Representations for Narrative Text. In *EMNLP)*.

[Goyal et al.2013] Amit Goyal, Ellen Riloff, and Hal Daumé III. 2013. A Computational Model for Plot Units. In *CIJ*.

[Gu et al.2011] Hansu Gu, Xing Xie, Qin Lv, Yaoping Ruan, and Li Shang. 2011. ETree: Effective and Efficient Event Modeling for Real-Time Online Social Media Networks. In *2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 300–307, August.

[Gu et al.2013] Hansu Gu, Mike Gartrell, Liang Zhang, Qin Lv, and Dirk Grunwald. 2013. AnchorMF: Towards Effective Event Context Identification. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 629–638, New York, NY, USA. ACM.

[Hajishirzi and Mueller2011] Hannaneh Hajishirzi and Erik T. Mueller. 2011. Symbolic Probabilistic Reasoning for Narratives. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

[Hajishirzi et al.2011] Hannaneh Hajishirzi, Julia Hockenmaier, Erik T. Mueller, and Eyal Amir. 2011. Reasoning about robocup soccer narratives. In *In Proc. Conference on Uncertainty in Artificial Intelligence (UAI*.

[Hajishirzi et al.2012] Hannaneh Hajishirzi, Mohammad Rastegari, Ali Farhadi, and Jessica K. Hodgins. 2012. Semantic understanding of professional soccer commentaries. *arXiv preprint arXiv:1210.4854*.

[Hardoon et al.2004] David Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664.

[Harnad1990] Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346.

[Hodosh and Hockenmaier2013] Micah Hodosh and Julia Hockenmaier. 2013. Sentence-based image description with scalable, explicit models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 294–300.

[Hodosh et al.2013] Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47(1):853–899.

[Kannan et al.2011] A. Kannan, P.P. Talukdar, N. Rasiwasia, and Qifa Ke. 2011. Improving product classification using images. In *2011 IEEE 11th International Conference on Data Mining (ICDM)*, pages 310 –319, December.

[Karpathy et al.2014] Andrej Karpathy, Armand Joulin, and Fei Fei F. Li. 2014. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*, pages 1889–1897.

[Keerthi et al.2008] S. Sathiya Keerthi, Sellamanickam Sundararajan, Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. 2008. A sequential dual method for large scale multi-class linear SVMs. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 408–416.

[Kim and Mooney2012] Joohyun Kim and Raymond J. Mooney. 2012. Unsupervised PCFG Induction for Grounded Language Learning with Highly Ambiguous Supervision. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 433–444, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Kim and Xing2014a] G. Kim and E. P. Xing. 2014a. Reconstructing Storyline Graphs for Image Recommendation from Web Community Photos. In *CVPR*.

[Kim and Xing2014b] G. Kim and L. Sigal E. P. Xing. 2014b. Jointly Summarizing Large-Scale Web Images and Videos for the Storyline Reconstruction. In *CVPR*.

[Koncel-Kedziorski et al.2014] R. Koncel-Kedziorski, Hannaneh Hajishirzi, and Ali Farhadi. 2014. Multi-Resolution Language Grounding with Weak Supervision. EMNLP.

[Krishnamoorthy et al.2013] Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond J. Mooney, Kate Saenko, and Sergio Guadarrama. 2013. Generating natural-language video descriptions using text-mined knowledge. In *AAAI*, volume 1, page 2.

[Kuznetsova et al.2012] Polina Kuznetsova, Vicente Ordonez, Alexander C. Berg, Tamara L. Berg, and Yejin Choi. 2012. Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 359–368. Association for Computational Linguistics.

[Kuznetsova et al.2013a] Polina Kuznetsova, Jianfu Chen, and Yejin Choi. 2013a. Understanding and Quantifying Creativity in Lexical Composition. In *EMNLP*, pages 1246–1258.

[Kuznetsova et al.2013b] Polina Kuznetsova, Vicente Ordonez, Alexander C. Berg, Tamara L. Berg, and Yejin Choi. 2013b. Generalizing Image Captions for Image-Text Parallel Corpus. In *ACL (2)*, pages 790–796.

