Semi-supervised Humor Detection with Adversarial Training Methods

Denizalp Goktas (dg2906), Linying Zhang (lz2629), Da Hua Chen (dc2802)
https://github.com/denizalp/Humor-Detection

1 Introduction

Humor is ubiquitous in everyday communication, yet the construction of deep learning models that provide structure to humor, recognize humor and generate humor has proven to be rather challenging [1] [2].

Because the literature on semi-supervised techniques in deep learning is sparse, our project will try to advance the literature by using recent advances in semi-supervised adversarial techniques to the humor detection problem [7]. We believe that a semi-supervised approach may allow the model to become familiar with many different types of humor, while adversarial training methods will allow it to become more resilient to changes in the text so that it can handle subtleties in sentences that may have different meanings.

2 Related Work and References

2.1 Background

From the perspective of artificial intelligence, there have been attempts to formalize and model different structures associated with humor, but one drawback is that such methods often generate a very specific category of humor and are not easily generalizable [3] [4]. Some work has been done to tackle the humor detection using classical machine learning techniques such as Random Forest (collection of decision trees) and K Nearest Neighbors (KNN) [5]. We will focus on deep learning methods. Deep learning papers on humor detection are fairly sparse, and primarily make use of supervised deep learning techniques [1] [2].

2.2 Challenges

One challenge is that we don't know whether our models extend out-of-domain, to a transfer learning setting for example. This is because there is no labeled corpus of funny texts partially due to the difficulty associated with the subjectivity of imposing an accurate binary outcome on something as ill-defined as humor. Some work has been done applying similar methods to Twitter data sets, but no large, labelled data sets exist for thorough validation.

2.3 Innovation Goals

Our paper's primary goal is to improve the generalizability of humor detection models by using adversarial training methods. Adversarial training methods aim to increase the robustness of deep learning models to small perturbations in the model inputs. One can consider the subjectivity introduced by different humor contexts to be causing perturbation to the inputs of the model. As a result, using adversarial training method can allow our model to become less dependent on the context provided by the training dataset.

3 Problem Formulation and Details

3.1 Dataset

We will use the Yelp Dataset Challenge dataset, which consists of 1.6 million reviews by 366,000 users for 61,000 businesses [8]. Each review consists of one or more sentences commenting on the business at hand, along with votes given by other users to the review particularly, funny, useful,and cool. We will be considering only the funny category in this project. Since there is so much data, we will only use a small sample for the training and test sets (about 15,000 samples each for training, testing, and unlabeled).

3.1.1 Examples

Here are two positive (labeled funny) examples in the training set:

- Absolutely love how you can create your own stirfry! My favorite flavor was the Spicy Korean Chili! Great food, great atmosphere, and great friendly staff! Bathroom entrance says "Unloading Zone" lmao
- Less than a year and their tires cant even handle a poke lol pathetic ill go back to discount tires, tires i got from them for my other car still great after almost two years. Great sales pitch not a lot of great product or service to back them up.

Here are two negative (not labeled funny) examples:

- Never again. Do not waste your time or money. Not worth any perceived savings. "Priority" is a scam. Rip off from start to finish.
- Delicious, the perfect combination of flavor and spice! Very unique find in the valley! The staff is so friendly and offers samples before you order! When I have visitors in town, this is a must go to!

3.2 Problem Description

The problem is a binary classification task. The input is some review and the output is the label "funny" (positive) or "not funny" (negative). Even though we have a vote total for the number of users who found a particular review funny, we note that there is a lot of variation in the reviews with only one funny vote. By no means a systematic and objective study, we looked at the types of reviews that received one and two funny votes by the collective Yelp user base. In most cases, by visual inspection, we noted that many of these reviews weren't in fact funny (in our opinions). Even among the reviews with more than 2 votes, many were not funny (as judged by us). In this exposition, we tackle the binary classification problem of funny vs. not funny. As we proceed, we make the decision that a review is considered funny if it has three or more funny votes. This may change in the future.

