Sparse, Contextually Informed Models for Irony Detection: Exploiting User Communities, Entities and Sentiment

Byron C. Wallace
University of Texas at Austin
byron.wallace@utexas.edu

Do Kook Choe and **Eugene Charniak**
Brown University
{dc65, ec}@cs.brown.edu

Abstract

Automatically detecting verbal irony (roughly, sarcasm) in online content is important for many practical applications (e.g., sentiment detection), but it is difficult. Previous approaches have relied predominantly on signal gleaned from word counts and grammatical cues. But such approaches fail to exploit the *context* in which comments are embedded. We thus propose a novel strategy for verbal irony classification that exploits contextual features, specifically by combining noun phrases and sentiment extracted from comments with the forum type (e.g., conservative or liberal) to which they were posted. We show that this approach improves verbal irony classification performance. Furthermore, because this method generates a very large feature space (and we expect predictive contextual features to be strong but few), we propose a mixed regularization strategy that places a sparsity-inducing ℓ_1 penalty on the contextual feature weights on top of the ℓ_2 penalty applied to all model coefficients. This increases model sparsity and reduces the variance of model performance.

1 Introduction and Motivation

Automated verbal irony detection is a challenging problem.[1] But recognizing when an author has intended a statement ironically is practically important for many text classification tasks (e.g., sentiment detection).

Previous models for irony detection (Tsur et al., 2010; Lukin and Walker, 2013; Riloff et al., 2013) have relied predominantly on features *intrinsic* to the texts to be classified. By contrast, here we propose exploiting *contextualizing* information, which is often available for web-based classification tasks. More specifically, we exploit signal gleaned from the conversational *threads* to which comments belong. Our approach capitalizes on the intuition that members of different user communities are likely to be sarcastic about different things. As a proxy for user community, we leverage knowledge of the specific forums to which comments were posted. For example, one may surmise that the statement 'I really am proud of Obama' is likely to have been intended ironically if it was posted to a forum frequented by political conservatives. But if this same utterance were posted to a liberal-leaning forum, it is more likely to have been intended in earnest. This sort of information is often directly or indirectly available on social media, but previous models have not capitalized on it. This is problematic; recent work has shown that humans require such contextualizing information to infer ironic intent (Wallace et

Figure 1: A reddit comment illustrating contextualizing features that we propose leveraging to improve classification. Here the highlighted entities (external the comment text itself) provide contextual signals indicating that the shown comment was intended ironically. As we shall see, *Obamacare* is in general a strong indicator of irony when present in posts to the *conservative* subreddit, but less so in posts to the *progressive* subreddit.

[1] In this paper we will be a bit cavalier in using the terms 'verbal irony' and 'sarcasm' interchangeably. We recognize that the latter is a special type of the former, the definition of which is difficult to pin down precisely.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 1035–1044,
Beijing, China, July 26-31, 2015.

al., 2014).

As a concrete example, we consider the task of identifying verbal irony in comments posted to reddit (`http://www.reddit.com`), a social-news website. Users post content (e.g., links to news stories) to reddit, which are then voted on by the community. Users may also discuss this content on the website; these are the comments that we will work with here. Reddit comprises many *subreddits*, which are user communities centered around specific topics of interest. In this work we consider comments posted to two pairs of polarized user communities, or subreddits: (1) *progressive* and *conservative* subreddits (comprising individuals on the left and right of the US political spectrum, respectively), and (2) *atheism* and *Christianity* subreddits.

Our aim is to develop a model that can recognize verbal irony in comments posted to such forums, e.g., automatically discern that the user who posted the comment shown in Figure 1 intended his or her comment ironically. To this end, we propose a strategy that capitalizes on available contextualizing information, such as interactions between the user community (subreddit) that comments were posted to, extracted entities (here we use noun phrases, or NNPs) and inferred sentiment.

The contributions of this work are summarized as follows.

- We demonstrate that contextual information, such as inferred user-community (in this case, the subreddit) can be crossed with extracted entities and sentiment to improve detection of verbal irony. This improves performance over baseline models (including those that exploit inferred sentiment, but not context).
- We introduce a novel composite regularization strategy that applies a sparsifying ℓ_1 penalty to the contextual/sentiment/entity feature weights in addition to the standard squared ℓ_2 penalty to all feature weights. This induces more compact, interpretable models that exhibit lower variance.

While discerning ironic comments on reddit is our immediate task, the proposed approach is generally applicable to a wide-range of subjective, web-based text classification tasks. Indeed, this approach would be useful for any scenario in which we expect different groups of individuals producing content to tend to discuss different entities in a way that correlates with the target categorization. The key is in identifying an available proxy for user groupings (here we rely on the subreddits to which a comment was posted). Such information is often available (or can be derived) for comments posted to different mediums on the web: for example on Twitter we know who a user follows; and on YouTube we know the channels to which videos belong.

