Predicting Audience's Laughter Using Convolutional Neural Network

Abstract

For the purpose of automatically evaluating speakers' humor usage, we build a presentation corpus containing humorous utterances based on TED talks. Compared to previous data resources supporting humor recognition research, ours has several advantages, including (a) both positive and negative instances coming from a homogeneous data set, (b) containing a large number of speakers, and (c) being open. Focusing on using lexical cues for humor recognition, we systematically compare a newly emerging text classification method based on Convolutional Neural Networks (CNNs) with a well-established conventional method using linguistic knowledge. The advantages of the CNN method are both getting higher detection accuracies and being able to learn essential features automatically.

Introduction

The ability to make effective presentations has been found to be linked with success at school and in the workplace. Humor plays an important role in successful public speaking, e.g., helping to reduce public speaking anxiety often regarded as the most prevalent type of social phobia, generating shared amusement to boost persuasive power, and serving as a means to attract attention and reduce tension BIBREF0 .

Automatically simulating an audience's reactions to humor will not only be useful for presentation training, but also improve conversational systems by giving machines more empathetic power. The present study reports our efforts in recognizing utterances that cause laughter in presentations. These include building a corpus from TED talks and using Convolutional Neural Networks (CNNs) in the recognition.

The remainder of the paper is organized as follows: Section SECREF2 briefly reviews the previous related research; Section SECREF3 describes the corpus we collected from TED talks; Section SECREF4 describes the text classification methods; Section SECREF5 reports on our experiments; finally, Section SECREF6 discusses the findings of our study and plans for future work.

Previous Research

Humor recognition refers to the task of deciding whether a sentence/spoken-utterance expresses a certain degree of humor. In most of the previous studies BIBREF1 , BIBREF2 , BIBREF3 , humor recognition was modeled as a binary classification task. In the seminal work BIBREF1 , a corpus of INLINEFORM0 “one-liners" was created using daily joke websites to collect humorous instances while using formal writing resources (e.g., news titles) to obtain non-humorous instances. Three humor-specific stylistic features, including alliteration, antonymy, and adult slang were utilized together with content-based features to build classifiers. In a recent work BIBREF3 , a new corpus was constructed from the Pun of the Day website. BIBREF3 explained and computed latent semantic structure features based on the following four aspects: (a) Incongruity, (b) Ambiguity, (c) Interpersonal Effect, and (d) Phonetic Style. In addition, Word2Vec BIBREF4 distributed representations were utilized in the model building.

Beyond lexical cues from text inputs, other research has also utilized speakers' acoustic cues BIBREF2 , BIBREF5 . These studies have typically used audio tracks from TV shows and their corresponding captions in order to categorize characters' speaking turns as humorous or non-humorous. Utterances prior to canned laughter that was manually inserted into the shows were treated as humorous, while other utterances were treated as negative cases.

Convolutional Neural Networks (CNNs) have recently been successfully used in several text

categorization tasks (e.g., review rating, sentiment recognition, and question type recognition). Kim2014,Johnson2015,Zhang2015 suggested that using a simple CNN setup, which entails one layer of convolution on top of word embedding vectors, achieves excellent results on multiple tasks. Deep learning recently has been applied to computational humor research BIBREF5 , BIBREF6 . In Bertero2016LREC, CNN was found to be the best model that uses both acoustic and lexical cues for humor recognition. By using Long Short Time Memory (LSTM) cells BIBREF7 , Bertero2016NAACL showed that Recurrent Neural Networks (RNNs) perform better on modeling sequential information than Conditional Random Fields (CRFs) BIBREF8 .

From the brief review, it is clear that corpora used in humor research so far are limited to one-line puns or jokes and conversations from TV comedy shows. There is a great need for an open corpus that can support investigating humor in presentations. CNN-based text categorization methods have been applied to humor recognition (e.g., in BIBREF5) but with limitations: (a) a rigorous comparison with the state-of-the-art conventional method examined in yang-EtAl:2015:EMNLP2 is missing; (b) CNN's performance in the previous research is not quite clear; and (c) some important techniques that can improve CNN performance (e.g., using varied-sized filters and dropout regularization BIBREF10) were not applied. Therefore, the present study is meant to address these limitations.

TED Talk Data

TED Talks are recordings from TED conferences and other special TED programs. In the present study, we focused on the transcripts of the talks. Most transcripts of the talks contain the markup `(Laughter)', which represents where audiences laughed aloud during the talks. This special markup was used to determine utterance labels.

We collected INLINEFORM0 TED Talk transcripts. An example transcription is given in Figure FIGREF4 .

The collected transcripts were split into sentences using the Stanford CoreNLP tool BIBREF11 . In this study, sentences containing or immediately followed by `(Laughter)' were used as `Laughter' sentences, as shown in Figure FIGREF4 ; all other sentences were defined as `No-Laughter' sentences. Following BIBREF1 and BIBREF3 , we selected the same numbers ( INLINEFORM1 ) of `Laughter' and `No-Laughter' sentences. To minimize possible topic shifts between positive and negative instances, for each positive instance, we picked one negative instance nearby (the context window was 7 sentences in this study). For example, in Figure FIGREF4 , a negative instance (corresponding to `sent-2') was selected from the nearby sentences ranging from `sent-7' to `sent+7'.

