

Zero-shot Learning of Classifiers from Natural Language Quantification

Shashank Srivastava Igor Labutov Tom Mitchell

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA

ssrivastava@cmu.edu ilabutov@cs.cmu.edu tom.mitchell@cmu.edu

Abstract

Humans can efficiently learn new concepts using language. We present a framework through which a set of explanations of a concept can be used to learn a classifier without access to any labeled examples. We use semantic parsing to map explanations to probabilistic assertions grounded in latent class labels and observed attributes of unlabeled data, and leverage the differential semantics of linguistic quantifiers (e.g., ‘usually’ vs ‘always’) to drive model training. Experiments on three domains show that the learned classifiers outperform previous approaches for learning with limited data, and are comparable with fully supervised classifiers trained from a small number of labeled examples.

1 Introduction

As computer systems that interact with us in natural language become pervasive (e.g., Siri, Alexa, Google Home), they suggest the possibility of letting users teach machines in language. The ability to learn from language can enable a paradigm of ubiquitous machine learning, allowing users to teach personalized concepts (e.g., identifying ‘important emails’ or ‘project-related emails’) when limited or no training data is available.

In this paper, we take a step towards solving this problem by exploring the use of quantifiers to train classifiers from declarative language. For illustration, consider the hypothetical example of a user explaining the concept of an “important email” through natural language statements (Figure 1). Our framework takes a set of such natural language explanations describing a concept (e.g., “emails that I reply to are usually important”) and a set of unlabeled instances as input, and produces

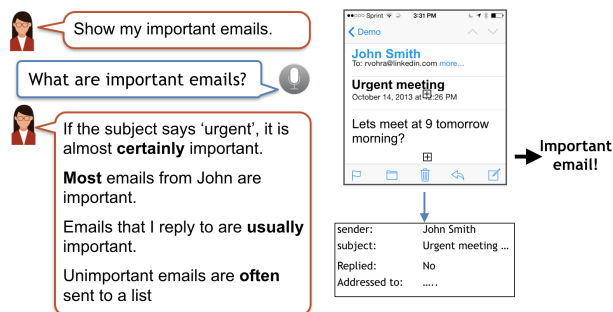


Figure 1: Supervision from language can enable concept learning from limited or even no labeled examples. Our approach assumes the learner has sensors that can extract attributes from data, such as those listed in the table, and language that can refer to these sensors and their values.

a binary classifier (for important emails) as output. Our hypothesis is that language describing concepts encodes key properties that can aid statistical learning. These include specification of relevant attributes (e.g., whether an email was replied to), relationships between such attributes and concept labels (e.g., if a reply implies the class label of that email is ‘important’), as well as the strength of these relationships (e.g., via quantifiers like ‘often’, ‘sometimes’, ‘rarely’). We infer these properties automatically, and use the semantics of linguistic quantifiers to drive the training of classifiers *without labeled examples for any concept*. This is a novel scenario, where previous approaches in semi-supervised and constraint-based learning are not directly applicable. Those approaches require manual pre-specification of expert knowledge for model training. In our approach, this knowledge is automatically inferred from noisy natural language explanations from a user.

Our approach is summarized in the schematic in Figure 2. First, we map the set of natural language explanations of a concept to logical forms

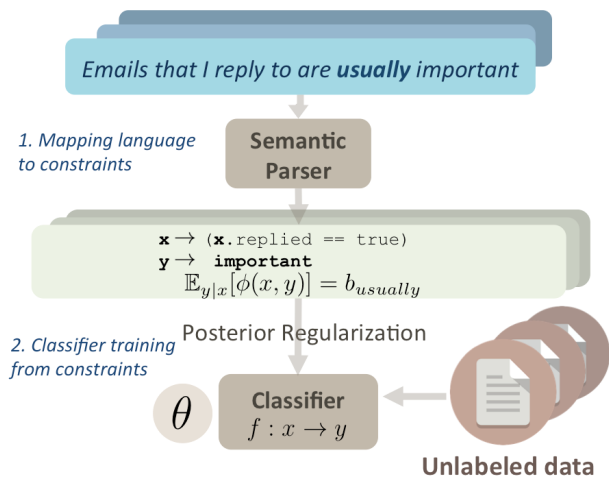


Figure 2: Our approach to Zero-shot learning from Language. Natural language explanations on how to classify concept examples are parsed into formal constraints relating features to concept labels. The constraints are combined with unlabeled data, using posterior regularization to yield a classifier.

that identify the attributes mentioned in the explanation, and describe the information conveyed about the attribute and the concept label as a quantitative constraint. This mapping is done through semantic parsing. The logical forms denote quantitative constraints, which are probabilistic assertions about observable attributes of the data and unobserved concept labels. Here the strength of a constraint is assumed to be specified by a linguistic quantifier (such as ‘all’, ‘some’, ‘few’, etc., which reflect degrees of generality of propositions). Next, we train a classification model that can assimilate these constraints by adapting the posterior regularization framework (Ganchev et al., 2010).