2 Exploiting context

2.1 Communities and sentiment

As discussed above, a shortcoming with existing models for detecting sarcasm/verbal irony on the web is their failure to capitalize on contextualizing information. But such information is critical to discerning irony. A large body of work on the use and interpretation of verbal irony supports this supposition (Grice, 1975; Clark and Gerrig, 1984; Wallace, 2013; Wallace et al., 2014). Individuals will be more likely, in general, to use sarcasm when discussing specific entities. Which entities will depend in part on the community to which the individual belongs. As a proxy for user community, here we leverage the subreddits to which comments were posted.

Sentiment may also play an important role. In general, verbal irony is almost always used to convey negative views via ostensibly positive utterances (Sperber and Wilson, 1981). And recent work (Riloff et al., 2013) has exploited features based on sentiment to improve irony detection.

To summarize: when assuming an ironic voice we expect that individuals will convey ostensibly positive sentiment about entities, and that these entities will depend on the type of individual in question. We propose capitalizing on such information by introducing features that encode subreddits, sentiment and noun phrases (NNPs), as we describe next.

2.2 Features

We leverage the feature sets enumerated in Table 1. Subreddits are observed variables. Noun phrase (NNP) extraction and sentiment inference are performed automatically via state of the art NLP tools. In particular, we use the Stanford Sentiment Analysis tool (Socher et al., 2013) to infer sentiment. To extract NNPs we use the Stanford

Feature	Description
Sentiment	The inferred sentiment (*negative/neutral* or *positive*) for a given comment.
Subreddit	the subreddit (e.g., *progressive* or *conservative*; *atheism* or *Christianity*) to which a comment was posted.
NNP	Noun phrases (e.g., proper nouns) extracted from comment texts.
NNP+	Noun phrases extracted from comment texts *and* the thread to which they belong (for example, 'Obamacare' from the title in Figure 1).

Table 1: Feature types that we exploit. We view the (observed) subreddit as a proxy for *user type*. We combine this with sentiment and extracted noun phrases (NNPs) to improve classifier performance.

Part of Speech tagger (Toutanova et al., 2003). We then introduce 'bag-of-NNP' features and features that indicate whether the sentiment inferred for a given sentence was positive or not.

Additionally, we introduce 'interaction' features that capture combinations of these. For example, a feature that indicates whether a given sentence mentions Obamacare (which will be one of many NNPs automatically extracted) *and* was posted in the *conservative* subreddit. This is an example of a two-way interaction. We also experiment with three-way interactions, crossing sentiment with NNPs and subreddits. An example is a feature that indicates if a sentence was: inferred to be positive *and* mentions Obamacare (NNP) *and* was part of a comment made in the conservative subreddit. Finally, we experiment with adding NNPs extracted from the comment thread in addition to the comment text.

These are rich features that capture signal not directly available from the sentences themselves. Features that encode subreddits crossed with extracted NNP's, in particular, offer a chance to explicitly account for differences in how the ironic device is used by individuals in different communities. However, this has the downside of introducing a large number of irrelevant terms into the model: we expect, *a priori*, that many entities will not correlate with the use of verbal irony. We would therefore expect this strategy to exhibit high variance in terms of predictive performance, and we later confirm this empirically. Ideally, a model would perform feature selection during parameter estimation, thus dropping irrelevant interaction terms. We next introduce a composite ℓ_1/ℓ_2 regularization strategy toward this end.

3 Enforcing sparsity

3.1 Preliminaries

In this work we consider linear models with binary outputs ($y \in \{-1, +1\}$). We will assume we have access to a training dataset comprising n instances, $\mathbf{x} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ and associated labels $\mathbf{y} = \{y_1, ..., y_n\}$. We then aim to find a weight-vector $\mathbf{w}$ that optimizes the following objective.

$$\underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \mathcal{L}(\text{sign}\{\mathbf{w} \cdot \mathbf{x}_i\}, y_i) + \alpha \mathcal{R}(\mathbf{w}) \quad (1)$$

Where $\mathcal{L}$ is a loss function, $\mathcal{R}(\mathbf{w})$ is a regularization term and α is a parameter expressing the relative emphasis placed on achieving minimum empirical loss versus producing a simple model (i.e., a weight vector with small weights). Typically one searches for a good α using the available training data. For $\mathcal{L}$, we will use the log-loss in this work, though other loss functions may be used in its place.

