TRANSFER OF KNOWLEDGE BETWEEN CONCEPTS: AN APPLICATION TO SINCERITY AND DECEPTION PREDICTION

Qinyi Luo[+], Rahul Gupta[o], Shrikanth Narayanan[o]

[+]Tsingua University, Beijing, China
[o]Signal Analysis and Interpretation Lab, University of Southern California, Los Angeles, CA, USA

ABSTRACT

Put abstract here

***Index Terms*—** Pathological speech disorders, machine learning, signal processing

1. INTRODUCTION

Put intro here

2. RELATED WORK

Put background here

3. DATASETS

Speech samples in the D dataset are more similar to natural speech, as the subjects were allowed to arrange speech and show emotions freely. On the contrary, participants for the S dataset were subjected to more restrictions, resulting in more acting. Moreover, labels for the D dataset are objective and binary, while sincerity ratings are subjective and continuous.

We use two datasets for the purpose of our experiments: (i) deception dataset and, (ii) sincerity dataset. These datasets were also used as part of the Interspeech 2016 computational paralinguistic challenge. We breifly describe these two datasets below.

Deceptive Speech Dataset (DSD): The DSD dataset is available from the University of Arizona and we use a set of 1059 speech samples obtained from x participants. In each of these samples, the participant either lies about his identity and behavior or tells the truth. Therefore, each of these samples is associated with a binary label of the speech being deceptive or not.‘ We refer the reader to [] for further details on this dataset.

Sincerity Speech Corpus (SSC): We use the SSC dataset provided by the Columbia Univesity, consisting of 655 speech samples from 22 speakers. Unlike the labelling schemes for the DSD dataset, the samples in the SSC dataset are rated for percieved sincerity by a group of 13 annotators in a range of 0-4. The scores for each annotator were further normalized to zero mean and unit standard deviation. Final sincerity score for each sample is computed as the mean of thus normalized sincerity score from each annotator. Further details regarding the dataset are available in [].

The goal of our experiments is to utilize the expected relationship between deception and perceived sincerity lables. We hypothesize the deceptive utterances would have a low percieved sincerity, and also, utterances with low percieved sincerity are more likely to be decptive. Therefore, in order to improve the performance for one dataset, we aim to utilize the knowledge from the other dataset by either directly using the datasamples from that dataset or using the model learnt on the other dataset. However, we need to account for several dissimilarities between the datasets for a maximal transfer of knowledge from one dataset to the others. These dissimilarities exist in the form of differences in dataset collection conditions and protocols, annotation procedures and the nature of lables. Speech samples in the DSD dataset are more similar to natural speech, as the subjects were allowed to arrange speech and show emotions freely. On the contrary, participants for the SSC dataset were asked to utter pre-specified sentences. Moreover, labels for the DSD dataset are objective and binary, while sincerity ratings in the SSC dataset are subjective and continuous. We address these issues using a few label generation and data transformation techniques. These label generation and data transformation operate on on features extracted from each of the two dataset, as described next.

3.1. Features

We use a set of 6376 acoustic features termed as the ComParE Acoustic Feature Set []. These features are statistics computed on prosodic cues (e.g. pitch, intensity, jitter and, shimmer) and spectral cues (Mel Frequency Cepstral Coefficients xxx..). Further details on these features are listed in Table x in [].

4. METHODOLOGY

The goal of experiments is to exploit the information from one dataset for improved performance of target prediction on the other dataset. For the purpose of this demonstration, we initially set a baseline on each of the two datasets. We then conduct two categories of experiments to utilize the information from a given dataset in improving performance for the other dataset. In the first set of experiments, while predicting the target labels for the dataset at hand, we append the datapoints from the other dataset with labels of interest generated synthetically. In the second experiment, we use the outputs from a sincerity prediction model as features during deception prediction and vice versa. Finally, we combine the two schemes of appending data and using predictions from the other model. Below, we describe each of these modeling schemes in detail.

4.1. Baseline experiments

In the baseline systems, we only use the datapoints corresponding to each individual dataset, without any transfer of knowledge beween the SSC and DSD datasets. We describe the choice of baseline modeling schemes for the deception and sincerity prediction below.