Intuitively, this can be seen as defining an optimization problem, where the objective is to find parameter estimates for the classifier that do not simply fit the data, but also agree with the human provided natural language advice to the greatest extent possible. Since logical forms can be grounded in a variety of sensors and external resources, an explicit model of semantic interpretation conceptually allows the framework to subsume a flexible range of grounding behaviors. The main contributions of this work are:

1. We introduce the problem of zero-shot learning of classifiers from language, and present an approach towards this.
2. We develop datasets for zero-shot classification from natural descriptions, exhibiting

tasks with various levels of difficulty.

3. We empirically show that coarse probability estimates to model linguistic quantifiers can effectively supervise model training across three domains of classification tasks.

2 Related Work

Many notable approaches have explored incorporation of background knowledge into the training of learning algorithms. However, none of them addresses the issue of learning from natural language. Prominent among these are the Constraint-driven learning (Chang et al., 2007a), Generalized Expectation (Mann and McCallum, 2010) and Posterior Regularization (Ganchev et al., 2010) and Bayesian Measurements (Liang et al., 2009) frameworks. All of these require domain knowledge to be manually programmed in before learning. Similarly, Probabilistic Soft Logic (Kimmig et al., 2012) allows users to specify rules in a logical language that can be used for reasoning over graphical models. More recently, multiple approaches have explored few-shot learning from perspective of term or attribute-based transfer (Lampert et al., 2014), or learning representations of instances as probabilistic programs (Lake et al., 2015).

Other work (Lei Ba et al., 2015; Elhoseiny et al., 2013) considers language terms such as colors and textures that can be directly grounded in visual meaning in images. Some previous work (Srivastava et al., 2017) has explored using language explanations for feature space construction in concept learning tasks, where the problem of learning to interpret language, and learning classifiers is treated jointly. However, this approach assumes availability of labeled data for learning classifiers. Also notable is recent work by Andreas et al. (2017), who propose using language descriptions as parameters to model structure in learning tasks in multiple settings. More generally, learning from language has also been previously explored in tasks such as playing games (Branavan et al., 2012), robot navigation (Karamcheti et al., 2017), etc.

Natural language quantification has been studied from multiple perspectives in formal logic (Barwise and Cooper, 1981), linguistics (Löbner, 1987; Bach et al., 2013) and cognitive psychology (Kurtzman and MacDonald, 1993). While quantification has traditionally been defined in set-theoretic terms in linguistic theories¹, our approach joins alternative

¹e.g., ‘some A are B’ $\Leftrightarrow A \cap B \neq \emptyset$

perspectives that represent quantifiers probabilistically (Moxey and Sanford, 1993; Yildirim et al., 2013). To the best of our knowledge, this is the first work to leverage the semantics of quantifiers to guide statistical learning models.

3 Learning Classifiers from Language

Our approach relies on first mapping natural language descriptions to quantitative constraints that specify statistical relationships between observable attributes of instances and their latent concept labels (Step 1 in Figure 2). These quantitative constraints are then imbued into the training of a classifier by guiding predictions from the learned models to concur with them (Step 2). We use semantic parsing to interpret sentences as quantitative constraints, and adapt the posterior regularization principle for our setting to estimate the classifier parameters. Next, we describe these steps in detail. Since learning in this work is largely driven by the semantics of linguistic quantifiers, we call our approach **Learning from Natural Quantification**, or **LNQ**.

3.1 Mapping language to constraints

A key challenge in learning from language is converting free-form language to representations that can be reasoned over, and grounded in data. For example, a description such as ‘*emails that I reply to are usually important*’ may be converted to a mathematical assertion such as $P(\text{important} \mid \text{replied} : \text{true}) = 0.7$, which statistical methods can reason with. Here, we argue that this process can be automated for a large number of real-world descriptions. In interpreting statements describing concepts, we infer the following key elements:

1. *Feature x* , which is grounded in observed attributes of the data. For our example, ‘emails replied to’ can refer to a predicate such as `replied:true`, which can be evaluated in context of emails to indicate the whether an email was replied to. Incorporating compositional representations enables more complex reasoning. e.g., ‘*the subject of course-related emails usually mentions CS100*’ can map to a composite predicate like `isStringMatch(field:subject, stringVal('CS100'))`, which can be evaluated for different emails to reflect whether their subject mentions ‘CS100’. Mapping language to executable feature functions has been shown to be effective (Srivastava et al., 2017). For sake of simplicity, here we assume that a statement refers to a

single feature, but the method can be extended to handle more complex descriptions.

2. *Concept label y* , specifying the class of instances a statement refers to. For binary classes, this reduces to examples or non-examples of a concept. For our running example, y corresponds to the positive class of important emails.

3. *Constraint-type* asserted by the statement. We argue that most concept descriptions belong to one of three categories shown in Table 2, and these constitute our vocabulary of constraint types for this work. For our running example (‘emails that I reply to are usually important’), the type corresponds to $P(y \mid x)$, since the syntax of the statement indicates an assertion conditioned on the feature indicating whether an email was replied to. On the other hand, an assertion such as ‘I usually reply to important emails’ indicates an assertion conditioned on the set important emails, and therefore corresponds to the type $P(x \mid y)$.