3.2 Sparsity via Regularization

Concerning $\mathcal{R}$, one popular regularization function is the squared ℓ_2 norm:

$$\sum_j \mathbf{w}_j^2 \quad (2)$$

This is the norm used in the standard Support Vector Machine (SVM) formulation, for example, and has been shown empirically to work well for text classification (Joachims, 1998). An alternative is to use the ℓ_1 norm:

$$\sum_j |\mathbf{w}_j| \quad (3)$$

Which has the advantage of inducing sparse models: i.e., using the ℓ_1 norm as a penalty tends to drive feature weights to 0.

Returning to the present task of detecting verbal irony in comments, it seems reasonable to assume that there will be a relatively small set of entities that correlate with sarcasm. But because we are introducing 'interaction' features that enumerate the cross-product of subreddits and entities (and, in some cases, sentiment), we have a large feature-space. This space includes features that correspond to NNPs extracted from, and sentiment inferred for, the sentence itself: we will denote the indices for these by $\mathcal{I}$. Other interaction features

correspond to entities extracted from the *threads* associated with comments: we denote the corresponding set of indices by $\mathcal{T}$. We expect only a fraction of the features comprising both $\mathcal{I}$ and $\mathcal{T}$ to have non-zero weights (i.e., to signal ironic intent).

This scenario is prone to the undesirable property of high-variance, and hence calls for stronger regularization. But in general replacing the squared ℓ_2 norm with an ℓ_1 penalty (over all weights) hampers classification performance (indeed, as we later report, this strategy performs very poorly here). Therefore, in our scenario we would like to place a sparsifying ℓ_1 regularizer over the contextual (interaction) features while still leveraging the squared ℓ_2-norm penalty for the standard bag-of-words (BoW) features.[2] We thus propose the following composite penalty:

$$\sum_j \mathbf{w}_j^2 + \sum_{k \in \mathcal{I}} |\mathbf{w}_k| + \sum_{l \in \mathcal{T}} |\mathbf{w}_l| \quad (4)$$

The idea is that this will drive many of the weights associated with the contextual features to zero, which is desirable in light of the intuition that a relatively small number of entities will likely indicate sarcasm. At the same time, this composite penalty applies only the squared ℓ_2 norm to the standard BoW features, given the comparatively strong predictive performance realized with this strategy.

Putting this together, we modify the original objective (Equation 1) as follows:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \mathcal{L}(\text{sign}\{\mathbf{w} \cdot \mathbf{x}_i\}, y_i) + \alpha_0 \sum_j \mathbf{w}_j^2 + \alpha_1 \sum_{k \in \mathcal{I}} |\mathbf{w}_k| + \alpha_2 \sum_{l \in \mathcal{T}} |\mathbf{w}_l| \quad (5)$$

Where we have placed separate α scalars on the respective penalty terms. Note that this is similar to the *elastic net* (Zou and Hastie, 2005) joint regularization and variable selection strategy. The distinction here is that we only apply the ℓ_1 penalty to (i.e., perform feature selection for) the subset of 'interaction' feature weights, which is in contrast to the elastic net, which imposes the composite penalty to *all* feature weights. One can view this as using the regularizer to encourage a sparsity pattern specific to the task at hand.

[2]Note that we apply both ℓ_1 and ℓ_2 penalties to the features in $\mathcal{I}$ and $\mathcal{T}$.

3.3 Inference

We fit this model via Stochastic Gradient Descent (SGD).[3] During each update, we impose both the squared ℓ_2 and ℓ_1 penalties; the latter is applied only to the contextual/interaction features in $\mathcal{I}$ and $\mathcal{T}$. For the ℓ_1 penalty, we adopt the cumulative truncated gradient method proposed by Tsuruoka et al. (2009).

4 Experimental Setup

4.1 Datasets

For our development dataset, we used a subset of the reddit irony corpus (Wallace et al., 2014) comprising annotated comments from the *progressive* and *conservative* subreddits. We also report results from experiments performed using a separate, held-out portion of this data, which we did not use during model refinement. Furthermore, we later present results on comments from the *atheism* and *Christianity* subreddits (we did not use this data during model development, either).

The development dataset includes 1,825 annotated comments (876 and 949 from the *progressive* and *conservative* subreddits, respectively). These comprise 5,625 sentences in total, each of which was independently labeled by three annotators as having been intended *ironically* or not. For additional details on the annotation process, see (Wallace et al., 2014). For simplicity, we consider a sentence to be 'ironic' ($y = 1$) when at least two of the three annotators designated it as such, and 'unironic' ($y = -1$) otherwise. Using this criteria, 286 (5%) of the labeled sentences are labeled 'ironic'.

The test portion of the political dataset comprises 996 annotated comments (409 *progressive* and 587 *conservative* comments), totalling 2,884 sentences. Using the same criteria as above – at least 2/3 annotators labeling a given sentence as 'ironic' – we have 154 'ironic' sentences (again about 5%).