3.3 Evaluation Metric

In Yelps Kaggle competition 1, they suggest a metric for evaluation of a count based target called Root Mean Squared Logarithmic Error (RMSLE). It is defined as:

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(\hat{y}_i+1)-\log(y_i+1))^2} \tag{1}$$

We will use this metric to evaluate the performance of our model since it will allow us to compare our model to existing benchmarks [2]. Note that this function will not be used to optimize our classifier.

4 Methods and Model Architecture

4.1 Model Architecture

For our preliminary results, we use a simple LSTM model that makes use of adversarial methods [6]. Following these preliminary results we will try different architectures involving bidirectional LSTMs and CNNs to improve our accuracy further.

In adversarial training, we train the classifier to become robust to small perturbations in the embeddings of the words. Adversarial training in the context of Natural Language Processing (NLP), has a different meaning than in the context of Computer Vision and is rather used as a tool to regularize the classifier [7]. In the context of humor classification, this means that that adversarial methods can allow our model to become less dependent on the context provided by our data-set and allow increased transfer-ability of the model to other application fields.

To train our model we will use two different adversarial methods: a) a regular adversarial method and b) a virtual-adversarial method. We will compare the errors we get by combining both methods and using each method individually.

The choice of parameters we use to pretrain the model: vocab size=18823, embedding dims=256, rnn cell size=1024, num candidate samples=1024, batch size=256, learning rate=0.001, learning rate decay factor=0.9999, max grad norm=1.0, num timesteps=500, keep prob emb=0.5.

The parameters used to in the actual training are similar to above. vocab size=18823, embedding dims=256, rnn cell size=1024, cl num layers=1, cl hidden size=30, batch size=64, learning rate=0.0005, learning rate decay factor=0.9998, max grad norm=1.0, num timesteps=500, keep prob emb=0.5.

4.2 Loss

Note that similarly to Goodfellow et al. [7], in our experiments, adversarial training of any sort refers to minimizing the negative log-likelihood plus the adversarial loss/losses with gradient descent.

The regular adversarial loss function is defined by:

$$L_{\text{adv}}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N} \log p\left(y_n | \boldsymbol{s}_n + \boldsymbol{r}_{\text{adv},n}; \boldsymbol{\theta}\right) \tag{2}$$

At each training step, we will calculate the following adversarial perturbation:

$$\boldsymbol{r}_{\text{adv}} = -\epsilon \boldsymbol{g}/\|\boldsymbol{g}\|_2 \text{ where } \boldsymbol{g} = \nabla_s \log p(y|\boldsymbol{s}; \hat{\boldsymbol{\theta}}) \tag{3}$$

The virtual-adversarial loss function formulated by Goodfellow et al. is defined as [7]:

$$L_{\text{v adv}}\left(\boldsymbol{\theta}\right) = \frac{1}{N'}\sum_{n'=1}^{N'} \text{KL}\left[p\left(\cdot|\boldsymbol{s}_{n'}; \hat{\boldsymbol{\theta}}\right) \| p\left(\cdot|\boldsymbol{s}_{n'} + \boldsymbol{r}_{\text{v-adv},n'}; \boldsymbol{\theta}\right)\right] \tag{4}$$

where N' is the total number of examples (labeled and unlabeled) and $\text{KL}[p\|q]$ is the KL divergence between distributions p and q.

This function's loss is calculated without having knowledge about y (the sample's label). As a result, virtual adversarial training can be used in a semi-supervised setting. At each step, we calculate the approximate adversarial perturbation and apply it to the word embeddings, following the steps provided by Goodfellow et al. [7]. The adversarial perturbation is approximated by:

$$\boldsymbol{r}_{\text{v -adv}} = \epsilon \boldsymbol{g}/\|\boldsymbol{g}\|_2 \text{ where } \boldsymbol{g} = \nabla_{\boldsymbol{s}+d}\text{KL}[p(\cdot|\boldsymbol{s}; \hat{\boldsymbol{\theta}})\|p(\cdot|\boldsymbol{s} + \boldsymbol{d}; \hat{\boldsymbol{\theta}})] \tag{5}$$

We have opted to use a combination of CNNs and RNNs for our architecture.