4. *Strength* of the constraint. We assume this to be specified by a quantifier. For our running example, this corresponds to the adverb ‘usually’. In this work, by *quantifier* we specifically refer to frequency adverbs (‘usually’, ‘rarely’, etc.) and frequency determiners (‘few’, ‘all’, etc.).² Our thesis is that the semantics of quantifiers can be leveraged to make statistical assertions about relationships involving attributes and concept labels. One way to do this might be to simply associate point estimates of probability values, suggesting the fraction of truth values for assertions described with these quantifiers. Table 1 shows probability values we assign to some common frequency quantifiers for English. These values were set simply based on the authors’ intuition about their semantics, and do not reflect any empirical distributions. See Figure 8 for empirical distributions corresponding to some linguistic quantifiers in our data. While these probability values maybe inaccurate, and the semantics of these quantifiers may also change based on context and the speaker, they can still serve as a strong signal for learning classifiers since they are not used as hard constraints, but serve to bias classifiers towards better generalization.

We use a semantic parsing model to map statements to formal semantic representations that specify these aspects. For example, the statement ‘Emails that I reply to are usually important’ is

²This is a significantly restricted definition, and does not address non-frequency determiners (e.g., ‘the’, ‘only’, etc.) or mass quantifiers (e.g. ‘a lot’, ‘little’), among other categories.

Frequency quantifier	Probability
all, always, certainly, definitely	0.95
usually, normally, generally, likely, typically	0.70
most, majority	0.60
often, half	0.50
many	0.40
sometimes, frequently, some	0.30
few, occasionally	0.20
rarely, seldom	0.10
never	0.05

Table 1: Probability values we assign to common linguistic quantifiers (hyper-parameters for method)

mapped to a logical form like `(x→replied:true y→positive type:y|x quant:usually)`.

3.1.1 Semantic Parser components

Given a descriptive statement s , the parsing problem consists of predicting a logical form l that best represents its meaning. In turn, we formulate the probability of the logical form l as decomposing into three component factors: (i) probability of observing a feature and concept labels l_{xy} based on the text of the sentence, (ii) probability of the type of the assertion l_{type} based on the identified feature, concept label and syntactic properties of the sentence s , and (iii) identifying the linguistic quantifier, l_{quant} , in the sentence.

$$P(l | s) = P(l_{xy} | s) P(l_{type} | l_{xy}, s) P(l_{quant} | s)$$

We model each of the three components as follows: by using a traditional semantic parser for the first component, training a Max-Ent classifier for the constraint-type for the second component, and looking for an explicit string match to identify the quantifier for the third component.

Identifying features and concept labels, l_{xy} : For identifying the feature and concept label mentioned in a sentence, we presume a linear score $\mathcal{S}(s, l_{xy}) = w^T \psi(s, l_{xy})$ indicating the goodness of assigning a partial logical form, l_{xy} , to a sentence s . Here, $\psi(s, l_{xy}) \in \mathbb{R}^n$ are features that can depend on both the sentence and the partial logical form, and $w \in \mathbb{R}^n$ is a parameter weight-vector for this component. Following recent work in semantic parsing (Liang et al., 2011), we assume a loglinear distribution over interpretations of a sentence.

$$P(l_{xy} | s) \propto w^T \psi(s, l_{xy})$$

Provided data consisting of statements labeled with logical forms, the model can be trained via maximum likelihood estimation, and be used to predict interpretations for new statements. For training this component, we use a CCG semantic parsing formalism, and follow the feature-set from Zettlemoyer and Collins (2007), consisting of simple indicator features for occurrences of keywords and lexicon entries. This is also compatible with the semantic parsing formalism in Srivastava et al. (2017), whose data (and accompanying lexicon) are also used in our evaluation. For other datasets with predefined features, this component is learned easily from simple lexicons consisting of trigger words for features and labels.³ This component is the only part of the parser that is domain-specific. We note that while this component assumes a domain-specific lexicon (and possibly statement annotated with logical forms), this effort is one-time-only, and will find re-use across the possibly large number of concepts in the domain (e.g., email categories).

Identifying assertion type, l_{type} : The principal novelty in our semantic parsing model is in identifying the type of constraint asserted by a statement. For this, we train a MaxEnt classifier, which uses positional and syntactic features based on the text-spans corresponding to feature and concept mentions to predict the constraint type. We extract the following features from a statement:

1. Boolean value indicating whether the text-span corresponding to the feature x precedes the text span for the concept label y .
2. Boolean value indicating if sentence is in passive (rather than active) voice, as identified by the occurrence of `nsubjpass` dependency relation.
3. Boolean value indicating whether head of the text-span for x is a noun, or a verb.
4. Features indicating the occurrence of conditional tokens ('if', 'then' and 'that') preceding or following text-spans for x and y .
5. Features indicating presence of a linguistic quantifier in a `det` or an `advmod` relation with syntactic head of x or y .