The 'religion' dataset (comments from *atheism* and *Christianity*) contains 1,682 labeled comments comprising 5615 sentences (2,966 and 2,649 from the atheism and Christian subreddits, respectively); 313 ($\sim$6%) were deemed 'ironic'.

[3]We have implemented this within the *sklearn* package (Pedregosa et al., 2011).

4.2 Experimental Details

We recorded results from 500 independently performed experiments on random train (80%)/test (20%) splits of the data. These splits were performed at the *comment* (rather than sentence) level, so as not to test on sentences belonging to comments encountered in the training set. We measured performance, however, at the sentence level (often only a single sentence in a given comment will have been labeled as 'ironic').

Our baseline approach is a standard squared-ℓ_2 regularized log-loss linear model (fit via SGD) that leverages uni- and bi-grams and features indicating grammatical cues, such as exclamation points and emoticons. We also experiment with a model that includes inferred sentiment indicators, but not context. We performed standard English stopwording, and we used Term Frequency Inverse-Document Frequency (TF-IDF) feature weighting. For the gradient descent procedure, we used a decaying learning rate (specifically, $\frac{1}{t}$, where t is the update count). We performed a coarse grid search to find values for α that maximize $F1$ on the training datasets. We took five full passes over the training data before terminating descent.

We report paired *recalls* and *precisions*, as observed on each random train/test split of the data. The former is defined as $\frac{TP}{TP+FN}$ and the latter as $\frac{TP}{TP+FP}$, where TP denotes the true positive count, FN the number of false negatives and FP the false positive count. We report these separately - rather than collapsing into $F1$ - because it is not clear that one would value recall and precision equally for irony detection, and because this allows us to tease out *how* the models differ in performance. Notably, for example, sentiment and context features both improve recall, but the latter does so without harming precision.

5 Results

5.1 Results on the Development Corpus

Figure 2 and Table 2 summarize the performance of the different approaches over 500 independently performed train/test splits of the political development corpus. For reference, a random chance strategy (which predicts 'ironic' with probability equal to the observed prevalence) achieves a median recall of 0.048 and a median precision of 0.047.

Figure 2 shows histograms of the observed absolute differences between the baseline linear classifier and the proposed augmentations. Adding the proposed features (which capitalize on sentiment and NNP-mentions on specific subreddits) increases absolute median recall by 3.4 percentage points (a relative gain of $\sim$12%). And this is achieved without sacrificing precision (in contrast to exploiting only sentiment). Furthermore, as we can see in Figures 2 and 3, the proposed regularization strategy shrinks the variance of the classifier. This variance reduction is achieved through greater model sparsity, as can be seen in Figure 4, which improves interpretability. We note that leveraging *only* an ℓ_1 regularization penalty (with the full feature-set) results in very poor performance (median recall and precision of 0.05 and 0.09, respectively). Similarly, the elastic-net strategy (Zou and Hastie, 2005) (in which we do not specify which features to apply the ℓ_1 penalty to), here achieves a median recall of 0.11 and a median precision of 0.07.

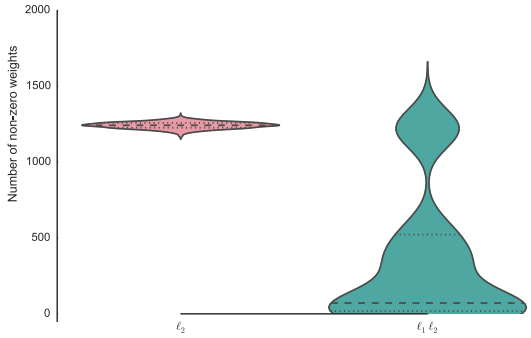

Figure 4: Empirical distributions (violin plots) of non-zero feature counts in the NNP $\times$ subreddit model (rows 3 and 4 in Figure 3) using standard ℓ_2-norm (left) and the proposed $\ell_1\ell_2$-norm (right) regularization approaches on the *atheism*/*Christianity* data over 500 independent train/test splits. The composite norm achieves much greater sparsity, resulting in lower variance. This sparsity also (arguably) provides greater interpretability; one can inspect contextual features with non-zero weights.

