Sarcasm Detection 🤔

Connie Chen, Mickey Shi, Nathaniel Imel, Sadaf Khan

Task Description: Primary & Adaptation Task

- **Primary task** (subtask A of iSarcasmEval)
- ***Detect:*** whether a given tweet was sarcastic or not

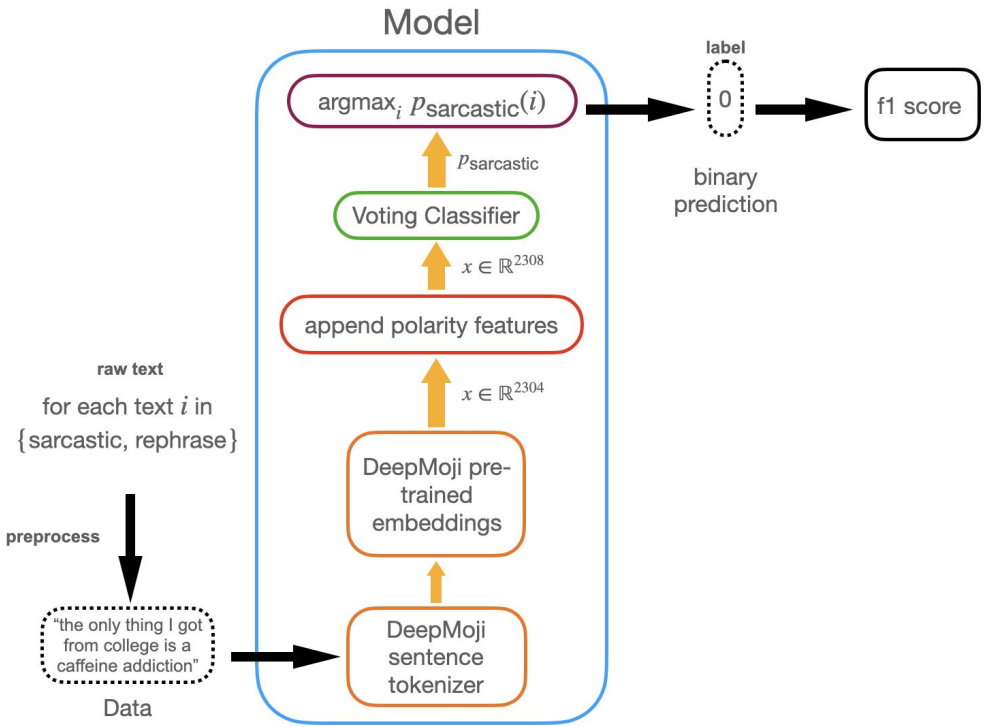
- **Adaptation task** (subtask B of iSarcasmEval)**:**
- ***Discriminate:*** Given two statements (one sarcastic and one annotator-provided rephrase of that sarcastic tweet), determine which of those statements is sarcastic and which one is not

Primary Task Additions for D4

- Adding NRCLex emotion vectors:
  - Encompassing a number of affect features, including fear, joy, negativity, positivity, anger
  - Assumption: lack of anger or a more generally dispassionate tone could signify sarcasm
- Preprocessing:
  - Inclusion of link ‘<link>’ tokens
  - Changing ‘&amp’ to ‘and’

Approach

- Models: top 3 performing models from D3
- Train on primary task, evaluate on adaptation discriminatory task (a “transfer” method).
- Used sklearn’s predict_proba() to generate confidence scores between sarcastic tweet and its non-sarcastic rephrase


Model
label
argmax$_i$ $p_{\text{sarcastic}}(i)$
0
f1 score
$p_{\text{sarcastic}}$
binary prediction
Voting Classifier
$x \in \mathbb{R}^{2308}$
append polarity features
$x \in \mathbb{R}^{2304}$
raw text
for each text i in {sarcastic, rephrase}
DeepMoji pre-trained embeddings
preprocess
"the only thing I got from college is a caffeine addiction"
Data
DeepMoji sentence tokenizer

Results: Primary vs. Adaptation

Classifier	Primary Dev	Primary Test	Adaptation Test
PredictRandom	0.4608	0.4152	0.4636
Voting + Polarity	0.8333	0.6061	0.9066
KNN + Polarity	0.7077	0.6066	0.6971
AdaBoost + Polarity	0.6721	0.5667	0.7232

Table 1: Summary of results across D4 classifiers

Issues & Successes

	Voting w/ polarity	KNN w/ polarity	Adaboost w/ polarity
Removing NRClex	0.906	0.697	0.723
Adding NRClex	0.902	0.691	0.723

- Removing NRClex emotion vectors slightly improved classification performance for adaptation and primary task

Adaptation

- Word length discrepancy between sarcastic tweets and rephrases - rephrases were generally much shorter and less complex structurally

  Sarcastic: Being a business major is legal conversion therapy and must be stopped
  Non-sarcastic: Most business majors are straight people

- Might be responsible for high F1s
- Possibly indicating that we're distinguishing between sentence complexity, rather than sarcasm
- Questionable choices of rephrasing: sarcastic phrase does not necessarily entail non-sarcastic rephrase, or vice versa

Final Lessons

- Use conda and set up dev environments early
- Choosing a relatively deprecated framework (DeepMoji) may lock you out of more modern tools, like BertTweet
- Task and evaluation are sometimes poorly defined (the SarcasmEval dataset was bad)
- System specifications can affect model performance
- Sarcasm is hard, even for human observers

Related Reading

NRCLex:

- Lexicon source is (C) 2016 National Research Council Canada (NRC) and this package is for research purposes only. Source: http://sentiment.nrc.ca/lexicons-for-research/

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2017. Automatic sarcasm detection: A survey.ACM Computing Surveys (CSUR), 50(5):1–22.

David Bamman and Noah Smith. 2015. Contextualized sarcasm detection on twitter. In Proceedings of the International AAAI Conference on Web and Social Media, volume 9, pages 574–577.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm Detection on Twitter: A Behavioral Modeling Approach. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15). Association for Computing Machinery, New York, NY, USA, 97–106.