Since the constraint type is determined by syntactic and dependency parse features, this

³We also identify whether a feature x is negated, through the existence of a `neg` dependency relation with the head of its text-span. e.g., *Important emails are usually not deleted*

Type	Example description	Conversion to Expectation Constraint
$P(y x)$	Emails that I reply to are usually important	$\mathbb{E}[\mathbb{I}_{y=important,reply(x):true}] - p_{usually} \times \mathbb{E}[\mathbb{I}_{reply(x):true}] = 0$
$P(x y)$	I often reply to important emails	$\mathbb{E}[\mathbb{I}_{y=important,reply(x):true}] - p_{often} \times \mathbb{E}[\mathbb{I}_{y=important}] = 0$
$P(y)$	I rarely get important emails	Same as $P(y x_0)$, where x_0 is a constant feature

Table 2: Common constraint-types, and their representation as expectations over feature values

component does not need to be retrained for new domains. In this work, we trained this classifier based on a manually annotated set of 80 sentences describing classes in the small UCI Zoo dataset (Lichman, 2013), and used this model for all experiments.

Identifying quantifiers, l_{quant} : Multiple linguistic quantifiers in a sentence are rare, and we simply look for the first occurrence of a linguistic quantifier in a sentence, i.e. $P(l_{quant}|s)$ is a deterministic function. We note that many real world descriptions of concepts lack an explicit quantifier. e.g., ‘*Emails from my boss are important*’. In this work, we ignore such statements for the purpose of training. Another treatment might be to model these statements as reflecting a default quantifier, but we do not explore this direction here. Finally, the decoupling of quantification from logical representation is a key decision. At the cost of linguistic coarseness, this allows modeling quantification irrespective of the logical representation (lambda calculus, predicate-argument structures, etc.).

3.2 Classifier training from constraints

In the previous section, we described how individual explanations can be mapped to probabilistic assertions about observable attributes (e.g., the statement ‘Emails that I reply to are usually important’ may map to $P(y = important | replied = true) = p_{usually}$). Here, we describe how a set of such assertions can be used in conjunction with unlabeled data to train classification models.

Our approach relies on having predictions from the classifier on a set of unlabeled examples ($X = \{x_1 \dots x_n\}$) agree with human-provided advice (in form of constraints). The unobserved concept labels ($Y = \{y_1 \dots y_n\}$) for the unlabeled data constitute *latent variables* for our method. The training procedure can be seen as iteratively inferring the latent concept labels for unlabeled examples so as to agree with the human advice, and updating the classification models by taking these labels as given. While there are multiple approaches for training statistical models with constraints on latent

variables, here we use the Posterior Regularization (PR) framework. The PR objective can be used to optimize a latent variable model subject to a set of constraints, which specify preferences for values of the posterior distributions $p_\theta(Y | X)$.

$$J_Q(\theta) = \mathcal{L}(\theta) - \min_{q \in Q} KL(q | p_\theta(Y|X))$$

Here, the set Q represents a set of *preferred* posterior distributions over latent variables Y , and is defined as $Q := \{q_X(Y) : \mathbb{E}_q[\phi(X, Y)] \leq \mathbf{b}\}$. The overall objective consists of two components, representing how well does a model θ explain the data (likelihood term $\mathcal{L}(\theta)$), and how far it is from the set Q (KL-divergence term).

In our case, each parsed statement defines a probabilistic constraint. The conjunction of all such constraints defines Q (representing models that exactly agree with human-provided advice). Thus, optimizing the objective reflects a tension between choosing models that increase data likelihood, and emulating language advice.

Converting to PR constraints: The set of constraints that PR can handle can be characterized as bounds on expected values of functions (ϕ) of X and Y (or equivalently, from linearity of expectation, as linear inequalities over expected values of functions of X and Y). To use the framework, we need to ensure that each constraint type in our vocabulary can be expressed in such a form.

Following the plan in Table 2, each constraint type can be converted in an equivalent form $\mathbb{E}_q[\phi(X, Y)] = b$, compatible with PR. In particular, each of these constraint types in our vocabulary can be expressed as equations about expectation values of joint indicator functions of label assignments to instances and their attributes. To explain, consider the assertion $P(y = important | replied : true) = p_{usually}$. The probability on the LHS can be expressed as the empirical fraction $\frac{\sum_i \mathbb{E}[\mathbb{I}_{y_i=important,replied:true}]}{\sum_i \mathbb{E}[\mathbb{I}_{replied:true}]}$, which leads to the linear constraints seen in Table 2 (expected values in the table hide summations over instances for brevity). Here, \mathbb{I} denote indicator functions. Thus, we can incorporate probability constraints into our

adaptation of the PR scheme.

Learning and Inference: We choose a loglinear parameterization for the concept classifier.

$$p_{\theta}(y_i | x_i) \propto \exp(y\theta^T x)$$

The training of the classifier follows the modified EM procedure described in [Ganchev et al. \(2010\)](#). As proposed in the original work, we solve a relaxed version of the optimization that allows slack variables, and modifies the PR objective with a L_2 regularizer. This allows solutions even when the problem is over-constrained, and the set Q is empty (e.g. due to contradictory advice).

$$J'(\theta, q) = \mathcal{L}(\theta) - KL(q|p_{\theta}(Y|X)) - \lambda \|\mathbb{E}_q[\phi(X, Y)] - b\|^2$$

The key step in the training is the computation of the posterior regularizer in the E-step.