5.2 Results on the Held-out (Test) Corpus

Table 4 reports results on the held-out political test dataset, achieved after training the models on the entirety of the development corpus. To account for the variance inherent to inference via SGD, we performed 100 runs of the SGD procedure and report median results from these runs. These results mostly agree with those reported for the development corpus: the proposed strategy improves median recall on the held-out corpus by nearly 4.0 percentage points, at a median cost of about 1 point in precision. By contrast, sentiment alone provides a 2% absolute improvement in recall at

	mean; median (25th, 75th)	mean; median (25th, 75th)
baseline (BoW)	0.288; 0.283 (0.231, 0.333)	0.129; 0.124 (0.103, 0.149)
	Δ recall	Δ precision
(overall) sent.	+0.036; +0.037 (+0.015, +0.063)	-0.008; -0.007 (-0.018, +0.003)
NNP	+0.021; +0.018 (+0.000, +0.036)	-0.008; -0.008 (-0.016, -0.001)
NNP × subreddit	+0.013; +0.016 (+0.000, +0.031)	-0.002; -0.003 (-0.009, +0.004)
NNP × subreddit (ℓ_1 ℓ_2)	+0.010; +0.000 (+0.000, +0.021)	-0.002; -0.002 (-0.007, +0.004)
NNP+ × sent. × subreddit + sent.	+0.036; +0.038 (+0.000, +0.065)	-0.000; -0.001 (-0.012, +0.011)
NNP+ × sent. × subreddit + sent. (ℓ_1 ℓ_2)	+0.035; +0.034 (+0.000, +0.062)	+0.001; +0.000 (-0.011, +0.011)

Table 2: Summary results over 500 random train/test splits of the development dataset. The top row reports mean and median baseline (BoW) recall and precision and lower and upper (25th and 75th) percentiles. We report pairwise differences w.r.t. this baseline in terms of recall and precision for each strategy. Exploiting NNP features and subreddits improves recall with little to not cost in precision. Capitalizing on sentiment alone improves recall but at a greater cost in precision. The proposed $\ell_1\ell_2$ regularization strategy achieves comparable performance with fewer features, and shrinks the variance over different train/test splits (as can bee seen in Figure 2).

	mean; median (25th, 75th)	mean; median (25th, 75th)
baseline (BoW)	0.281; 0.268 (0.222, 0.327)	0.189; 0.187 (0.144, 0.230)
	Δ recall	Δ precision
(overall) sent.	+0.001; +0.000 (-0.011, +0.015)	-0.014; -0.012 (-0.023, -0.002)
NNP	+0.018; +0.018 (+0.000, +0.039)	-0.009; -0.010 (-0.021, +0.001)
NNP × subreddit	+0.024; +0.025 (+0.000, +0.046)	+0.002; +0.001 (-0.011, +0.013)
NNP × subreddit (ℓ_1 ℓ_2)	+0.013; +0.015 (+0.000, +0.033)	+0.002; +0.002 (-0.009, +0.011)
NNP+ × sent. × subreddit + sent.	+0.023; +0.024 (+0.000, +0.046)	+0.001; +0.001 (-0.012, +0.013)
NNP+ × sent. × subreddit + sent. (ℓ_1 ℓ_2)	+0.014; +0.015 (+0.000, +0.036)	-0.008; -0.008 (-0.021, +0.004)

Table 3: Results on the *atheism* and *Christianity* subreddits. In general sentiment does not help on this dataset (see row 1). But the NNP and subreddit features again consistently improve recall without hurting precision. And, as above, $\ell_1\ell_2$ regularization shrinks variance (see Figures 2 and 3).

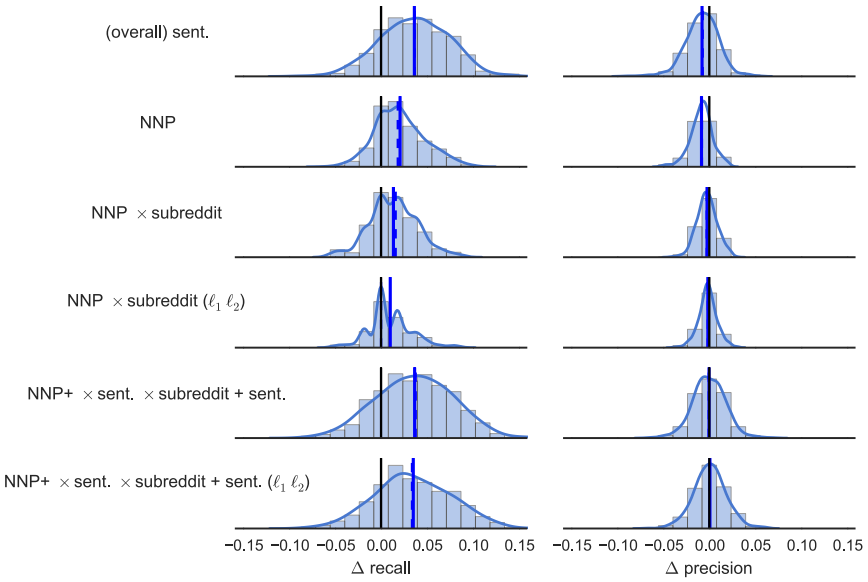

Figure 2: Results from 500 independent train/test splits of the development subset of our political data. Shown are histograms with smoothed kernel density estimates of differences in recall and precision between the baseline bag-of-words based approach and each feature space/method (one per row). The solid black line at 0 indicates no difference; solid and dotted blue lines demarcate means and medians, respectively. Features are as in Table 1. The × symbol denotes interactions; + indicates addition. The proposed contextual features substantially improve recall, with little to no loss in precision. Moreover, in general, the $\ell_1\ell_2$ regularization approach reduces variance. (We note that in constructing histograms we have excluded a handful of points – never more than 1% – where the difference exceeded 0.15).