Methods

Conventional Model

Following yang-EtAl:2015:EMNLP2, we applied Random Forest BIBREF12 to perform humor recognition by using the following two groups of features. The first group are latent semantic structural features covering the following 4 categories: Incongruity (2), Ambiguity (6), Interpersonal Effect (4), and Phonetic Pattern (4). The second group are semantic distance features, including the humor label classes from 5 sentences in the training set that are closest to this sentence (found by using a k-Nearest Neighbors (kNN) method), and each sentence's averaged Word2Vec representations (INLINEFORM0). More details can be found in BIBREF3 .

CNN model

Our CNN-based text classification's setup follows Kim2014. Figure FIGREF17 depicts the model's details.

From the left side's input texts to the right side's prediction labels, different shapes of tensors flow through the entire network for solving the classification task in an end-to-end mode.

Firstly, tokenized text strings were converted to a INLINEFORM0 tensor with shape INLINEFORM1 , where INLINEFORM2 represents sentences' maximum length while INLINEFORM3 represents the word-embedding dimension. In this study, we utilized the Word2Vec BIBREF4 embedding vectors (INLINEFORM4) that were trained on 100 billion words of Google News. Next, the embedding matrix was fed into a INLINEFORM5 convolution network with multiple filters. To cover varied reception fields, we used filters of sizes of INLINEFORM6 , INLINEFORM7 , and INLINEFORM8 . For each filter size, INLINEFORM9 filters were utilized. Then, max pooling, which stands for finding the largest value from a vector, was applied to each feature map (total INLINEFORM10 feature maps) output by the INLINEFORM11 convolution. Finally, maximum values from all of INLINEFORM12 filters were formed as a flattened vector to go through a fully connected (FC) layer to predict two possible labels (Laughter vs. No-Laughter). Note that for INLINEFORM13 convolution and FC layer's input, we applied `dropout' BIBREF10 regularization, which entails randomly setting a proportion of network weights to be zero during model training, to overcome over-fitting. By using cross-entropy as the learning metric, the whole sequential network (all weights and bias) could be optimized by using any SGD optimization, e.g., Adam BIBREF13 , Adadelta BIBREF14 , and so on.

Experiments

We used two corpora: the TED Talk corpus (denoted as TED) and the Pun of the Day corpus (denoted as Pun). Note that we normalized words in the Pun data to lowercase to avoid a possibly elevated result caused by a special pattern: in the original format, all negative instances started with capital letters. The

Pun data allows us to verify that our implementation is consistent with the work reported in yang-EtAl:2015:EMNLP2.

In our experiment, we firstly divided each corpus into two parts. The smaller part (the Dev set) was used for setting various hyper-parameters used in text classifiers. The larger portion (the CV set) was then formulated as a 10-fold cross-validation setup for obtaining a stable and comprehensive model evaluation result. For the PUN data, the Dev contains 482 sentences, while the CV set contains 4344 sentences. For the TED data, the Dev set contains 1046 utterances, while the CV set contains 8406 utterances. Note that, with a goal of building a speaker-independent humor detector, when partitioning our TED data set, we always kept all utterances of a single talk within the same partition. To our knowledge, this is the first time that such a strict experimental setup has been used in recognizing humor in conversations, and it makes the humor recognition task on the TED data quite challenging.

When building conventional models, we developed our own feature extraction scripts and used the SKLL python package for building Random Forest models. When implementing CNN, we used the Keras Python package. Regarding hyper-parameter tweaking, we utilized the Tree Parzen Estimation (TPE) method as detailed in TPE. After running 200 iterations of tweaking, we ended up with the following selection: INLINEFORM0 is 6 (entailing that the various filter sizes are INLINEFORM1), INLINEFORM2 is 100, INLINEFORM3 is INLINEFORM4 and INLINEFORM5 is INLINEFORM6 , optimization uses Adam BIBREF13 . When training the CNN model, we randomly selected INLINEFORM7 of the training data as the validation set for using early stopping to avoid over-fitting.

On the Pun data, the CNN model shows consistent improved performance over the conventional model, as suggested in BIBREF3 . In particular, precision has been greatly increased from INLINEFORM0 to INLINEFORM1 . On the TED data, we also observed that the CNN model helps to increase precision (from INLINEFORM2 to INLINEFORM3) and accuracy (from INLINEFORM4 to INLINEFORM5). The

empirical evaluation results suggest that the CNN-based model has an advantage on the humor recognition task. In addition, focusing on the system development time, generating and implementing those features in the conventional model would take days or even weeks. However, the CNN model automatically learns its optimal feature representation and can adjust the features automatically across data sets. This makes the CNN model quite versatile for supporting different tasks and data domains. Compared with the humor recognition results on the Pun data, the results on the TED data are still quite low, and more research is needed to fully handle humor in authentic presentations.

Discussion

For the purpose of monitoring how well speakers can use humor during their presentations, we have created a corpus from TED talks. Compared to the existing (albeit limited) corpora for humor recognition research, ours has the following advantages: (a) it was collected from authentic talks, rather than from TV shows performed by professional actors based on scripts; (b) it contains about 100 times more speakers compared to the limited number of actors in existing corpora. We compared two types of leading text-based humor recognition methods: a conventional classifier (e.g., Random Forest) based on human-engineered features vs. an end-to-end CNN method, which relies on its inherent representation learning. We found that the CNN method has better performance. More importantly, the representation learning of the CNN method makes it very efficient when facing new data sets.

Stemming from the present study, we envision that more research is worth pursuing: (a) for presentations, cues from other modalities such as audio or video will be included, similar to Bertero2016LREC; (b) context information from multiple utterances will be modeled by using sequential modeling methods.