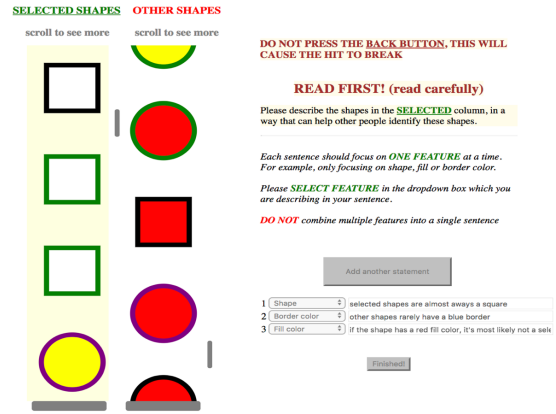
$$\operatorname{argmin}_q KL(q | p_{\theta}) + \lambda \|\mathbb{E}_q[\phi(X, Y)] - b\|^2$$

This objective is strictly convex, and all constraints are linear in q . We follow the optimization procedure from [Bellare et al. \(2009\)](#), whereby the minimization problem in the E-step can be efficiently solved through gradient steps in the dual space. In the M-step, we update the model parameters for the classifier based on label distributions q estimated in the E-step. This simply reduces to estimating the parameters θ for the logistic regression classifier, when class label probabilities are known. In all experiments, we run EM for 20 iterations and use a regularization coefficient of $\lambda = 0.1$.

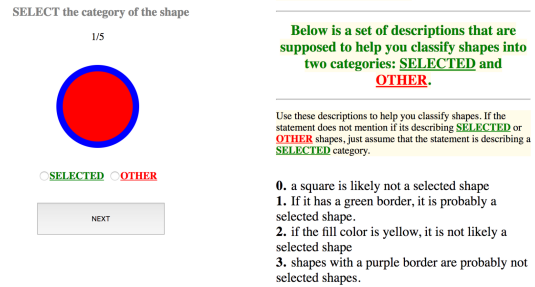
4 Datasets

For evaluating our approach, we created datasets of classification tasks paired with descriptions of the classes, as well as used some existing resources. In this section, we summarize these steps.

Shapes data: To experiment with our approach in a wider range of controlled settings, part of our evaluation focuses on synthetic concepts. For this, we created a set of 50 shape classification tasks that exhibit a range of difficulty, and elicited language descriptions spanning a variety of quantifier expressions. The tasks require classifying geometric shapes with a set of predefined attributes (fill color, border, color, shape, size) into two concept-labels (abstractly named ‘selected shape’, and ‘other’). The datasets were created through a generative process, where features x_i are conditionally independent given the concept-label. Each feature’s conditional distribution is sampled from a symmetric



(a) Statement generation task



(b) Concept Quiz

Figure 3: Shapes data: Mechanical Turk tasks for (a) collecting concept descriptions, and (b) human evaluation from concept descriptions

Dirichlet distribution, and varying the concentration parameter α allows tuning the noise level of the generated datasets (quantified via their Bayes Optimal accuracy⁴). A dataset is then generated by sampling from these conditional distributions. We sample a total of 50 such datasets, consisting of 100 training and 100 test examples each, where each example is a shape and its assigned label.

For each dataset, we then collected statements from Mechanical Turk workers that describe the concept. The task required turkers to study a sample of shapes presented on the screen for each of the two concept-labels (see Figure 3(a)). They were then asked to write a set of statements that would help others classify these shapes without seeing the data. In total, 30 workers participated in this task, generating a mean of 4.3 statements per dataset.


Email data: [Srivastava et al. \(2017\)](#) provide a dataset of language explanations from human users describing 7 categories of emails, as well as 1030 examples of emails belonging to those categories. While this work uses labeled examples, and focuses

⁴This is the accuracy of a theoretically optimal classifier, which knows the true distribution of the data and labels


<p><u>Shapes:</u> If a shape doesn't have a blue border, it is probably not a selected shape. Selected shapes occasionally have a yellow fill.</p>
<p><u>Emails:</u> Emails that mention the word 'meet' in the subject are usually meeting requests Personal reminders almost always have the same recipient and sender</p>
<p><u>Birds:</u> A specimen that has a striped crown is likely to be a selected bird. Birds in the other category rarely ever have dagger-shaped beaks</p>

Table 3: Examples of explanations for each domain

SELECTED BIRDS
scroll to see more



OTHER BIRDS
scroll to see more



READ FIRST! (read carefully)

Please describe the birds in the **SELECTED** column, in a way that can help other people identify these shapes.

Each sentence should focus on **ONE FEATURE** at a time. For example, only focusing on crown color, primary color or wing pattern.

Please **SELECT FEATURE** in the dropdown box which you are describing in your sentence and use the table below to help you identify names for these features.

DO NOT combine multiple features into a single sentence.