	median recall (std. dev.)	median precision (std. dev.)
baseline	0.331 (0.146)	0.148 (0.022)
(overall) sent.	0.351 (0.054)	0.125 (0.003)
NNP	0.364 (0.119)	0.135 (0.021)
NNP × subreddit	0.357 (0.108)	0.143 (0.020)
NNP+ × sent. × subreddit	0.344 (0.116)	0.142 (0.019)
NNP+ × sent. × subreddit (ℓ_1 ℓ_2)	0.325 (0.052)	0.141 (0.008)
NNP+ × sent. × subreddit + sent.	0.377 (0.104)	0.141 (0.014)
NNP+ × sent. × subreddit + sent. (ℓ_1 ℓ_2)	0.370 (0.056)	0.140 (0.008)

Table 4: Results on the held-out political dataset, using the entire development corpus as a training set. Abbreviations are as described in the caption for Figure 2. Due to the variance inherent to the stochastic gradient descent procedure, we repeat the experiment 100 times and report the median performance and standard deviations (of different SGD runs). Results are consistent with those reported for the development corpus.

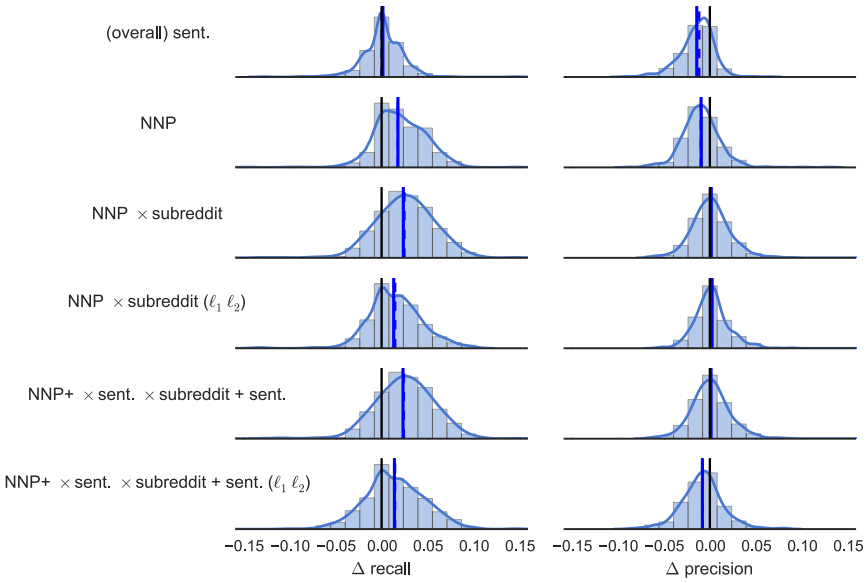

Figure 3: Results from 500 independent train/test splits of the development subset of the religion corpus). The description is the same as for Figure 2.

the expense of more than 2 points in precision.

5.3 Results on the religion dataset

To assess the general applicability of the proposed approach, we also evaluate the method on comments from a separate pair of polarized communities: *atheism* and *Christianity*, as described in Section 4.1. This dataset was not used during model development. We follow the experimental setup described in Section 4.2.

In this case, capitalizing on the NNP × subreddit features produces a mean 2.3% absolute gain in recall (median: 2.4%) over the baseline approach, with a (very) slight gain in precision. The ℓ_1 ℓ_2 approach achieves a lower expected gain in recall (median: 1.5%), but again shrinks the variance w.r.t. model performance (see Figure 3). Moreover, as we show in Figure 4, this is achieved with a much more compact (sparser) model. We note that for the religion data, inferred sentiment features do not seem to improve performance, in contrast to the results on the political subreddits. At present, we are not sure why this is the case.

These results demonstrate that introducing features that encode entities and user communities (NNPs × subreddit) improve recall for irony detection in comments addressing relatively diverse topics (politics and religion).