[Kuznetsova et al.2014] Polina Kuznetsova, Vicente Ordonez, Tamara Berg, and Yejin Choi. 2014. TreeTalk: Composition and Compression of Trees for Image Descriptions. *Transactions of the Association for Computational Linguistics*.

[Le et al.2013] Dieu-Thu Le, Jasper RR Uijlings, and Raffaella Bernardi. 2013. Exploiting language models for visual recognition. In *EMNLP*, pages 769–779.

[Lee et al.2010] Cheng Few Lee, Jack C. Lee, and Alice C. Lee. 2010. Normal, lognormal distribution and option pricing model. In *Handbook of Quantitative Finance and Risk Management*, page 421–428. Springer.

[Li et al.2013] Hongsong Li, Kenny Q. Zhu, and Haixun Wang. 2013. Data-Driven Metaphor Recognition and Explanation. *TACL*, 1:379–390.

[Liang et al.2013] Percy Liang, Michael I. Jordan, and Dan Klein. 2013. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.

[Lin et al.2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, Zürich.

[Mason and Charniak2014] Rebecca Mason and Eugene Charniak. 2014. Nonparametric Method for Data-driven Image Captioning. In *NAACL*.

[Matuszek et al.2014] Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. 2014. Learning from unscripted deictic gesture and language for human-robot interactions.

[McIntyre and Lapata2009] Neil McIntyre and Mirella Lapata. 2009. Learning to Tell Tales: A Data-driven Approach to Story Generation. In *ACL*.

[Meeker2014] Mary Meeker. 2014. *Internet Trends 2014*.

[Miller1995] George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

[Modi and Titov2014] Ashutosh Modi and Ivan Titov. 2014. Inducing neural models of script knowledge. *CoNLL-2014*, page 49.

[Oliva and Torralba2001] Aude Oliva and Antonio Torralba. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175.

[Ordonez et al.2011] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. 2011. Im2text: Describing Images Using 1 Million Captioned Photographs. In *NIPS*, volume 1, page 4.

[Ordonez et al.2013] Vicente Ordonez, Jia Deng, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. 2013. From large scale image categorization to entry-level categories. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2768–2775. IEEE.

[Ozbal and Strapparava2012] Gozde Ozbal and Carlo Strapparava. 2012. A Computational Approach to the Automation of Creative Naming. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 703–711, Jeju Island, Korea, July. Association for Computational Linguistics.

[Pang et al.2003] Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 102–109. Association for Computational Linguistics.

[Papineni et al.2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

[Pecher et al.2011] Diane Pecher, Inge Boot, and Saskia Van Dantzig. 2011. Abstract concepts: Sensory-motor grounding, metaphors, and beyond.

[Poon2013] Hoifung Poon. 2013. Grounded unsupervised semantic parsing. In *ACL*, pages 933–943.

[Rashtchian et al.2010] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. Collecting Image Annotations Using Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 139–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Rasiwasia et al.2007] Nikhil Rasiwasia, Pedro J. Moreno, and Nuno Vasconcelos. 2007. Bridging the gap: Query by semantic example. *Multimedia, IEEE Transactions on*, 9(5):923–938.

[Rasiwasia et al.2010] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert R.G. Lanckriet, Roger Levy, and Nuno Vasconcelos. 2010. A New Approach to Cross-modal Multimedia Retrieval. In *Proceedings of the International Conference on Multimedia*, MM '10, pages 251–260, New York, NY, USA. ACM.

[Riedel et al.2013] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas.

[Roller2004] Ben Taskar Carlos Guestrin Daphne Roller. 2004. Max-margin markov networks. In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*, volume 16, page 25.

[Schank and Abelson1977] Roger C. Schank and Robert P. Abelson. 1977. Scripts, plans, goals, and understanding: An inquiry into human knowledge structures.

[Sculley et al.2011] D. Sculley, Matthew Eric Otey, Michael Pohl, Bridget Spitznagel, John Hainsworth, and Yunkai Zhou. 2011. Detecting adversarial advertisements in the wild. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 274–282.