- Bill shape**
curved, dagger, hooked, hooked (seabird), all-purpose, cone
- Size**
very large, large, medium, small, very small
- Shape**
long-legged-like | duck-like | gull-like | hummingbird-like | pigeon-like | tree-climbing-like | hawk-like | sandpiper-like | swallow-like | perching-like
- Tail pattern**
solid | spotted | striped | multi-colored
- Primary color**
blue | brown | grey | yellow | olive | green | black | white | red | buff
- Crown color**
blue | brown | grey | yellow | olive | green | black | white | red | buff
- Wing pattern**
solid, spotted, striped, multi-colored

Add another statement

1	Primary color	all selected birds have a brown primary color
2	- what feature? -	
3	- what feature? -	
4	- what feature? -	

Figure 4: Statement generation task for Birds data

on mapping natural language explanations (~30 explanations per email category) to compositional feature functions, we can also use statements in their data for evaluating our approach. While language quantifiers were not studied in the original work, we found about a third of the statements in this data to mention a quantifier.

Birds data: The CUB-200 dataset (Wah et al., 2011) contains images of birds annotated with observable attributes such as size, primary color, wing-patterns, etc. We selected a subset of the data consisting of 10 species of birds and 53 attributes (60 examples per species). Turkers were shown examples of birds from a species, and negative examples consisting of a mix of birds from other

Approach	Avg Accuracy	Labels	Descriptions
LNQ	0.751	no	yes
Bayes Optimal	0.831	-	-
FLGE+	0.659	no	yes
FLGE	0.598	no	yes
LR	0.737	yes	no
Random	0.524	-	-
Ablation:			
LNQ (coarse quant)	0.679	no	yes
LNQ (no quant)	0.545	no	yes
Human:			
Human teacher	0.802	yes	writes
Human learner	0.734	no	yes

Table 4: Classification performance on Shapes datasets (averaged over 50 classification tasks).

species, and were asked to describe the classes (similar to the Shapes data, see Figure 4). During the task, users also had access to a table enumerating groundable attributes they could refer to. In all, 60 workers participated, generating 6.1 statements on average.

5 Experiments

Incorporating constraints from language has not been addressed before, and hence previous approaches for learning from limited data such as Mann and McCallum (2010); Chang et al. (2007b) would not directly work for this setting. Our baselines hence consist of extended versions of previous approaches that incorporate output from the parser, as well as fully supervised classifiers trained from a small number of labeled examples.

Classification performance: The top section in Table 4 summarizes performance of various classifiers on the Shape datasets, averaged over all 50 classification tasks. FLGE+ refers to a baseline

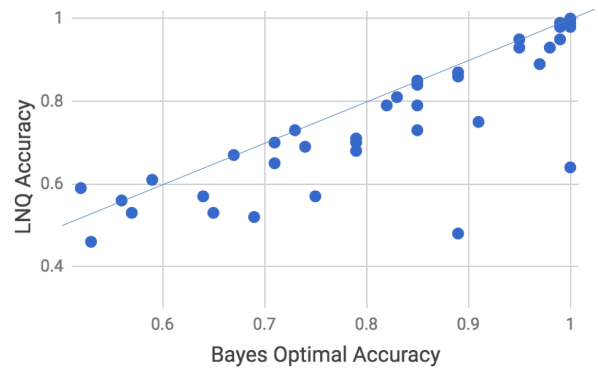


Figure 5: LNQ vs Bayes Optimal Classifier performance for Shape datasets. Each dot represents a dataset generated from a known distribution.

that uses the Feature Labeling through Generalized Expectation criterion, following the approach in Druck et al. (2008); Mann and McCallum (2010). The approach is based on labeling features are indicating specific class-labels, which corresponds to specifying constraints of type $P(y|x)$ ⁵. While the original approach (Druck et al., 2008) sets this value to 0.9, we provide the method the quantitative probabilities used by LNQ. Since the original method cannot handle language descriptions, we also provide the approach the concept label y and feature x as identified by the parser. FLGE represents the version that is not provided quantifier probabilities. LR refers to a supervised logistic regression model trained on $n = 8$ randomly chosen labeled instances.⁶ We note that LNQ performs substantially better than both FLGE+ and LR on average. This validates our modeling principle for learning classifiers from explanations alone, and also suggests value in our PR-based formulation, which can handle multiple constraint types. We further note that not using quantifier probabilities significantly deteriorates FLGE’s performance.

Figure 5 provides a more detailed characterization of LNQ’s performance. Each blue dot represents performance on a shape classification task. The horizontal axis represents the accuracy of the Bayes Optimal classifier, and the vertical represents accuracy of the LNQ approach. The blue line represents the trajectory for $x = y$, representing a perfect statistical classifier in the asymptotic case of infinite samples. We note that LNQ is effective in learning competent classifiers for all levels of hardness. Secondly, except for a small number of outliers, the approach works especially well for learning easy concepts (towards the right). From an error-analysis, we found that a majority of these errors are due to problems in parsing (e.g., missed negation, incorrect constraint type) or due to poor explanations from the teacher (bad grammar, or simply incorrect information).