5.4 Predictive features

We report the interaction features that are the best predictors of verbal irony in the respective subreddits (for both polar community pairs). Specifically, we estimated the weights for every interaction feature using the entire training dataset, and repeated this process 100 times to account for variation due to the SGD procedure.

progressive		*conservative*	
feature	weight	feature	weight
freedom	0.102 (0.048)	racist	0.148 (0.043)
god	0.085 (0.045)	news	0.100 (0.044)
christmas	0.081 (0.046)	way	0.078 (0.044)
jesus	0.060 (0.038)	obamacare	0.068 (0.041)
kenya	0.052 (0.035)	white	0.059 (0.037)
brave	0.043 (0.035)	let	0.058 (0.038)
bravo	0.041 (0.035)	course	0.046 (0.033)
know	0.038 (0.030)	huh	0.044 (0.036)
dennis	0.038 (0.029)	education	0.043 (0.032)
ronald	0.036 (0.030)	president	0.039 (0.031)

Table 5: Average weights (and standard deviations calculated across samples) for top 10 NNP × subreddit features from the *progressive* and *conservative* subreddits.

Table 5 displays the top 10 NNP × subreddit features for the political subreddits, with respect to the mean magnitude of the weights associated with them. We report these means and the standard deviations calculated across the 100 runs. This table implies, for example, that mentions of 'freedom' and 'kenya' indicate irony in the *progressive* subreddit; while mentions of 'obamacare' and 'president' (for example) in the *conservative* subreddit tend to imply irony.

Table 6 reports analagous results for the religion subreddits. Here we can see, e.g., that 'god' is a good predictor of irony in the *atheism* subreddit, and 'professor' is in the *Christianity* subreddit.

We also report the top ranking 'three-way' interaction features that cross NNP's extracted from

atheism		*Christianity*	
feature	weight	feature	weight
right	0.353 (0.014)	professor	0.297 (0.013)
god	0.324 (0.013)	let	0.084 (0.014)
women	0.214 (0.013)	peter	0.080 (0.019)
christ	0.160 (0.014)	geez	0.054 (0.016)
news	0.146 (0.013)	evil	0.054 (0.015)
trust	0.139 (0.013)	killing	0.053 (0.015)
shit	0.132 (0.015)	liberal	0.049 (0.014)
believe	0.123 (0.013)	antichrist	0.049 (0.014)
great	0.121 (0.016)	rock	0.047 (0.014)
ftfy	0.108 (0.016)	pedophilia	0.046 (0.014)

Table 6: Top 10 NNP × subreddit features from the *atheism* and *Christianity* subreddits.

progressive		*conservative*	
feature	weight	feature	weight
american (+)	0.045 (0.023)	mr (+)	0.041 (0.021)
yay (+)	0.042 (0.022)	cruz (+)	0.040 (0.021)
ollie (+)	0.036 (0.019)	king (+)	0.036 (0.019)
north (+)	0.036 (0.019)	onion (+)	0.035 (0.018)
fuck (+)	0.034 (0.018)	russia (+)	0.034 (0.018)
washington (+)	0.034 (0.018)	oprah (+)	0.030 (0.016)
times* (+)	0.034 (0.018)	science (+)	0.027 (0.015)
world (+)	0.030 (0.016)	math (+)	0.027 (0.015)
magic (+)	0.024 (0.013)	america (+)	0.026 (0.014)
where (+)	0.024 (0.013)	ben (+)	0.020 (0.011)

Table 7: Average weights for top 10 NNP × subreddit × sentiment features. The parenthetical '+' indicates that the inferred sentiment was positive. In general, (ostensibly) positive sentiment indicates irony.

sentences with subreddits and the inferred sentiment for the political corpus (Table 7). This would imply, e.g., that if a sentence in the *progressive* subreddit conveys an ostensibly positive sentiment about the political commentator 'Ollie',[4] then this sentence is likely to have been intended ironically.

Some of these may seem counter-intuitive, such as ostensibly positive sentiment regarding 'Cruz' (as in the conservative senator Ted Cruz) in the conservative subreddit. On inspection of the comments, it would seem Ted Cruz does not find general support even in this community. Example comments include: "Stay classy Ted Cruz" and "Great idea on the talkathon Cruz". The 'mr' and 'king' terms are almost exclusively references to Obama in the *conservative* subreddit. In any case, because these are three-way interaction terms, they are all relatively rare: therefore we would caution against over interpretation here.

6 Related Work

The task of automated irony detection has recently received a great deal of attention from the NLP and ML communities (Tepperman et al., 2006; Davidov et al., 2010; Carvalho et al., 2009; Burfoot and Baldwin, 2009; Tsur et al., 2010; González-Ibáñez et al., 2011; Filatova, 2012; Reyes et al., 2012; Lukin and Walker, 2013; Riloff et al., 2013). This work has mostly focussed on exploiting token-based indicators of verbal irony. For example, it is clear that gratuitous punctuation (e.g. "oh really??!!!") signals irony (Carvalho et al., 2009).