[Seo et al.2014] Min Joon Seo, Hannaneh Hajishirzi, Ali Farhadi, and Oren Etzioni. 2014. Diagram Understanding in Geometry Questions. In *The AAAI Conference on Artificial Intelligence (AAAI-2014), Québec City, Québec, Canada.*

[Seo et al.2015] Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving Geometry Problems: Combining Text and Diagram Interpretation.

[Shen et al.2009] Dou Shen, Ying Li, Xiao Li, and Dengyong Zhou. 2009. Product query classification. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, page 741–750, New York, NY, USA. ACM.

[Shen et al.2012] Dan Shen, Jean-David Ruvini, and Badrul Sarwar. 2012. Large-scale item categorization for e-commerce. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, page 595–604, New York, NY, USA. ACM.

[Shutova2010] Ekaterina Shutova. 2010. Models of metaphor in NLP. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 688–697, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Silberer and Lapata2014] Carina Silberer and Mirella Lapata. 2014. Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 721–732. The Association for Computer Linguistics.

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

[Sun and Lim2001] Aixin Sun and Ee-Peng Lim. 2001. Hierarchical text classification and evaluation. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, page 521–528.

[Tang et al.2012] Kevin Tang, Li Fei-Fei, and Daphne Koller. 2012. Learning Latent Temporal Structure for Complex Event Detection. In *CVPR*.

[Teo et al.2010] Choon Hui Teo, S. V. N. Vishwanthan, Alex J. Smola, and Quoc V. Le. 2010. Bundle methods for regularized risk minimization. *The Journal of Machine Learning Research*, 11:311–365.

[Torralba et al.2008] Antonio Torralba, Robert Fergus, and William T. Freeman. 2008. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1958–1970.

[Tschiatschek et al.2014] Sebastian Tschiatschek, Rishabh Iyer, Haochen Wei, and and Jeff Bilmes. 2014. Learning Mixtures of Submodular Functions for Image Collection Summarization. In *NIPS*.

[Tsochantaridis et al.2006] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, Yasemin Altun, and Yoram Singer. 2006. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(2):1453.

[Tsvetkov et al.2014] Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of ACL*.

[Vapnik1999] Vladimir Vapnik. 1999. *The nature of statistical learning theory*. springer.

[Veale2011] Tony Veale. 2011. Creative Language Retrieval: A Robust Hybrid of Information Retrieval and Linguistic Creativity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 278–287, Portland, Oregon, USA, June. Association for Computational Linguistics.

[Venugopalan et al.2015] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2015. Translating videos to natural language using deep recurrent neural networks.

[Wilson et al.2005] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.

[Wittgenstein2010] Ludwig Wittgenstein. 2010. *Philosophical investigations.* John Wiley & Sons.

[Xu et al.2013] Shize Xu, Shanshan Wang, and Yan Zhang. 2013. Summarizing Complex Events: a Cross-Modal Solution of Storylines Extraction and Reconstruction. In *EMNLP*, pages 1281–1291.

[Yu and Siskind2013] Haonan Yu and Jeffrey Mark Siskind. 2013. Grounded language learning from video described with sentences. In *ACL (1)*, pages 53–63.

[Zadrozny et al.2003] B. Zadrozny, J. Langford, and N. Abe. 2003. Cost-sensitive learning by cost-proportionate example weighting. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, page 435–442.

[Zettlemoyer and Collins2005] L. S. Zettlemoyer and M. Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proc. of UAI*, volume 5.

[Zhang and Weld2013] Congle Zhang and Daniel S Weld. 2013. Harvesting Parallel News Streams to Generate Paraphrases of Event Relations. In *EMNLP*, pages 1776–1786.

[Zhou and Liu2006] Z. Zhou and X. Liu. 2006. On multi-class cost-sensitive learning. In *Proceedings of the 21st national conference on artificial intelligence*, volume 21, page 567.

[Ziegler et al.2004] Cai-Nicolas Ziegler, Georg Lausen, and Lars Schmidt-Thieme. 2004. Taxonomy-driven computation of product recommendations. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, page 406–415, New York, NY, USA. ACM.