Figure 6 shows results for email classification tasks. In the figure, LN* refers to the approach in Srivastava et al. (2017), which uses natural language descriptions to define compositional features for email classification, but does not incorporate

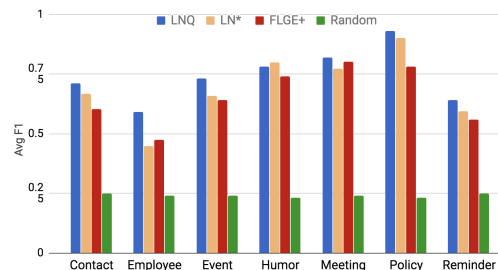


Figure 6: Classification performance (F1) on Email data. (LN* Results from Srivastava et al. (2017))

supervision from quantification. For this task, we found very few of the natural language descriptions to contain quantifiers for some of the individual email categories, making a direct comparison impractical. Thus in this case, we evaluate methods by combining supervision from descriptions in addition to 10 labeled examples (also in line with evaluation in the original paper). We note that additionally incorporating quantification (LNQ) consistently improves classification performance across email categories. On this task, LNQ improves upon FLGE+ and LN* for 6 of the 7 email categories.

Figure 7 shows classification results on the Birds data. Here, LR refers to a logistic regression model trained on $n=10$ examples. The trends in this case are similar, where LNQ consistently outperforms FLGE+, and is competitive with LR.

Ablating quantification: From Table 4, we further observe that the differential associative strengths of linguistic quantifiers are crucial for our method’s classification performance. LNQ (no quant) refers to a variant that assigns the same probability value (average of values in Table 1), irrespective of quantifier. This yields a near random performance, which is what we’d expect if the learning is being driven by the differential strengths of quantifiers. LNQ (coarse quant) refers to a variant that rounds assigned quantifier probabilities in Table 1 to 0 or 1. (i.e., quantifiers such as *rarely* get mapped to 0, while *always* gets mapped to a probability of 1). While its performance (0.679) suggests that simple binary feedback is a substantial signal, the difference from the full model indicates value in using soft probabilities. On the other hand, in a sensitivity study, we found the performance of the approach to be robust to small changes in the probability values of quantifiers.

Comparison with human performance: For the Shapes data, we evaluated human teachers’ own understanding of concepts they teach by evaluating

⁵In general, Generalized Expectation can also handle broader constraint types, similar to Posterior Regularization

⁶LNQ models are indistinct from LR w.r.t. parametrization, but trained to maximize a different objective. The choice of n here is arbitrary, but is roughly twice the number of explanations for each task in this domain

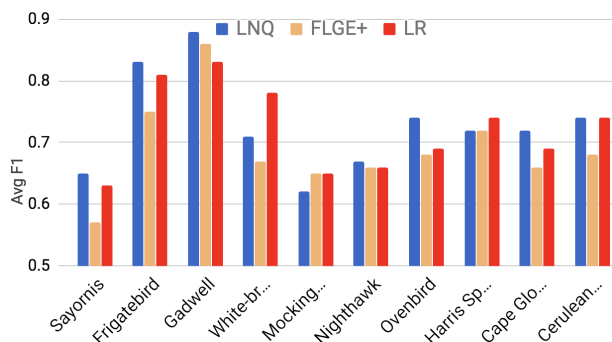


Figure 7: Classification performance on Birds data

them on a quiz based on predicting labels for examples from the test set (see Figure 3(b)). Second, we solicit additional workers that were not exposed to examples from the dataset, and present them only with the statements describing that data (created by a teacher), which is comparable supervision to what LNQ receives. We then evaluate their performance at the same task. From Table 4, we note that a human teacher’s average performance is significantly worse ($p < 0.05$, Wilcoxon signed-rank test) than the Bayes Optimal classifier indicating that the teacher’s own synthesis of concepts is noisy. The human learner performance is expectedly lower, but interestingly is also significantly worse than LNQ. While this might be potentially be caused by factors such as user fatigue, this might also suggest that automated methods can be better at reasoning with constraints than humans in certain scenarios. These results need to be validated through comprehensive experiments in more domains.

Empirical semantics of quantifiers: We can estimate the distributions of probability values for different quantifiers from our labeled data. For this, we aggregate sentences mentioning a quantifier, and calculate the empirical value of the (conditional) probability associated with the statement, leading to a set of probability values for each quantifier. Figure 8 shows empirical distributions of probability values for six quantifiers. We note that while a few estimates (e.g., ‘rarely’ and ‘often’) roughly align with pre-registered beliefs, others are somewhat off (e.g., ‘likely’ shows a much higher value), and yet others (e.g., ‘sometimes’) show a large spread of values to be meaningfully modeled as point values. LNQ’s performance, in spite of this, shows strong stability in the approach. We don’t use these empirical probabilities in experiments, (instead of pre-registered values), so as not to tune the hyperparameters to a specific dataset.