Davidov et al. (2010) proposed a semi-supervised approach in which they look for sentence *templates* indicative of irony. Elsewhere, Riloff et al. (2013) proposed a method that exploits apparently contrasting sentiment in the same utterance to detect irony. While innovative, these approaches still rely on features intrinsic to comments; i.e., they do not attempt to capitalize on contextualizing features external to the comment text. This means that there will necessarily be certain (subtle) ironies that escape detection by such approaches. For example, without any additional information about the speaker, it would be impossible to deduce whether the comment "Obamacare is a great program" is intended sarcastically.

Other related recent work has shown the promise of sparse models, both for prediction and interpretation (Eisenstein et al., 2011a; Eisenstein et al., 2011b; Yogatama and Smith, 2014a). Yogatama (2014a; 2014b), e.g., has leveraged the *group lasso* approach to impose 'structured' sparsity on feature weights. Our work here may similarly be viewed as assuming a specific sparsity pattern (specifically that feature weights for 'interaction features' will be sparse) and expressing this via regularization.

7 Conclusions and Future Directions

We have shown that we can leverage contextualizing information to improve identification of verbal irony in online comments. This is in contrast to previous models, which have relied predominantly on features that are *intrinsic* to the texts to be classified. We exploited features that indicate user communities crossed with sentiment and extracted noun phrases. This led to consistently improved recall with little to no cost in precision. We also proposed a novel composite regularization strategy that imposes a sparsifying ℓ_1 penalty on the interaction features, as we expect most of these to be irrelevant. This reduced performance variance.

Future work will include expanding the corpus and experimenting with datasets outside of the political domain. We also plan to evaluate this strategy on data from different online sources, e.g., Twitter or YouTube.

[4] 'Ollie' is a conservative political commentator.

Acknowledgements

This work was supported by ARO grant W911NF-14-1-0442.

References

C Burfoot and T Baldwin. 2009. Automatic satire detection: are you having a laugh? In *ACL-IJCNLP*, pages 161–164. ACL.

P Carvalho, L Sarmento, MJ Silva, and E de Oliveira. 2009. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In *CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM.

HH Clark and RJ Gerrig. 1984. On the pretense theory of irony. *Journal of Experimental Psychology*, 113:121–126.

D Davidov, O Tsur, and A Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. *Conference on Natural Language Learning (CoNLL)*, page 107.

J Eisenstein, A Ahmed, and EP Xing. 2011a. Sparse additive generative models of text. In *International Conference on Machine Learning (ICML)*.

J Eisenstein, NA Smith, and EP Xing. 2011b. Discovering sociolinguistic associations with structured sparsity. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1365–1374.

E Filatova. 2012. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *LREC*, volume 12, pages 392–398.

R González-Ibáñez, S Muresan, and N Wacholder. 2011. Identifying sarcasm in twitter: a closer look. In *ACL*, volume 2, pages 581–586. Citeseer.

HP Grice. 1975. Logic and conversation. *1975*, pages 41–58.

T Joachims. 1998. *Text categorization with support vector machines: Learning with many relevant features*. Springer.

S Lukin and M Walker. 2013. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. *NAACL*, pages 30–40.

F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

A Reyes, P Rosso, and T Veale. 2012. A multidimensional approach for detecting irony in twitter. *LREC*, pages 1–30.

E Riloff, A Qadir, P Surve, LD Silva, N Gilbert, and R Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *EMNLP*, pages 704–714.

R Socher, A Perelygin, JY Wu, J Chuang, CD Manning, AY Ng, and C Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642. Citeseer.

D Sperber and D Wilson. 1981. Irony and the use-mention distinction. *1981*.

J Tepperman, D Traum, and S Narayanan. 2006. "Yeah Right": Sarcasm Recognition for Spoken Dialogue Systems.

K Toutanova, D Klein, CD Manning, and Y Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

O Tsur, D Davidov, and A Rappoport. 2010. ICWSM-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *AAAI Conference on Weblogs and Social Media*.

Y Tsuruoka, J Tsujii, and S Ananiadou. 2009. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of the Joint Conference of the Annual Meeting of the ACL and the International Joint Conference on Natural Language Processing of the AFNLP*, pages 477–485. Association for Computational Linguistics.

BC Wallace, DK Choe, L Kertz, and E Charniak. 2014. Humans require context to infer ironic intent (so computers probably do, too). Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pages 512–516.

BC Wallace. 2013. Computational irony: A survey and new perspectives. *Artificial Intelligence Review*, pages 1–17.

D Yogatama and NA Smith. 2014a. Linguistic structured sparsity in text categorization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 786–796.

D Yogatama and NA Smith. 2014b. Making the most of bag of words: Sentence regularization with alternating direction method of multipliers. In *Proceedings of The 31st International Conference on Machine Learning*, pages 656–664.

H Zou and T Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.