Deception prediction We train a Support Vector Machine (SVM) classifier on the Compare Acoustic Feature Set [] to predict the binary label of an utterance being deceptive or not. Same classifier was also used in the baseline presented in the Interspeech challenge 2016 []. For evaluation purpose, we perform a 10 fold cross-validation, each fold containing an independent set of speakers. 8 partitions are used as training set, 1 as development set and 1 as testing set. SVM parameters such as kernel and box-constraint are tuned on the development set. We use Unweighted Average Recall (UAR) as the evaluation metric on the DSD dataset, as was also the case during the Interspeech challenge 2016 []. The baseline results are listed in Table **??**. Note that the results are slightly different than the one presented in the Interspeech 2016 challenge paper [] as their evaluation defined a different partitioning scheme for the development and the testing set.

Sincerity predicition Since the sincerity predicition involves continuous lables, we train a Support Vector Regressor (SVR) with spearman correlation as the evaluation metric. Same regressor and evaluation metric were used during the Interspeech challenge 2016. In this case, we perform a leave one out speaker crossvalidation due to smaller number of datasamples. Apart from the speaker in the testing set, other speakers are roughly equally divided between the training and the development set. The parameters for the SVR (kerner and box-constraint) are tuned on a development set. Table **??** presents the baseline results. The results are slightly different from the ones presented in the Interspeech 2016 challenge paper [] due to difference in partition and the fact that during the challenge, the box-constraint parameter was tuned globally for each fold. We, on the other hand, find the best parameters for each for indipendently based on the development set.

4.2. Knowledge transfer: appending datapoints

In this section, we append the datapoints from the other dataset in order to improve the performance for the task of interest. First, we generate synthetic labels for the task at hand for each datapoint in the other dataset. These datapoints and synthetic labels are appended with the datapoints and the lables of the original dataset for final model training. We test two synthetic label generation algorithms: Rxxx (Ransac) and label transformation. We discuss these algorithms in detail below.

4.2.1. Rxxx (RANSAC)

During the RANSAC [] implementation, an initial model is trained on the available dataset with lables. Then, the algorithm performs prediction on the datapoints without labels. A fraction of these datapoints in then added to the existing dataset with labels and an updated model is trained with the additional labelled data. Researchers have proposed several criterion for selecting the fraction of data (e.g. adding the datapoints farthest away from class boundaries). The procedure is repeated iteratively till a certain criterion is met (in our experiments, we perform the iterations till the performance on the held out development set is maximized). Since the task objective is different for the DSD and the SSC corpus, we describe them seprately below.

Deception prediction For the purpose of deception prediction, synthetic labels are obtained on the entire SSC dataset. We use the same train, test and development set split as in the baseline experiments. Initially, an SVM model is obtained using the train partition of the DSD dataset and datapoints are added from the SSC dataset till the performance on the development partition of the DSD dataset is maximized. In each iteration, datapoints farthest away from the class boundary are added (more details on this criterion is available in []). We vary the proportion of SSC data added at each iteration from farthest 5% to all of the data.

Sincerity prediction The details of sincerity prediction are similar to deception prediction, except for the fact that the model we train is an SVR. We implement the regression version of RANSAC algorithm as proposed in [] and the train, test and development set partitions are kept same as the baseline experiments. Table **??** shows the results for the RANSAC algorithm implementation.

4.2.2. Label transformation

Previously, we hypothesized that there exists a relation between the sincerity and the deception labels. The RANSAC algorithm does not make of sincerity labels during training decption models (and vice versa). Alternatively during the label transformation, we transform the continous sincerity labels into binary deception labels and vice versa for deception and sincerity prediction, respectively. We describe the label transformation for deception and sincerity prediction below.

Deception prediction In this experiment, we append a part of the sincerity dataset to the DSD dataset durint model training. The portion of the sincerity dataset with percieved sincerity below (/above) a certain threshold are marked as deceptive (/non-deceptive). These two threshold are tuned as a pair for maximal performance on the development partition of the DSD dataset.

Sincerity prediction Obtaining sincerity labels on the deception dataset involves transforming binary labels into a continous sincerity scale. For our experiments, we approximate all deceptive utterances to carry a fixed sincerity rating. Similarly, all non-decptive utterances are labeled with a different constant sincerity ratings. These ratings for the deceptive and non-deceptive utterances are again tuned as a pair for maximal performance on the development partition of the SSC dataset. The results for the label transformation are also listed in Table **??**.

4.3. Knowledge transfer: using sincerity/deception predictions as features

In this section, we propose using sincerity prediction as features during deception prediction and vice versa. This approach is motivated from our previous hypothesis that a relationship exists between deception and sincerity perception, therefore obtaining a sincerity score would be useful for deception prediction and vice versa. In the experiment, we predict the sincerity scores (/deception prediction) on the DSD dataset to aid deception (/sincerity) prediction. These sincerity (/deception) scores are obtained from a model trained on an adapted version of the SSC (/DSD) dataset. As there exists a mismatch between the SSC and DSD datasets, we perform a domain adaptation to reduce the differences in the distribution of SSC and DSD datasets. We provide the experimental details for the experiments below.