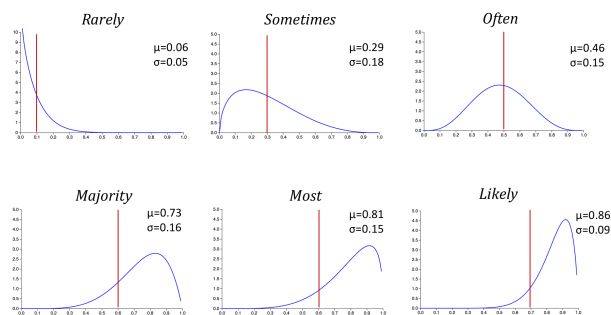


Figure 8: Empirical probability distributions for six quantifiers (Shapes data). Plots show Beta distributions with Method-of-Moment estimates. Red bars correspond to values from Table 1

Such estimates would not be available for a new task without labeled data. Further, using labeled data for estimating these probabilities, and then using the learned model for predicting labels would constitute overfitting, biasing evaluation.

6 Discussion and Future Work

Our approach is surprisingly effective in learning from free-form language. However, it does not address linguistic issues such as modifiers (e.g., *very likely*), nested quantification, etc. On the other hand, we found no instances of nested quantification in the data, suggesting that people might be primed to use simpler language when teaching. While we approximate quantifier semantics as absolute probability values, they may vary significantly based on the context, as shown by cognitive studies such as [Newstead and Collis \(1987\)](#). Future work can model how these parameters can be adapted in a task specific way (e.g., cases such as cancer prediction where base rates are small), and provide better models of quantifier semantics. e.g., as distributions, rather than point values.

Our approach is a step towards the idea of using language to guide learning of statistical models. This is an exciting direction, which contrasts with the predominant theme of using statistical learning methods to advance the field of NLP. We believe that language may have as much to help learning, as statistical learning has helped NLP.

Acknowledgments

This research was supported by the CMU - Yahoo! InMind project. The authors would also like to thank the anonymous reviewers for helpful comments and suggestions.

References

- Jacob Andreas, Dan Klein, and Sergey Levine. 2017. Learning with latent language. *CoRR* abs/1711.00482. <http://arxiv.org/abs/1711.00482>.
- Elke Bach, Eloise Jelinek, Angelika Kratzer, and Barbara BH Partee. 2013. *Quantification in natural languages*, volume 54. Springer Science & Business Media.
- Jon Barwise and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and philosophy* 4(2):159–219.
- Kedar Bellare, Gregory Druck, and Andrew McCallum. 2009. Alternating projections for learning with expectation constraints. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, pages 43–50.
- SRK Branavan, David Silver, and Regina Barzilay. 2012. Learning to win by reading manuals in a monte-carlo framework. *Journal of Artificial Intelligence Research* 43:661–704.
- Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2007a. Guiding semi-supervision with constraint-driven learning. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, Prague, Czech Republic, pages 280–287. <http://www.aclweb.org/anthology/P07-1036>.
- Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2007b. Guiding semi-supervision with constraint-driven learning. In *ACL*. pages 280–287.
- Gregory Druck, Gideon Mann, and Andrew McCallum. 2008. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 595–602.
- Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. 2013. Write a classifier: Zero-shot learning using purely textual descriptions. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research* 11(Jul):2001–2049.
- Siddharth Karamcheti, Edward C Williams, Dilip Arumugam, Mina Rhee, Nakul Gopalan, Lawson LS Wong, and Stefanie Tellex. 2017. A tale of two draggns: A hybrid approach for interpreting action-oriented and goal-oriented instructions. *arXiv preprint arXiv:1707.08668*.
- Angelika Kimmig, Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2012. A short introduction to probabilistic soft logic. In *NIPS Workshop on Probabilistic Programming: Foundations and Applications*.
- Howard S Kurtzman and Maryellen C MacDonald. 1993. Resolution of quantifier scope ambiguities. *Cognition* 48(3):243–279.
- Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science* 350(6266):1332–1338.
- Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(3):453–465.
- Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, and Ruslan Salakhutdinov. 2015. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Percy Liang, Michael I Jordan, and Dan Klein. 2009. Learning from measurements in exponential families. In *Proceedings of the 26th annual international conference on machine learning*. ACM, pages 641–648.
- Percy Liang, Michael I Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pages 590–599.
- M. Lichman. 2013. *UCI machine learning repository*. <http://archive.ics.uci.edu/ml>.
- Sebastian Löbner. 1987. Quantification as a major module of natural language semantics. *Studies in discourse representation theory and the theory of generalized quantifiers* 8:53.
- Gideon S Mann and Andrew McCallum. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of machine learning research* 11(Feb):955–984.
- Linda M Moxey and Anthony J Sanford. 1993. Prior expectation and the interpretation of natural language quantifiers. *European Journal of Cognitive Psychology* 5(1):73–91.
- Stephen E Newstead and Janet M Collis. 1987. Context and the interpretation of quantifiers of frequency. *Ergonomics* 30(10):1447–1462.
- Shashank Srivastava, Igor Labutov, and Tom Mitchell. 2017. Joint concept learning and semantic parsing from natural language explanations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1528–1537. <http://aclweb.org/anthology/D17-1161>.
- C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical report.

Ilker Yildirim, Judith Degen, Michael K Tanenhaus, and T Florian Jaeger. 2013. Linguistic variability and adaptation in quantifier meanings. In *CogSci*.

Luke S Zettlemoyer and Michael Collins. 2007. On-line learning of relaxed ccg grammars for parsing to logical form. In *EMNLP-CoNLL*. pages 678–687.