Irony Detection – Data Combination

Project for Deep Learning for NLP, WS1718

Anne Beyer

Center for Information and Language Processing

January 31, 2018

Irony Detection

Given:

- One larger data set (tweets, SemEval-2018 Task 3)
- One smaller data set (reddit comments, Kaggle)

Task:

- Predict whether comments are ironic (e.g. LSTM+logistic regression)
- How can one domain (large data set) help prediction on another domain (smaller dataset)?

Data

SemEval dataset Taks 3a of SemEval 2018, 1911 ironic – 1923 non-ironic tweets, extracted using #not #sarcasm #irony, manually checked and balanced

Kaggle dataset Part of a bigger database, 537 ironic – 1412 non-ironic reddit comments, labeled by 3 annotators

Dictionary All words from Kaggle corpus with $freq > 1$ (4608)

Dev and test set 25% splits of Kaggle data

Training set Different setups

Model Architecture

```
irony_model = Sequential()
irony_model.add(Embedding(VOCAB_SIZE,EMBEDDING_SIZE))
irony_model.add(Bidirectional(LSTM(HIDDEN_SIZE)))
irony_model.add(Dense(1,activation="sigmoid"))
irony_model.compile(loss='binary_crossentropy',
optimizer='RMSprop',metrics=['accuracy'])
```

Models

baseline out-of-domain model Only trained on SemEval data

baseline in-domain mode Only trained on training part of Kaggle data (with class weights)

joint model Concatenation of SemEval and Kaggle (with class weights)

weighted model Joint model with sample-weights ($Kaggle * 4$)

continue model Baseline out-of-domain model trained futrher on Kaggle training data (with class weights)

Results

Evaluation of	Accuracy	Precision	Recall	F_1
baseline out-of-domain model	0.34	0.30	0.92	0.45
baseline in-domain model	0.65	0.42	0.45	0.43
joint model	0.67	0.44	0.36	0.39
weighted model	0.68	0.42	0.18	0.26
continue model	0.66	0.43	0.40	0.42

Discussion

- Who would have thought that irony is hard to predict...
- Combination of sample-weight and class-weight
- ...