Deception prediction We use the train, test and development set partitions for the DSD dataset as described before. The sincerity scores for the DSD dataset are obtained from a model trained on a domain adapted version of the SSC dataset. We use the Geodesic Kernel Flow (GFK) adaptation [] to this effect. **Check** We treat the SSC dataset as the source dataset and adapt it to have a similar distibution like the DSD dataset. We then train a model on the adapted SSC dataset and make predictions on the DSD dataset. The sincerity scores are used in conjunction with the original acoustic features (in section **??**) to predict deception in a stacked generalized framework []. In the stacked generalized framework a model is first trained to obtain predictions from the original features. These predictions are

combined with the predicted sincerity scores on the DSD dataset by another SVM model to provide the final decision.

Sincerity prediction Akin to the deception prediction experiments, we use the predefined train, test and development set partitions for the SSC dataset. We predict the deception outcomes for the SSC dataset using a model trained on an adapted version of the DSD dataset. The adaptation is again performed using GFK adaptation, with DSD dataset modified as source dataset to resemble SSC dataset as the target dataset. An SVM classifier trained on the adapted version of the DSD dataset predicts deception outcomes on the SSC dataset. However, instead of using the binary deception lables, we use distance of datapoints from SVM decision hyperplanes as a soft deception score. We combine these deception scores with the original acoustic features again using a stacked generalization framework, where an initial prediction is first made just based on original features. Another SVR model then combines these initial predictions with the deception scores (distance from SVM hyperplanes) to provide the final sincerity scores.

In the next section

5. RESULTS

Generally speaking, the performance improved after the fusion of labels, no matter which classifier/regressor was used to derive Lbl_{orig}. It did matter, however, whether the GFK domain adaptation was used to obtain Lbl_{other}, as shown in the table that without the domain adaptation, Lbl_{other} would very much resemble random guessing. When using Lbl_{orig} only, i.e. using the methods in 4.2.1 and 4.2.2 and their combination with the baseline method, the performance didn't show improvement in most cases as compared to the baselines.

In deception recognition,

Table1. UAR in deception recognition

	Using Lbl_{orig} only	Using Lbl_{other} only	Using fusion of the 2 labels
Baselines	0.6964		0.7202
RANSAC	(a) 0.6964[1] (b) 0.6964 (c) 0.7026	0.6081	(a) 0.7202 (b) 0.7193 (c) 0.7096
Transform labels	0.6717	*w/o GFK: 0.5275	0.6990
Combined[2]	(a) (b) (c) 0.6933		(a) 0.7005 (b) 0.7005 (c) 0.7059

[1] Listed in the table are results of the three strategies we applied. The strategies are: (a) add 5 percent of unlabeled data in each iteration; (b) add 5 percent of unlabeled data in each iteration; (c) add all of the unlabeled data in one iteration. Please refer to section 4.2.1 for more details.

[2] In the combined method, as stated in 4.2.3, the three methods above were integrated by selecting the classifier that performed the best on the development set in each fold. The said classifier was then used in the fusion. The identifier (a), (b), (c) indicates which RANSAC strategy was used.

Table2. Spearman's correlation in sincerity evaluation

	Using Lbl_{orig} only	Using Lbl_{other} only	Using fusion of the 2 labels
Baselines	0.4651 (0.4158)[1]		0.4853 (0.4784)
RANSAC	0.4645[2](0.3366)	0.1867 (0.1755)	0.4850 (0.4804)
Transform labels	0.4344 (0.2936)	*w/o GFK: 0.0084 (0.0101)	0.4937 (0.4818)
Combined[3]	0.4590 (0.3108)		0.4896 (0.4738)

[1] To conduct significance tests, Pearson's correlation was also computed and is shown in parentheses in the table.

[2] Between the two strategies we applied, only the best result is shown in the table. The strategy that generated the best result was then used in the fusion, while the other was not. Both two results are as follows: (a) add one cluster of the unlabeled data in each iteration: 0.4645; (b) add all of the unlabeled data in one iteration: 0.4351.

[3] In the combined method, as stated in 4.2.3, the three methods above were integrated by selecting the regressor that performed the best on the development set in each fold. The said regressor was then used in the fusion.

6. DISCUSSION

7. CONCLUSION

[?]