5 Preliminary Results

Preliminary results were obtained using 80-20 train-test split on 10,000 total positive and negative examples and 10,000 unlabeled examples. The vocabulary size is 18,823 and the word embedding dimension is 256.

We pretrain on the data to learn the embedding. The loss from 500 steps is shown in Figure 1. The loss decreases quickly in the first 200 steps and then slows down. Each batch (256 samples) took approx 27 sec to train, so the pretain took about 4 hrs on Google Cloud without GPU.

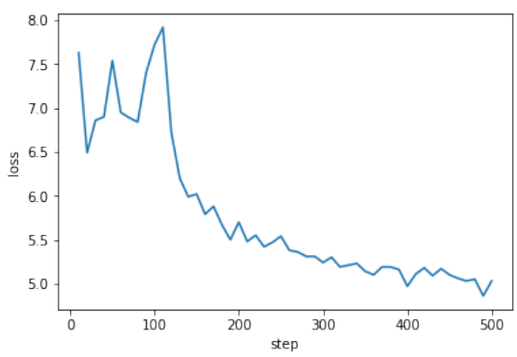

Figure 1: The loss of adversarial training from pretrain.

We train two models: model with random perturbation (RP) and model with virtual adversarial training (VAT) for 300 steps. The loss of RP and VAT is shown in Figure 2 and Figure 3 respectively. For VAT, the loss first decreases and then fluctuates in the first 200 steps and become stable at 0.70 for the remaining 100 steps. The loss in RP is higher than that in VAT, and no clear trend was observed during training.

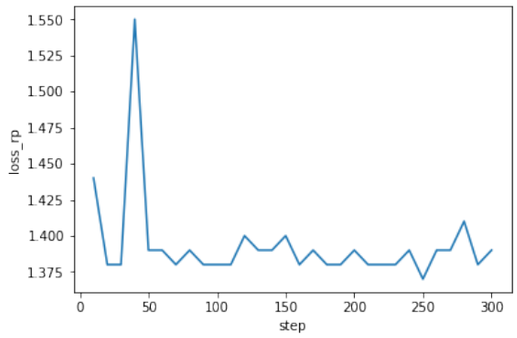

Figure 2: The loss of random perturbation in training.

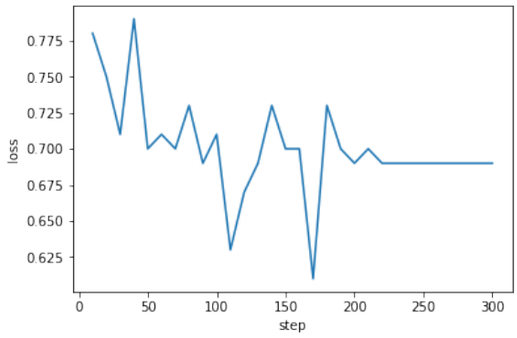

Figure 3: The loss of virtual adversarial training in training.

6 Works Cited

[1] Chen, P.-Y. and Soo, V.-W. Humor recognition using deep learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* volume 2, pp. 113117, 2018.

[2] de Oliveira, Luke/Rodrigo, Alfredo Lainez (2017): Humor Detection in Yelp reviews

[3] Petrovic, S., & Matthews, D. (2013, August). Unsupervised joke generation from big data. In ACL (2) (pp. 228-232)

[4] G. Ritchie. Computational mechanisms for pun generation in *Proceedings of the 10th European Natural Language Generation Workshop,* 2005.

[5] D. Yang, A. Lavie, C. Dyer, and E. Hovy. Humor recognition and humor anchor extraction. 2015.

[6] https://www.kdnuggets.com/2015/07/deep-learning-adversarial-examples-misconceptions.html

[7] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial Training Methods for Semi-Supervised Text Classification. In *Proceedings of the International Conference on Learning Representations (ICLR)*

[8] https://www.yelp.com/dataset/challenge